Global_infected

October 2, 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  …  9/22/20  9/23/20  \
0          0        0        0        0        0  …    39096    39145
1          0        0        0        0        0  …    12666    12787
2          0        0        0        0        0  …    50214    50400
3          0        0        0        0        0  …     1681     1753
4          0        0        0        0        0  …     4236     4363
..       …      …      …      …      … …      …      …
261        0        0        0        0        0  …    36580    37083
262        0        0        0        0        0  …       10       10
263        0        0        0        0        0  …     2028     2029
```

```
264         0         0         0         0         0  …     14389     14443
265         0         0         0         0         0  …      7711      7725

     9/24/20  9/25/20  9/26/20  9/27/20  9/28/20  9/29/20  9/30/20  10/1/20
0      39170    39186    39192    39227    39233    39254    39268    39285
1      12921    13045    13153    13259    13391    13518    13649    13806
2      50579    50754    50914    51067    51213    51368    51530    51690
3       1753     1836     1836     1836     1966     1966     2050     2050
4       4475     4590     4672     4718     4797     4905     4972     5114
..       …       …       …       …       …       …       …       …
261    37591    37963    38253    38703    39121    39541    39899    40322
262       10       10       10       10       10       10       10       10
263     2029     2029     2030     2030     2031     2031     2034     2039
264    14491    14515    14612    14641    14660    14715    14759    14802
265     7752     7787     7803     7812     7816     7837     7838     7850

[266 rows x 258 columns]
```

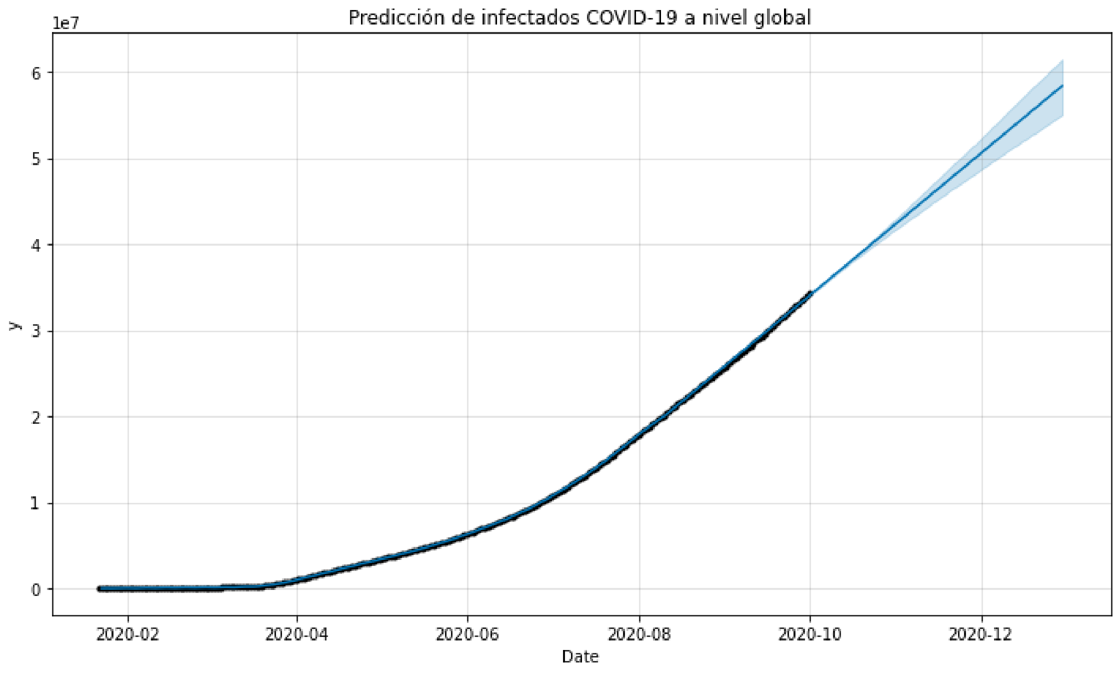
[2]:

```python
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  …  9/22/20  9/23/20  \
0          0        0        0        0        0  …    39096    39145
1          0        0        0        0        0  …    12666    12787
2          0        0        0        0        0  …    50214    50400
3          0        0        0        0        0  …     1681     1753
4          0        0        0        0        0  …     4236     4363
..       …       …       …       …       … …      …       …
261        0        0        0        0        0  …    36580    37083
262        0        0        0        0        0  …       10       10
263        0        0        0        0        0  …     2028     2029
264        0        0        0        0        0  …    14389    14443
265        0        0        0        0        0  …     7711     7725

     9/24/20  9/25/20  9/26/20  9/27/20  9/28/20  9/29/20  9/30/20  10/1/20
0      39170    39186    39192    39227    39233    39254    39268    39285
1      12921    13045    13153    13259    13391    13518    13649    13806
2      50579    50754    50914    51067    51213    51368    51530    51690
3       1753     1836     1836     1836     1966     1966     2050     2050
4       4475     4590     4672     4718     4797     4905     4972     5114
..       …       …       …       …       …       …       …       …
261    37591    37963    38253    38703    39121    39541    39899    40322
262       10       10       10       10       10       10       10       10
263     2029     2029     2030     2030     2031     2031     2034     2039
264    14491    14515    14612    14641    14660    14715    14759    14802
265     7752     7787     7803     7812     7816     7837     7838     7850

[266 rows x 258 columns]
```

[4]:
```python
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]:
```
                                     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]
```

[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                 17
1               Albania                157
2               Algeria                160
```

```
3                Andorra                  0
4                 Angola                142
..                   …                  …
261  West Bank and Gaza                 423
262      Western Sahara                   0
263               Yemen                   5
264              Zambia                  43
265            Zimbabwe                  12

[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
```

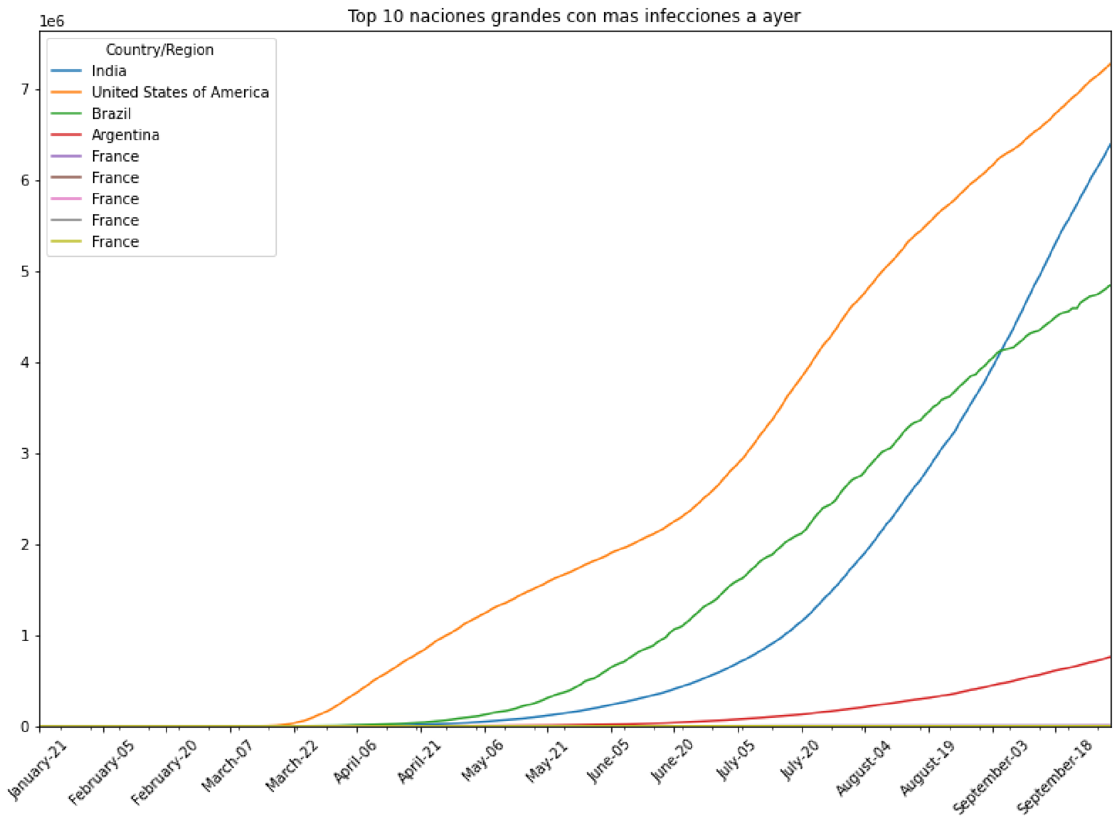
[7]:
```
df_country.sort_values(by='Infected/Million', ascending=False).head(10)
```

[7]:
```
    Country/Region  infected last_day  PopTotal  Infected/Million
140         Israel               7996   8655541        923.801297
92         Czechia               5336  10708982        498.273319
168     Montenegro                215    628062        342.322892
6        Argentina              14001  45195777        309.785580
19         Bahrain                510   1701582        299.721083
23         Belgium               2607  11589616        224.942742
88      Costa Rica               1068   5094114        209.653730
212          Spain               9419  46754782        201.455329
119         France              12918  65273512        197.905699
177    Netherlands               3252  17134873        189.788392
```

[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().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
```

[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                 17  38928341          0.436700
1            Albania                157   2877800         54.555563
2            Algeria                160  43851043          3.648716
3            Andorra                  0     77265          0.000000
4             Angola                142  32866267          4.320539
..               ...                ...       ...               ...
241       Uzbekistan                473  33469199         14.132397
242   Western Sahara                  0    597330          0.000000
243            Yemen                  5  29825967          0.167639
244           Zambia                 43  18383956          2.338996
245         Zimbabwe                 12  14862927          0.807378

[246 rows x 4 columns]
```

[11]:

```
#pintar_grafico_daily(df,naciones_pintar,"Top 10 countries more cases, daily␣
 ↪percentage change")
```

[ ]: