Covid-19 Project

Data cleaning and transformation

Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Loading Data

```
raw_data_confirmed =
pd.read_csv('C:/Users/as355/Downloads/Covid-19/time_series_covid19_con
firmed_global.csv')
raw_data_deaths =
pd.read_csv('C:/Users/as355/Downloads/Covid-19/time_series_covid19_dea
ths_global.csv')
raw_data_recovered =
pd.read_csv('C:/Users/as355/Downloads/Covid-19/time_series_covid19_rec
overed_global.csv')

print("The shape of confirmed:",raw_data_confirmed.shape)
print("The shape of deaths:",raw_data_deaths.shape)
print("The shape of recovered:",raw_data_recovered.shape)
```

```
The shape of confirmed: (271, 350)
The shape of deaths: (271, 350)
The shape of recovered: (256, 350)
```

```
raw_data_recovered
```

```
    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

..             ...                  ...        ...         ...      ...

251            NaN              Vietnam  14.058324  108.277199        0
```

```
252           NaN  West Bank and Gaza  31.952200   35.233200        0

253           NaN               Yemen  15.552727   48.516388        0

254           NaN              Zambia -13.133897   27.849332        0

255           NaN            Zimbabwe -19.015438   29.154857        0


     1/23/20  1/24/20  1/25/20  1/26/20  1/27/20  ...  12/23/20
12/24/20  \
0          0        0        0        0        0  ...     39692
40359
1          0        0        0        0        0  ...     29799
30276
2          0        0        0        0        0  ...     64401
64777
3          0        0        0        0        0  ...      7106
7171
4          0        0        0        0        0  ...      9729
9729
..       ...      ...      ...      ...      ...  ...       ...
...
251        0        0        0        0        0  ...      1281
1281
252        0        0        0        0        0  ...    102942
105233
253        0        0        0        0        0  ...      1384
1384
254        0        0        0        0        0  ...     17969
18036
255        0        0        0        0        0  ...     10259
10314

     12/25/20  12/26/20  12/27/20  12/28/20  12/29/20  12/30/20
12/31/20  \
0       40444     40784     41096     41441     41543     41612
41727
1       30790     31181     31565     32122     32700     33185
33634
2       65144     65505     65862     66214     66550     66855
67127
3        7203      7252      7288      7318      7360      7384
7432
4        9921      9976     10354     10354     10627     10859
11044
..        ...       ...       ...       ...       ...       ...
...
251      1303      1303      1303      1303      1319      1323
1325
```

```
252    106718     108118     109377     110927     113082     115127
117183
253      1384       1384       1384       1384       1387       1392
1394
254     18127      18200      18210      18296      18380      18530
18660
255     10468      10519      10593      10705      11067      11154
11250

     1/1/21
0     41727
1     33634
2     67395
3      7463
4     11146
..      ...
251    1325
252  118926
253    1396
254   18773
255   11347

[256 rows x 350 columns]
```

Unpivoting the data

```
raw_data_confirmed1 =
pd.melt(raw_data_confirmed,id_vars=['Province/State','Country/Region',
'Lat','Long'],var_name='Date')
raw_data_deaths1 =
pd.melt(raw_data_deaths,id_vars=['Province/State','Country/Region','La
t','Long'],var_name='Date')
raw_data_recovered1 =
pd.melt(raw_data_recovered,id_vars=['Province/State','Country/Region',
'Lat','Long'],var_name='Date')

print("The shape of confirmed1:",raw_data_confirmed1.shape)
print("The shape of deaths1:",raw_data_deaths1.shape)
print("The shape of recovered1:",raw_data_recovered1.shape)

The shape of confirmed1: (93766, 6)
The shape of deaths1: (93766, 6)
The shape of recovered1: (88576, 6)
```

Converting the new column date in the dataframes to 'datetime' format

```
raw_data_confirmed1['Date'] =
pd.to_datetime(raw_data_confirmed1['Date'])
raw_data_confirmed1
```

```
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\2916553968.py:1:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is
consistent and as-expected, please specify a format.
  raw_data_confirmed1['Date'] =
pd.to_datetime(raw_data_confirmed1['Date'])
```

```
      Province/State      Country/Region        Lat        Long
Date  \
0                NaN         Afghanistan  33.939110   67.709953 2020-
01-22
1                NaN             Albania  41.153300   20.168300 2020-
01-22
2                NaN             Algeria  28.033900    1.659600 2020-
01-22
3                NaN             Andorra  42.506300    1.521800 2020-
01-22
4                NaN              Angola -11.202700   17.873900 2020-
01-22
...              ...                 ...        ...         ...
...
93761            NaN             Vietnam  14.058324  108.277199 2021-
01-01
93762            NaN  West Bank and Gaza  31.952200   35.233200 2021-
01-01
93763            NaN               Yemen  15.552727   48.516388 2021-
01-01
93764            NaN              Zambia -13.133897   27.849332 2021-
01-01
93765            NaN            Zimbabwe -19.015438   29.154857 2021-
01-01

        value
0           0
1           0
2           0
3           0
4           0
...       ...
93761    1474
93762  139223
93763    2101
93764   20997
```

```
93765   14084

[93766 rows x 6 columns]
```

```
raw_data_deaths1['Date'] = pd.to_datetime(raw_data_deaths1['Date'])
raw_data_deaths1
```

```
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\407412485.py:1: 
UserWarning: Could not infer format, so each element will be parsed 
individually, falling back to `dateutil`. To ensure parsing is 
consistent and as-expected, please specify a format.
  raw_data_deaths1['Date'] = pd.to_datetime(raw_data_deaths1['Date'])

      Province/State      Country/Region        Lat        Long 
Date  \
0                NaN         Afghanistan  33.939110   67.709953 2020-
01-22   
1                NaN             Albania  41.153300   20.168300 2020-
01-22   
2                NaN             Algeria  28.033900    1.659600 2020-
01-22   
3                NaN             Andorra  42.506300    1.521800 2020-
01-22   
4                NaN              Angola -11.202700   17.873900 2020-
01-22   
...              ...                 ...        ...         ...   
...   
93761            NaN             Vietnam  14.058324  108.277199 2021-
01-01   
93762            NaN  West Bank and Gaza  31.952200   35.233200 2021-
01-01   
93763            NaN               Yemen  15.552727   48.516388 2021-
01-01   
93764            NaN              Zambia -13.133897   27.849332 2021-
01-01   
93765            NaN            Zimbabwe -19.015438   29.154857 2021-
01-01   

       value  
0          0  
1          0  
2          0  
3          0  
4          0  
...      ...  
93761     35  
93762   1418  
93763    610  
93764    390  
93765    369
```

```
[93766 rows x 6 columns]
```

```
raw_data_recovered1['Date'] =
pd.to_datetime(raw_data_recovered1['Date'])
raw_data_recovered1
```

```
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\1902036656.py:1:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is
consistent and as-expected, please specify a format.
  raw_data_recovered1['Date'] =
pd.to_datetime(raw_data_recovered1['Date'])
```

```
      Province/State      Country/Region        Lat        Long
Date  \
0                NaN         Afghanistan  33.939110   67.709953 2020-
01-22
1                NaN             Albania  41.153300   20.168300 2020-
01-22
2                NaN             Algeria  28.033900    1.659600 2020-
01-22
3                NaN             Andorra  42.506300    1.521800 2020-
01-22
4                NaN              Angola -11.202700   17.873900 2020-
01-22
...              ...                 ...        ...         ...
...
88571            NaN             Vietnam  14.058324  108.277199 2021-
01-01
88572            NaN  West Bank and Gaza  31.952200   35.233200 2021-
01-01
88573            NaN               Yemen  15.552727   48.516388 2021-
01-01
88574            NaN              Zambia -13.133897   27.849332 2021-
01-01
88575            NaN            Zimbabwe -19.015438   29.154857 2021-
01-01

        value
0           0
1           0
2           0
3           0
4           0
...       ...
88571    1325
88572  118926
88573    1396
88574   18773
```

```
88575   11347

[88576 rows x 6 columns]
```

Renaming columns of dataframes

```
raw_data_confirmed1.columns =
raw_data_confirmed1.columns.str.replace('value','Confirmed')
raw_data_deaths1.columns =
raw_data_deaths1.columns.str.replace('value','Deaths')
raw_data_recovered1.columns =
raw_data_recovered1.columns.str.replace('value','Recovered')

raw_data_recovered1
```

```
      Province/State      Country/Region        Lat        Long
Date  \
0                NaN         Afghanistan  33.939110   67.709953 2020-
01-22
1                NaN             Albania  41.153300   20.168300 2020-
01-22
2                NaN             Algeria  28.033900    1.659600 2020-
01-22
3                NaN             Andorra  42.506300    1.521800 2020-
01-22
4                NaN              Angola -11.202700   17.873900 2020-
01-22
...              ...                 ...        ...         ...
...
88571            NaN             Vietnam  14.058324  108.277199 2021-
01-01
88572            NaN  West Bank and Gaza  31.952200   35.233200 2021-
01-01
88573            NaN               Yemen  15.552727   48.516388 2021-
01-01
88574            NaN              Zambia -13.133897   27.849332 2021-
01-01
88575            NaN            Zimbabwe -19.015438   29.154857 2021-
01-01

       Recovered
0              0
1              0
2              0
3              0
4              0
...          ...
88571       1325
88572     118926
88573       1396
```

```
88574       18773
88575       11347

[88576 rows x 6 columns]
```

Checking null values

```
raw_data_confirmed1.isnull().sum()
```

```
Province/State    65048
Country/Region        0
Lat                 346
Long                346
Date                  0
Confirmed             0
dtype: int64
```

```
raw_data_deaths1.isnull().sum()
```

```
Province/State    65048
Country/Region        0
Lat                 346
Long                346
Date                  0
Deaths                0
dtype: int64
```

```
raw_data_recovered1.isnull().sum()
```

```
Province/State    65394
Country/Region        0
Lat                   0
Long                  0
Date                  0
Recovered             0
dtype: int64
```

Dealing with null values

```
raw_data_confirmed1['Province/
State'].fillna(raw_data_confirmed1['Country/Region'],inplace=True)
raw_data_deaths1['Province/State'].fillna(raw_data_deaths1['Country/
Region'],inplace=True)
raw_data_recovered1['Province/State'].fillna(raw_data_recovered1['Coun
try/Region'],inplace=True)
```

```
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\370297445.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
```

```
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
raw_data_confirmed1['Province/State'].fillna(raw_data_confirmed1['Coun
try/Region'],inplace=True)
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\370297445.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
raw_data_deaths1['Province/State'].fillna(raw_data_deaths1['Country/
Region'],inplace=True)
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\370297445.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
raw_data_recovered1['Province/State'].fillna(raw_data_recovered1['Coun
try/Region'],inplace=True)
```

```
raw_data_recovered1.isnull().sum()
```

```
Province/State    0
Country/Region    0
Lat               0
Long              0
Date              0
```

```
Recovered         0
dtype: int64
```

```
print("The shape of confirmed1:",raw_data_confirmed1.shape)
print("The shape of deaths1:",raw_data_deaths1.shape)
print("The shape of recovered1:",raw_data_recovered1.shape)
```

```
The shape of confirmed1: (93766, 6)
The shape of deaths1: (93766, 6)
The shape of recovered1: (88576, 6)
```

```
raw_data_recovered1
```

```
           Province/State      Country/Region        Lat  
Long  \
0             Afghanistan         Afghanistan  33.939110   67.709953

1                 Albania             Albania  41.153300   20.168300

2                 Algeria             Algeria  28.033900    1.659600

3                 Andorra             Andorra  42.506300    1.521800

4                  Angola              Angola -11.202700   17.873900

...                   ...                 ...        ...         ...

88571             Vietnam             Vietnam  14.058324  108.277199

88572  West Bank and Gaza  West Bank and Gaza  31.952200   35.233200

88573               Yemen               Yemen  15.552727   48.516388

88574              Zambia              Zambia -13.133897   27.849332

88575            Zimbabwe            Zimbabwe -19.015438   29.154857


            Date  Recovered
0     2020-01-22          0
1     2020-01-22          0
2     2020-01-22          0
3     2020-01-22          0
4     2020-01-22          0
...          ...        ...
88571 2021-01-01       1325
88572 2021-01-01     118926
88573 2021-01-01       1396
88574 2021-01-01      18773
88575 2021-01-01      11347
```

```
[88576 rows x 6 columns]
```

Joining Dataframes to form a new dataframe

```
full_join =
raw_data_confirmed1.merge(raw_data_deaths1[['Province/State','Country/
Region','Date','Deaths']],how='left',left_on=['Province/
State','Country/Region','Date'],right_on=['Province/State','Country/
Region','Date'])
full_join.head()
```

```
  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0

   Deaths
0       0
1       0
2       0
3       0
4       0
```

```
full_join =
full_join.merge(raw_data_recovered1[['Province/State','Country/Region'
,'Date','Recovered']],how='left',left_on=['Province/State','Country/
Region','Date'],right_on=['Province/State','Country/Region','Date'])
full_join.head()
```

```
  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0
```

```
   Deaths  Recovered
0       0        0.0
1       0        0.0
2       0        0.0
3       0        0.0
4       0        0.0
```

```python
full_join.isnull().sum()
```

```
Province/State       0
Country/Region       0
Lat                346
Long               346
Date                 0
Confirmed            0
Deaths               0
Recovered         5536
dtype: int64
```

```python
full_join
```

```
           Province/State      Country/Region        Lat
Long  \
0             Afghanistan         Afghanistan  33.939110   67.709953

1                 Albania             Albania  41.153300   20.168300

2                 Algeria             Algeria  28.033900    1.659600

3                 Andorra             Andorra  42.506300    1.521800

4                  Angola              Angola -11.202700   17.873900

...                   ...                 ...        ...         ...

93761             Vietnam             Vietnam  14.058324  108.277199

93762  West Bank and Gaza  West Bank and Gaza  31.952200   35.233200

93763               Yemen               Yemen  15.552727   48.516388

93764              Zambia              Zambia -13.133897   27.849332

93765            Zimbabwe            Zimbabwe -19.015438   29.154857


            Date  Confirmed  Deaths  Recovered
0     2020-01-22          0       0        0.0
1     2020-01-22          0       0        0.0
2     2020-01-22          0       0        0.0
```

```
3     2020-01-22          0       0        0.0
4     2020-01-22          0       0        0.0
...          ...        ...     ...        ...
93761 2021-01-01       1474      35     1325.0
93762 2021-01-01     139223    1418   118926.0
93763 2021-01-01       2101     610     1396.0
93764 2021-01-01      20997     390    18773.0
93765 2021-01-01      14084     369    11347.0

[93766 rows x 8 columns]
```

Adding a month and year column

```
full_join['Month-year'] = full_join['Date'].dt.strftime('%b-%Y')

full_join.head()

  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0

   Deaths  Recovered Month-year
0       0        0.0   Jan-2020
1       0        0.0   Jan-2020
2       0        0.0   Jan-2020
3       0        0.0   Jan-2020
4       0        0.0   Jan-2020
```

Creating a new date column - 1 on a new df

```
full_join2 = full_join.copy()

full_join2['Date-1'] = full_join['Date']+pd.Timedelta(days=1)
full_join2.rename(columns={'Confirmed':'Confirmed-1','Deaths':'Deaths-
1','Recovered':'Recovered-1','Date':'Date Minus 1'},inplace=True)

full_join2.head()

  Province/State Country/Region       Lat       Long Date Minus 1  \
0    Afghanistan    Afghanistan  33.93911  67.709953   2020-01-22
1        Albania        Albania  41.15330  20.168300   2020-01-22
```

```
2        Algeria        Algeria  28.03390   1.659600   2020-01-22
3        Andorra        Andorra  42.50630   1.521800   2020-01-22
4         Angola         Angola -11.20270  17.873900   2020-01-22

   Confirmed-1  Deaths-1  Recovered-1 Month-year     Date-1
0            0         0          0.0   Jan-2020 2020-01-23
1            0         0          0.0   Jan-2020 2020-01-23
2            0         0          0.0   Jan-2020 2020-01-23
3            0         0          0.0   Jan-2020 2020-01-23
4            0         0          0.0   Jan-2020 2020-01-23
```

Creating a yet another new df by joining prev df and recently created df

```
full_join3 =
full_join.merge(full_join2[['Province/State','Country/Region','Date
Minus 1','Confirmed-1','Deaths-1','Recovered-1','Date-
1']],how='left',left_on=['Province/State','Country/
Region','Date'],right_on=['Province/State','Country/Region','Date-1'])
full_join3.head()
```

```
  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0

   Deaths  Recovered Month-year Date Minus 1  Confirmed-1  Deaths-1  \
0       0        0.0   Jan-2020          NaT          NaN       NaN
1       0        0.0   Jan-2020          NaT          NaN       NaN
2       0        0.0   Jan-2020          NaT          NaN       NaN
3       0        0.0   Jan-2020          NaT          NaN       NaN
4       0        0.0   Jan-2020          NaT          NaN       NaN

   Recovered-1 Date-1
0          NaN    NaT
1          NaN    NaT
2          NaN    NaT
3          NaN    NaT
4          NaN    NaT
```

Creating columns in df3

```
full_join3['Confirmed Daily'] = full_join3['Confirmed'] - 
full_join3['Confirmed-1']
full_join3['Deaths Daily'] = full_join3['Deaths'] - 
full_join3['Deaths-1']
full_join3['Recovered Daily'] = full_join3['Recovered'] - 
full_join3['Recovered-1']

full_join3.shape

(93766, 17)

full_join3.head()

  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0

   Deaths  Recovered Month-year Date Minus 1  Confirmed-1  Deaths-1  \
0       0        0.0   Jan-2020          NaT          NaN       NaN
1       0        0.0   Jan-2020          NaT          NaN       NaN
2       0        0.0   Jan-2020          NaT          NaN       NaN
3       0        0.0   Jan-2020          NaT          NaN       NaN
4       0        0.0   Jan-2020          NaT          NaN       NaN

   Recovered-1 Date-1  Confirmed Daily  Deaths Daily  Recovered Daily

0          NaN    NaT              NaN           NaN              NaN

1          NaN    NaT              NaN           NaN              NaN

2          NaN    NaT              NaN           NaN              NaN

3          NaN    NaT              NaN           NaN              NaN

4          NaN    NaT              NaN           NaN              NaN


######################################################################
#######################
######################## Braking the numbers by Day 
#########################################
```

```
######################################################################
#######################


#creating a new df
full_join2 = full_join.copy()

#creating a new date columns - 1
full_join2['Date - 1'] = full_join2['Date'] + pd.Timedelta(days=1)
full_join2.rename(columns={'Confirmed': 'Confirmed - 1', 'Deaths':
'Deaths - 1', 'Recovered': 'Recovered - 1',
                          'Date': 'Date Minus 1'}, inplace=True)

#Joing on the 2 DFs
full_join3 = full_join.merge(full_join2[['Province/State',
'Country/Region','Confirmed - 1', 'Deaths - 1',
                            'Recovered - 1', 'Date - 1', 'Date Minus
1']], how = 'left',
                             left_on =
['Province/State','Country/Region','Date'],
                             right_on = ['Province/State',
'Country/Region','Date - 1'])

#minus_onedf.rename(columns={'Confirmed': 'Confirmed - 1', 'Deaths':
'Deaths - 1', 'Recovered': 'Recovered - 1'}, inplace=True)

full_join3.head()

# Additional Calculations
full_join3['Confirmed Daily'] = full_join3['Confirmed'] -
full_join3['Confirmed - 1']
full_join3['Deaths Daily'] = full_join3['Deaths'] - full_join3['Deaths
- 1']
full_join3['Recovered Daily'] = full_join3['Recovered'] -
full_join3['Recovered - 1']

print(full_join3.shape)
```

```
(93766, 17)
```

```
full_join3.head()
```

```
  Province/State Country/Region       Lat       Long       Date
Confirmed  \
0    Afghanistan    Afghanistan  33.93911  67.709953 2020-01-22
0
1        Albania        Albania  41.15330  20.168300 2020-01-22
0
2        Algeria        Algeria  28.03390   1.659600 2020-01-22
0
3        Andorra        Andorra  42.50630   1.521800 2020-01-22
```

```
0
4         Angola         Angola -11.20270  17.873900 2020-01-22
0

   Deaths  Recovered Month-year  Confirmed - 1  Deaths - 1  Recovered
- 1  \
0       0        0.0   Jan-2020            NaN         NaN
NaN
1       0        0.0   Jan-2020            NaN         NaN
NaN
2       0        0.0   Jan-2020            NaN         NaN
NaN
3       0        0.0   Jan-2020            NaN         NaN
NaN
4       0        0.0   Jan-2020            NaN         NaN
NaN

  Date - 1 Date Minus 1  Confirmed Daily  Deaths Daily  Recovered
Daily
0      NaT          NaT              NaN           NaN
NaN
1      NaT          NaT              NaN           NaN
NaN
2      NaT          NaT              NaN           NaN
NaN
3      NaT          NaT              NaN           NaN
NaN
4      NaT          NaT              NaN           NaN
NaN
```

manually adding values for first date

```
full_join3['Confirmed Daily'].loc[full_join3['Date']=='2020-01-22'] =
full_join3['Confirmed']
full_join3['Deaths Daily'].loc[full_join3['Date']=='2020-01-22'] =
full_join3['Deaths']
full_join3['Recovered Daily'].loc[full_join3['Date']=='2020-01-22'] =
full_join3['Recovered']

C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:1:
FutureWarning: ChainedAssignmentError: behaviour will change in pandas
3.0!
You are setting values through chained assignment. Currently this
works in certain cases, but when using Copy-on-Write (which will
become the default behaviour in pandas 3.0) this will never work to
update the original DataFrame or Series, because the intermediate
object on which we are setting values will behave as a copy.
A typical example is when you are setting values in a column of a
DataFrame, like:
```

```
df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the
assignment in a single step and ensure this keeps updating the
original `df`.

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy

  full_join3['Confirmed Daily'].loc[full_join3['Date']=='2020-01-22']
= full_join3['Confirmed']
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  full_join3['Confirmed Daily'].loc[full_join3['Date']=='2020-01-22']
= full_join3['Confirmed']
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:2:
FutureWarning: ChainedAssignmentError: behaviour will change in pandas
3.0!
You are setting values through chained assignment. Currently this
works in certain cases, but when using Copy-on-Write (which will
become the default behaviour in pandas 3.0) this will never work to
update the original DataFrame or Series, because the intermediate
object on which we are setting values will behave as a copy.
A typical example is when you are setting values in a column of a
DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the
assignment in a single step and ensure this keeps updating the
original `df`.

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy

  full_join3['Deaths Daily'].loc[full_join3['Date']=='2020-01-22'] =
full_join3['Deaths']
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  full_join3['Deaths Daily'].loc[full_join3['Date']=='2020-01-22'] =
full_join3['Deaths']
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:3:
FutureWarning: ChainedAssignmentError: behaviour will change in pandas
3.0!
You are setting values through chained assignment. Currently this
works in certain cases, but when using Copy-on-Write (which will
become the default behaviour in pandas 3.0) this will never work to
update the original DataFrame or Series, because the intermediate
object on which we are setting values will behave as a copy.
A typical example is when you are setting values in a column of a
DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the
assignment in a single step and ensure this keeps updating the
original `df`.

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy

  full_join3['Recovered Daily'].loc[full_join3['Date']=='2020-01-22']
= full_join3['Recovered']
C:\Users\as355\AppData\Local\Temp\ipykernel_15976\891303359.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  full_join3['Recovered Daily'].loc[full_join3['Date']=='2020-01-22']
= full_join3['Recovered']
```

Deleting certain columns

```
del full_join3['Confirmed - 1']
del full_join3['Deaths - 1']
del full_join3['Recovered - 1']
del full_join3['Date - 1']
del full_join3['Date Minus 1']

full_join3
```

```
           Province/State      Country/Region        Lat
Long  \
0             Afghanistan         Afghanistan  33.939110   67.709953
```

```
1                   Albania             Albania  41.153300   20.168300

2                   Algeria             Algeria  28.033900    1.659600

3                   Andorra             Andorra  42.506300    1.521800

4                    Angola              Angola -11.202700   17.873900

...                     ...                 ...        ...         ...

93761               Vietnam             Vietnam  14.058324  108.277199

93762  West Bank and Gaza  West Bank and Gaza  31.952200   35.233200

93763                 Yemen               Yemen  15.552727   48.516388

93764                Zambia              Zambia -13.133897   27.849332

93765              Zimbabwe            Zimbabwe -19.015438   29.154857


            Date  Confirmed  Deaths  Recovered Month-year  Confirmed
Daily  \
0     2020-01-22          0       0        0.0   Jan-2020
0.0
1     2020-01-22          0       0        0.0   Jan-2020
0.0
2     2020-01-22          0       0        0.0   Jan-2020
0.0
3     2020-01-22          0       0        0.0   Jan-2020
0.0
4     2020-01-22          0       0        0.0   Jan-2020
0.0
...          ...        ...     ...        ...        ...
...
93761 2021-01-01       1474      35     1325.0   Jan-2021
9.0
93762 2021-01-01     139223    1418   118926.0   Jan-2021
1219.0
93763 2021-01-01       2101     610     1396.0   Jan-2021
2.0
93764 2021-01-01      20997     390    18773.0   Jan-2021
272.0
93765 2021-01-01      14084     369    11347.0   Jan-2021
217.0

       Deaths Daily  Recovered Daily
0               0.0              0.0
1               0.0              0.0
2               0.0              0.0
```

```
3                0.0              0.0
4                0.0              0.0
...              ...              ...
93761            0.0              0.0
93762           18.0           1743.0
93763            0.0              2.0
93764            2.0            113.0
93765            6.0             97.0

[93766 rows x 12 columns]
```

Removing negative values

```
full_join3[full_join3['Deaths Daily']<0]
```

```
               Province/State          Country/Region        Lat
Long  \
14778                 Iceland                 Iceland  64.963100  -
19.020800
15653             Philippines             Philippines  12.879721
121.774017
15862                 Iceland                 Iceland  64.963100  -
19.020800
16134                   India                   India  20.593684
78.962880
16311                  Quebec                  Canada  52.939900  -
73.549100
...                       ...                     ...        ...
...
87140                 Ireland                 Ireland  53.142400   -
7.692100
87259                   Yemen                   Yemen  15.552727
48.516388
88203                  France                  France  46.227600
2.213700
88585              Tajikistan              Tajikistan  38.861000
71.276100
92439  Bosnia and Herzegovina  Bosnia and Herzegovina  43.915900
17.679100

            Date  Confirmed  Deaths  Recovered Month-year  Confirmed
Daily  \
14778 2020-03-16        180       0        0.0   Mar-2020
9.0
15653 2020-03-19        217      17        8.0   Mar-2020
15.0
15862 2020-03-20        409       0        5.0   Mar-2020
79.0
16134 2020-03-21        330       4       23.0   Mar-2020
86.0
```

```
16311 2020-03-22        219       4         NaN   Mar-2020
38.0
...          ...        ...     ...         ...        ...
...
87140 2020-12-08      74682    2097     23364.0   Dec-2020
214.0
87259 2020-12-08       2078     606      1382.0   Dec-2020           -
305.0
88203 2020-12-12    2350793   57210    152555.0   Dec-2020           -
130.0
88585 2020-12-13      12704      88     12133.0   Dec-2020
80.0
92439 2020-12-28     109911    3942     76121.0   Dec-2020
220.0

       Deaths Daily  Recovered Daily
14778          -5.0             -8.0
15653          -2.0              3.0
15862          -1.0              0.0
16134          -1.0              3.0
16311          -1.0              NaN
...             ...              ...
87140          -2.0              0.0
87259         -43.0           -172.0
88203          -1.0              0.0
88585          -1.0             83.0
92439         -11.0            404.0

[83 rows x 12 columns]
```

```
full_join3['Deaths Daily'] = np.where(full_join3['Deaths
Daily']<0,0,full_join3['Deaths Daily'])

full_join3['Confirmed Daily'] = np.where(full_join3['Confirmed
Daily']<0,0,full_join3['Confirmed Daily'])

full_join3['Recovered Daily'] = np.where(full_join3['Recovered
Daily']<0,0,full_join3['Recovered Daily'])

full_join3
```

```
           Province/State      Country/Region        Lat
Long  \
0             Afghanistan         Afghanistan  33.939110   67.709953

1                 Albania             Albania  41.153300   20.168300

2                 Algeria             Algeria  28.033900    1.659600

3                 Andorra             Andorra  42.506300    1.521800
```

```
4                   Angola              Angola -11.202700   17.873900

...                    ...                 ...        ...         ...

93761              Vietnam             Vietnam  14.058324  108.277199

93762  West Bank and Gaza  West Bank and Gaza  31.952200   35.233200

93763                Yemen               Yemen  15.552727   48.516388

93764               Zambia              Zambia -13.133897   27.849332

93765             Zimbabwe            Zimbabwe -19.015438   29.154857


            Date  Confirmed  Deaths  Recovered Month-year  Confirmed
Daily  \
0     2020-01-22          0       0        0.0   Jan-2020
0.0
1     2020-01-22          0       0        0.0   Jan-2020
0.0
2     2020-01-22          0       0        0.0   Jan-2020
0.0
3     2020-01-22          0       0        0.0   Jan-2020
0.0
4     2020-01-22          0       0        0.0   Jan-2020
0.0
...          ...        ...     ...        ...        ...
...
93761 2021-01-01       1474      35     1325.0   Jan-2021
9.0
93762 2021-01-01     139223    1418   118926.0   Jan-2021
1219.0
93763 2021-01-01       2101     610     1396.0   Jan-2021
2.0
93764 2021-01-01      20997     390    18773.0   Jan-2021
272.0
93765 2021-01-01      14084     369    11347.0   Jan-2021
217.0

       Deaths Daily  Recovered Daily
0               0.0              0.0
1               0.0              0.0
2               0.0              0.0
3               0.0              0.0
4               0.0              0.0
...             ...              ...
93761           0.0              0.0
93762          18.0           1743.0
93763           0.0              2.0
```

```
93764            2.0              113.0
93765            6.0               97.0

[93766 rows x 12 columns]

full_join3.pd.to_csv('C:\Users\as355\Downloads/prepared_Covid_19_date
```