

```
In [2]: !pip install pandas

Collecting pandas
  Downloading pandas-1.1.2-cp38-cp38-win_amd64.whl (9.6 MB)
Requirement already satisfied: python-dateutil<=2.7.3 in c:\users\sony\appdata\local\programs\python\python38\lib\site-packages (from pandas) (2.8.1)
Collecting numpy>=1.15.4
  Downloading numpy-1.19.2-cp38-cp38-win_amd64.whl (13.0 MB)
Collecting pytz>=2017.2
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
Requirement already satisfied: six>=1.5 in c:\users\sony\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil<=2.7.3->pandas) (1.15.0)
Installing collected packages: numpy, pytz, pandas
Successfully installed numpy-1.19.2 pandas-1.1.2 pytz-2020.1

In [5]: import pandas as pd
ds=pd.read_csv(r"C:\Users\sony\Desktop\data sets\covid19.csv")
ds

Out[5]:
Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
182     West Bank and Gaza      10621      78      3752.0  6791.0      152      2.0      0      0.73      35.33      2.08      8916.0      1705      19.12      Eastern Mediterranean
183     Western Sahara          10      1      8.0  1.0      0      0.0      0      10.00      80.00      12.50      10.0      0      0.00      Africa
184      Yemen          1691      483      833.0  375.0      10      4.0      36      28.56      49.26      57.98      1619.0      72      4.45      Eastern Mediterranean
185      Zambia          4552      140      2815.0  1597.0      71      1.0      465      3.08      61.84      4.97      3326.0      1226      36.86      Africa
186      Zimbabwe          2704      36      542.0  2126.0      192      2.0      24      1.33      20.04      6.64      1713.0      991      57.85      Africa

187 rows x 15 columns

In [6]: ds.head()

Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa

In [7]: ds.head(28)

Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa
5      Argentina        167416      3059      72575.0  91782.0      4990      120.0      2057      1.83      43.35      4.21      130774.0      36642      28.02      Americas
6      Armenia          37390      711      26665.0  10014.0      73      6.0      187      1.90      71.32      2.67      34981.0      2409      6.89      Europe
7      Australia        15303      167      9311.0  5825.0      368      6.0      137      1.09      60.84      1.79      12428.0      2875      23.13      Western Pacific
8      Austria          20558      713      18246.0  1599.0      86      1.0      37      3.47      86.75      3.91      19743.0      815      4.13      Europe
9      Azerbaijan        30446      423      23242.0  6781.0      396      6.0      558      1.39      76.34      1.82      27980.0      2556      9.16      Europe
10     Bahamas          382      11      91.0  280.0      40      NaN      0      2.88      23.82      12.09      174.0      208      119.54      Americas
11     Bahrain          39482      141      36110.0  3231.0      351      1.0      421      0.36      91.46      0.39      36936.0      2546      6.89      Eastern Mediterranean
12     Bangladesh        236225      2965      125683.0  97577.0      2772      37.0      1801      1.31      55.56      2.36      207453.0      18772      9.05      South-East Asia
13     Barbados          86      3      65.0  18.0      4      0.0      5      3.49      75.58      4.62      NaN      10      13.16      Americas
14     Argentina        167416      3059      72575.0  91782.0      4990      120.0      2057      1.83      43.35      4.21      130774.0      36642      28.02      Americas
15     Armenia          37390      711      26665.0  10014.0      73      6.0      187      1.90      71.32      2.67      34981.0      2409      6.89      Europe
16     Australia        15303      167      9311.0  5825.0      368      6.0      137      1.09      60.84      1.79      12428.0      2875      23.13      Western Pacific
17     Austria          20558      713      18246.0  1599.0      86      1.0      37      3.47      86.75      3.91      19743.0      815      4.13      Europe
18     Azerbaijan        30446      423      23242.0  6781.0      396      6.0      558      1.39      76.34      1.82      27980.0      2556      9.16      Europe
19     Bahamas          382      11      91.0  280.0      40      NaN      0      2.88      23.82      12.09      174.0      208      119.54      Americas
20     Bahrain          39482      141      36110.0  3231.0      351      1.0      421      0.36      91.46      0.39      36936.0      2546      6.89      Eastern Mediterranean
21     Bangladesh        236225      2965      125683.0  97577.0      2772      37.0      1801      1.31      55.56      2.36      207453.0      18772      9.05      South-East Asia
22     Barbados          86      3      65.0  18.0      4      0.0      5      3.49      75.58      4.62      NaN      10      13.16      Americas
23     Argentina        167416      3059      72575.0  91782.0      4990      120.0      2057      1.83      43.35      4.21      130774.0      36642      28.02      Americas
24     Armenia          37390      711      26665.0  10014.0      73      6.0      187      1.90      71.32      2.67      34981.0      2409      6.89      Europe
25     Australia        15303      167      9311.0  5825.0      368      6.0      137      1.09      60.84      1.79      12428.0      2875      23.13      Western Pacific
26     Austria          20558      713      18246.0  1599.0      86      1.0      37      3.47      86.75      3.91      19743.0      815      4.13      Europe
27     Azerbaijan        30446      423      23242.0  6781.0      396      6.0      558      1.39      76.34      1.82      27980.0      2556      9.16      Europe
28     Bahamas          382      11      91.0  280.0      40      NaN      0      2.88      23.82      12.09      174.0      208      119.54      Americas

In [8]: ds.tail(7)

Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
180     Venezuela        15988      146      9959.0  5883.0      525      4.0      213      0.91      62.29      1.47      12334.0      3654      29.63      Americas
181      Vietnam          431      0      36610.0  660.0      11      0.0      0      0.00      0.00      0.00      384.0      47      12.24      Western Pacific
182     West Bank and Gaza      10621      78      3752.0  6791.0      152      2.0      0      0.73      35.33      2.08      8916.0      1705      19.12      Eastern Mediterranean
183     Western Sahara          10      1      8.0  1.0      0      0.0      0      10.00      80.00      12.50      10.0      0      0.00      Africa
184      Yemen          1691      483      833.0  375.0      10      4.0      36      28.56      49.26      57.98      1619.0      72      4.45      Eastern Mediterranean
185      Zambia          4552      140      2815.0  1597.0      71      1.0      465      3.08      61.84      4.97      3326.0      1226      36.86      Africa
186      Zimbabwe          2704      36      542.0  2126.0      192      2.0      24      1.33      20.04      6.64      1713.0      991      57.85      Africa

187 rows x 15 columns

In [9]: ds.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Country/Region  187 non-null    object
1   Confirmed      187 non-null    int64
2   Deaths        187 non-null    int64
3   Recovered      186 non-null    float64
4   Active         186 non-null    float64
5   New cases      187 non-null    int64
6   New deaths     186 non-null    float64
7   New recovered  187 non-null    int64
8   Deaths / 100 Cases  186 non-null    float64
9   Recovered / 100 Cases  187 non-null    float64
10  Deaths / 100 Recovered  187 non-null    float64
11  Confirmed last week  187 non-null    int64
12  1 week change   186 non-null    int64
13  1 week % increase  186 non-null    float64
14  WHO Region     187 non-null    object
dtypes: float64(8), int64(5), object(2)
memory usage: 22.0+ KB

In [12]: ds.shape

(187, 15)

In [17]: #handling duplicates
#using append() will return a copy without affecting the original DataFrame, we are capturing this copy in temp so we aren't working with the real data.
#notice call .shape quickly proves our DataFrame rows have doubled.
temp_ds.append(ds)
temp_ds.shape

Out[17]:
(374, 15)

In [20]: #dropping duplicates
#just like append(), the drop_duplicates() method will also return a copy of your DataFrame, but this time with duplicates removed.
temp_ds.drop_duplicates(inplace=True)
temp_ds

Out[20]:
Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
182     West Bank and Gaza      10621      78      3752.0  6791.0      152      2.0      0      0.73      35.33      2.08      8916.0      1705      19.12      Eastern Mediterranean
183     Western Sahara          10      1      8.0  1.0      0      0.0      0      10.00      80.00      12.50      10.0      0      0.00      Africa
184      Yemen          1691      483      833.0  375.0      10      4.0      36      28.56      49.26      57.98      1619.0      72      4.45      Eastern Mediterranean
185      Zambia          4552      140      2815.0  1597.0      71      1.0      465      3.08      61.84      4.97      3326.0      1226      36.86      Africa
186      Zimbabwe          2704      36      542.0  2126.0      192      2.0      24      1.33      20.04      6.64      1713.0      991      57.85      Africa

187 rows x 15 columns

In [23]: #when inplace = True , the data is modified in place, which means it will return nothing and the dataframe is now updated. when inplace = False , which is the temp_ds.drop_duplicates(inplace=False)
temp_ds

Out[23]:
Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
182     West Bank and Gaza      10621      78      3752.0  6791.0      152      2.0      0      0.73      35.33      2.08      8916.0      1705      19.12      Eastern Mediterranean
183     Western Sahara          10      1      8.0  1.0      0      0.0      0      10.00      80.00      12.50      10.0      0      0.00      Africa
184      Yemen          1691      483      833.0  375.0      10      4.0      36      28.56      49.26      57.98      1619.0      72      4.45      Eastern Mediterranean
185      Zambia          4552      140      2815.0  1597.0      71      1.0      465      3.08      61.84      4.97      3326.0      1226      36.86      Africa
186      Zimbabwe          2704      36      542.0  2126.0      192      2.0      24      1.33      20.04      6.64      1713.0      991      57.85      Africa

187 rows x 15 columns

In [24]: #keep has 3 options...
#1: first: (default) Drop duplicates except for the first occurrence.(default option is first.This means that if two rows are the same pandas will drop the second)
#2:last: Drop duplicates except for the last occurrence.(using last has the opposite effect: the first row is dropped.)
#3:False: drop all duplicates.(will drop all duplicates. If two rows are the same then both will be dropped.)
#if you're wondering why you would want to do this, one reason is that it allows you to locate all duplicates in your dataset.
temp_ds.drop_duplicates(inplace=True,keep=False)
temp_ds.shape

Out[24]:
(187, 15)

In [25]: #column cleanup
ds.columns

Out[25]:
Index(['Country/Region', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'New cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases', 'Recovered / 100 Cases', 'Deaths / 100 Recovered', 'Confirmed last week', '1 week change', '1 week % increase', 'WHO Region'],
      dtype='object')

In [27]: ds.rename(columns={
    'Confirmed':'Confirmed cases'
    },inplace=True)
ds.columns

Out[27]:
Index(['Country/Region', 'Confirmed cases', 'Deaths', 'Recovered', 'Active', 'New cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases', 'Recovered / 100 Cases', 'Deaths / 100 Recovered', 'Confirmed last week', '1 week change', '1 week % increase', 'WHO Region'],
      dtype='object')

In [28]: ds

Country/Region  Confirmed cases  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      Afghanistan      36263      1269      25198.0  9796.0      106      10.0      18      3.50      69.49      5.04      35526.0      737      2.07      Eastern Mediterranean
1      Albania          4880      144      2745.0  1991.0      117      6.0      63      2.95      56.25      5.25      4171.0      709      17.00      Europe
2      Algeria          27973      1163      18837.0  7973.0      616      8.0      749      4.16      67.34      6.17      23691.0      4282      18.07      Africa
3      Andorra          907      52      803.0  52.0      10      0.0      0      5.73      88.53      6.48      884.0      23      2.60      Europe
4      Angola           950      41      242.0  667.0      18      1.0      0      4.32      25.47      16.94      749.0      201      26.84      Africa
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
182     West Bank and Gaza      10621      78      3752.0  6791.0      152      2.0      0      0.73      35.33      2.08      8916.0      1705      19.12      Eastern Mediterranean
183     Western Sahara          10      1      8.0  1.0      0      0.0      0      10.00      80.00      12.50      10.0      0      0.00      Africa
184      Yemen          1691      483      833.0  375.0      10      4.0      36      28.56      49.26      57.98      1619.0      72      4.45      Eastern Mediterranean
185      Zambia          4552      140      2815.0  1597.0      71      1.0      465      3.08      61.84      4.97      3326.0      1226      36.86      Africa
186      Zimbabwe          2704      36      542.0  2126.0      192      2.0      24      1.33      20.04      6.64      1713.0      991      57.85      Africa

187 rows x 15 columns

In [29]: #renaming each column using lambda function...
#ds.columns = [col.lower() for col in ds]

In [30]: #work with missing values
#replace nulls with non-null values, a technique known as imputation
#let's calculate to total number of nulls in each column of our dataset. The first step is to check which cells in our DataFrame are null:
ds.isnull()

Out[30]:
Country/Region  Confirmed cases  Deaths  Recovered  Active  New cases  New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  Deaths / 100 Recovered  Confirmed last week  1 week change  1 week % increase  WHO Region
0      False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
1      False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
2      False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
3      False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
4      False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...
182     False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
183     False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
184     False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
185     False      False      False      False      False      False      False      False      False      False      False      False      False      False      False
186     False      False      False      False      False      False      False      False      False      False      False      False      False      False      False

187 rows x 15 columns

In [34]: #notice isnull() returns a DataFrame where each cell is either True or False depending on that cell's null status.
#to count the number of nulls in each column we use an aggregate function for summing:
ds.isnull().sum()

Out[34]:
Country/Region      0
Confirmed cases      0
Deaths               0
Recovered            2
Active              1
New cases            0
New deaths           1
New recovered        0
Deaths / 100 Cases  1
Recovered / 100 Cases  0
Deaths / 100 Recovered  0
Confirmed last week  1
1 week change        1
1 week % increase    1
WHO Region           0
dtype: int64

In [35]: #removing numm values
ds.shape

Out[35]:
(187, 15)

In [36]: #What's with this axis=parameter?
#if we want to drop column axis parameter is used..
#as we learned above, this is a tuple that represents the shape of the DataFrame, i.e. 187 rows and 15 columns. Note that the rows are at index zero of this is why axis=1 affects columns. This comes from NumPy of why learning NumPy..

In [38]: #there may be instances where dropping every row with a null value removes too big a chunk from your dataset, so instead we can impute that null with another rec'ds["Recovered"]
rec1

Out[38]:
0      25198.0
1      2745.0
2      18837.0
3      803.0
4      242.0

182     3752.0
183      8.0
184      833.0
185      2815.0
186      542.0
Name: Recovered, Length: 187, dtype: float64

In [39]: rec1.head

Out[39]:
0      25198.0
1      2745.0
2      18837.0
3      803.0
4      242.0

182     3752.0
183      8.0
184      833.0
185      2815.0
186      542.0
Name: Recovered, Length: 187, dtype: float64

In [40]: rec1.head(3)

Out[40]:
0      25198.0
1      2745.0
2      18837.0
Name: Recovered, dtype: float64

In [42]: #calculating mean of 'Recovered' column
rec1_mean=rec1.mean()

Out[42]:
151178.35135135135

In [43]: #with the mean, let's fill the nulls using fillna()
rec1.fillna(rec1_mean,inplace=True)
#we have now replaced all nulls in revenue with the mean of the column. Notice that by using inplace=True we have actually affected the original ds:

In [44]: ds.isnull().sum()

Out[44]:
Country/Region      0
Confirmed cases      0
Deaths               0
Recovered            0
Active              1
New cases            0
New deaths           1
New recovered        0
Deaths / 100 Cases  1
Recovered / 100 Cases  0
Deaths / 100 Recovered  0
Confirmed last week  1
1 week change        1
1 week % increase    1
WHO Region           0
dtype: int64

In [45]: #using describe() on an entire DataFrame we can get a summary of the distribution of continuous variables:
ds.describe()

Out[45]:
Confirmed cases      187.000000      Deaths      187.000000      Recovered      1.860000e+02      Active      187.000000      New cases      186.000000      New deaths      187.000000      Deaths / 100 Cases      186.000000      Recovered / 100 Cases      187.000000      Deaths / 100 Recovered      187.00      Confirmed last week      187.000000      1 week change      187.000000      1 week % increase      186.000000
count      1.870000e+02      187.000000      1.870000e+02      1.860000e+02      187.000000      186.000000      187.000000      186.000000      187.000000      186.000000      187.000000      187.00      187.000000      186.000000
mean      8.813094e+04      3497.518177      5.111785e+04      3.417303e+04      1222.957219      29.112803      933.812834      3.015731      62.263594      inf      7.910509e+04      9448.459893      12.652097
std      3.833187e+05      14100.02482      1.901515e+05      2.138991e+05      5710.374790      120.342229      4197.719635      4.363241      28.820676      NaN      3.391372e+05      47491.127684      20.804555
min      1.114000e+01      0.000000      6.010000e+00      0.000000e+00      0.000000      0.000000      0.000000      0.000000      0.000000      0.000000      1.00      1.000000e+01      47.000000      -3.840000
25%      1.114000e+01      18.500000      6.010000e+00      1.407500e+00      0.000000      0.000000      0.000000      0.000000      0.942500      0.45      1.064250e+01      49.000000      2.762500
50%      5.059000e+03      108.000000      3.111000e+04      1.599500e+03      49.000000      1.000000      221.000000      3.140000      81.320000      3.62      5.329500e+03      432.000000      16.547500
75%      4.046050e+04      734.000000      2.421000e+04      9.281500e+03      419.500000      6.000000      33728.000000      28.560000      100.000000      inf      3.834677e+06      455562.000000      226.320000
max      4.290250e+06      148011.000000      1.846641e+06      2.816444e+06      56336.000000      1076.000000      33728.000000      28.560000      100.000000      inf      3.834677e+06      455562.000000      226.320000

In [48]: #understanding which numbers are continuous also comes in handy when thinking about the type of plot to use to represent your data visually.
ds[Deaths].describe()

Out[48]:
count      187.000000
mean      3497.518177
std      14100.002482
min      0.000000
25%      18.500000
50%      108.000000
75%      734.000000
max      148011.000000
Name: Deaths, dtype: float64

In [55]: #Relationships between continuous variables
#by using the correlation method .corr() we can generate the relationship between each continuous variable:
ds.corr()

Out[55]:
Confirmed cases      1.000000      Deaths      0.934698      Recovered      0.900602      Active      0.927018      New cases      0.909720      New deaths      0.871645      New recovered      0.895252      Deaths / 100 Cases      0.063066      Recovered / 100 Cases      -0.064815      Deaths / 100 Recovered      0.025175      Confirmed last week      0.999127      1 week change      0.954710      1 week % increase      -0.001413
Deaths      0.934698      1.000000      
```