GDV - Lab 2 - Project Student Submission

Authors: Bryan Van Huyneghem & Anoek Strumane (group 8)

1. Preprocessing of the CoViD dataset

The CoViD dataset comes as many csv files, one for each day, that have to be combined into one dataframe. We will first load all csv files into dataframes using pandas and afterwards merge these into one large dataframe.

1.1 Initial csv header exploration

Upon exploring the csv files and interpreting their columns, it becomes apparent:

- **that not all csv files have the same (amount of) columns;**
- **that certain columns have names that closely resemble each other, e.g. Country_Region and Country/Region;**
- **that certain columns only appear to have entries for certain countries, e.g. FIPS (Federal Information Processing Standards (source)) is a column unique to entries related to the U.S., however the FIPS of 250.0 relates to France;**
- **that countries that have data on multiple of their provinces have a column named Combined_Key, which combines data from three comlumns, i.e. Admin2, Province_State and Country_Region into one column.**

Closer inspection reveals:

- **that the csv files use Province/State , Country/Region, Last Update, Latitude and Longitude rather than Province_State, Country_Region, Last_Update, Lat and Long_ until *03-24-2020*;**
- **Incidence_Rate and Case-Fatality_Ratio are added from *05-29-2020* and onwards. No csv file before this date contains these columns;**
- ***Mainland China* turns into *China* in later csv files, but these two inidicate the same region. Thus, *Mainland China* needs to be changed to *China*.**
- **the column Last Update uses different timestamps than Last_Update. We will have to parse the former dates into the same format as entries for Last_Update.**

Thus, it makes sense to first import all csv files until 03-24-2020 and change their column names appropriately, so that they match the column names of all other csv files.

1.2 Preprocessing

We should first note that we have chosen to append all data to each other, because this will allow us to easily group by regions. Alternatively, we could have use a multi index, using the Country_Region, Province_State and Admin2 as indices, which would result in a row for every unique location in the dataset and a column for every date. This could be interesting for time series analysis, but we have opted to instead stick to our initial approach of appending information. This results in one column for the data, i.e. Last_Update.

In [1]:

```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from pyspark.sql.types import DoubleType
from pyspark.sql import functions as F
from pyspark.sql.window import Window
import glob
```

```
import operator

conf = SparkConf().setAppName("test").setMaster("local")
sc = SparkContext(conf=conf)
```

In [2]:

```
spark = SparkSession(sc)
```

We detect how many csv files our dataset has.

In [3]:

```
all_files = glob.glob("csse_covid_19_daily_reports/*.csv")
print("Total amount of csv files:", len(all_files))
all_files.sort()
```

```
Total amount of csv files: 294
```

In [4]:

```
type_1_file_names = []
type_2_file_names = []
type_3_file_names = []
type = 1

for file_name in all_files:
    if file_name == "csse_covid_19_daily_reports/03-22-2020.csv":
        type = 2
    elif file_name == "csse_covid_19_daily_reports/05-29-2020.csv":
        type = 3
    if type == 1:
        type_1_file_names.append(file_name)
    elif type == 2:
        type_2_file_names.append(file_name)
    else:
        type_3_file_names.append(file_name)
```

1.2.1 Preprocessing the initial csv files

In [5]:

```
df_first = spark.read.format("csv").option("header", "true").option("inferschema", "true
").load(type_1_file_names)

df_first.show()
```

```
+--------------+--------------+-------------------+---------+------+---------+---------+
---------+
|Province/State|Country/Region|        Last Update|Confirmed|Deaths|Recovered|Latitude|Lo
ngitude|
+--------------+--------------+-------------------+---------+------+---------+---------+
---------+
|         Hubei|         China|2020-03-21T10:13:08|    67800|  3139|    58946| 30.9756|
112.2707|
|          null|         Italy|2020-03-21T17:43:03|    53578|  4825|     6072| 41.8719|
12.5674|
|          null|         Spain|2020-03-21T13:13:30|    25374|  1375|     2125| 40.4637|
-3.7492|
|          null|       Germany|2020-03-21T20:43:02|    22213|    84|      233| 51.1657|
10.4515|
|          null|          Iran|2020-03-21T11:13:12|    20610|  1556|     7635| 32.4279|
53.688|
|        France|        France|2020-03-21T20:43:02|    14282|   562|       12| 46.2276|
2.2137|
|      New York|            US|2020-03-21T22:43:04|    11710|    60|        0| 42.1657|
-74.9481|
|          null|  Korea, South|2020-03-21T11:13:12|     8799|   102|     1540| 35.9078|
127.7669|
|          null|   Switzerland|2020-03-21T20:43:02|     6575|    75|       15| 46.8182|
```

```
8.2275|
|United Kingdom|United Kingdom|2020-03-21T20:43:03|     5018|   233|        65| 55.3781|
-3.436|
|   Netherlands|   Netherlands|2020-03-21T14:43:04|     3631|   136|         2| 52.1326|
5.2913|
|          null|       Belgium|2020-03-21T11:13:12|     2815|    67|       263| 50.5039|
4.4699|
|          null|       Austria|2020-03-21T14:43:03|     2814|     8|         9| 47.5162|
14.5501|
|          null|        Norway|2020-03-21T17:13:07|     2118|     7|         1|  60.472|
8.4689|
|    Washington|            US|2020-03-21T22:43:04|     1793|    94|         0| 47.4009|-
121.4905|
|          null|        Sweden|2020-03-21T14:43:03|     1763|    20|        16| 60.1282|
18.6435|
|     Guangdong|         China|2020-03-21T12:43:08|     1400|     8|      1325| 23.3417|
113.4244|
|    California|            US|2020-03-21T22:43:04|     1364|    24|         0| 36.1162|-
119.6816|
|    New Jersey|            US|2020-03-21T19:43:03|     1327|    16|         0| 40.2989|
-74.521|
|       Denmark|       Denmark|2020-03-21T12:43:08|     1326|    13|         1| 56.2639|
9.5018|
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
only showing top 20 rows
```

We must appropriately rename the columns for convenient's sake during the appends:

- **Province/State ==> Province_State**
- **Country/Region ==> Country_Region**
- **Latitude** remains the same
- **Longitude** remains the same
- **Last Update ==> Last_Update**

In [6]:

```
df_first = df_first.withColumnRenamed("Province/State", "Province_State").withColumnRenam
ed("Country/Region", "Country_Region").withColumnRenamed("Last Update", "Last_Update")
df_first.show()
```

```
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
|Province_State|Country_Region|        Last_Update|Confirmed|Deaths|Recovered|Latitude|Lo
ngitude|
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
|         Hubei|         China|2020-03-21T10:13:08|    67800|  3139|    58946| 30.9756|
112.2707|
|          null|         Italy|2020-03-21T17:43:03|    53578|  4825|     6072| 41.8719|
12.5674|
|          null|         Spain|2020-03-21T13:13:30|    25374|  1375|     2125| 40.4637|
-3.7492|
|          null|       Germany|2020-03-21T20:43:02|    22213|    84|      233| 51.1657|
10.4515|
|          null|          Iran|2020-03-21T11:13:12|    20610|  1556|     7635| 32.4279|
53.688|
|        France|        France|2020-03-21T20:43:02|    14282|   562|       12| 46.2276|
2.2137|
|      New York|            US|2020-03-21T22:43:04|    11710|    60|        0| 42.1657|
-74.9481|
|          null|  Korea, South|2020-03-21T11:13:12|     8799|   102|     1540| 35.9078|
127.7669|
|          null|   Switzerland|2020-03-21T20:43:02|     6575|    75|       15| 46.8182|
8.2275|
|United Kingdom|United Kingdom|2020-03-21T20:43:03|     5018|   233|       65| 55.3781|
-3.436|
|   Netherlands|   Netherlands|2020-03-21T14:43:04|     3631|   136|        2| 52.1326|
5.2913|
```

```
|          null|       Belgium|2020-03-21T11:13:12|     2815|    67|      263| 50.5039|
4.4699|
|          null|       Austria|2020-03-21T14:43:03|     2814|     8|        9| 47.5162|
14.5501|
|          null|        Norway|2020-03-21T17:13:07|     2118|     7|        1|  60.472|
8.4689|
|    Washington|            US|2020-03-21T22:43:04|     1793|    94|        0| 47.4009|-
121.4905|
|          null|        Sweden|2020-03-21T14:43:03|     1763|    20|       16| 60.1282|
18.6435|
|     Guangdong|         China|2020-03-21T12:43:08|     1400|     8|     1325| 23.3417|
113.4244|
|    California|            US|2020-03-21T22:43:04|     1364|    24|        0| 36.1162|-
119.6816|
|    New Jersey|            US|2020-03-21T19:43:03|     1327|    16|        0| 40.2989|
-74.521|
|       Denmark|       Denmark|2020-03-21T12:43:08|     1326|    13|        1| 56.2639|
9.5018|
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
only showing top 20 rows
```

The entries in the column Last_Update must be parsed correctly to the format that is used in the csv files that have not yet been handled.

There are two formats:

- **e.g.** ***1/22/2020 17:00***
- **e.g.** ***2020-02-25T15:23:04***

These must be converted to the format:

- **e.g.** ***2020-03-23 23:19:34***

In [7]:

```
df_first = df_first.withColumn("Last_Update", F.to_timestamp(df_first["Last_Update"]))
df_first.show()
```

```
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
|Province_State|Country_Region|        Last_Update|Confirmed|Deaths|Recovered|Latitude|Lo
ngitude|
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+
|         Hubei|         China|2020-03-21 10:13:08|    67800|  3139|    58946| 30.9756|
112.2707|
|          null|         Italy|2020-03-21 17:43:03|    53578|  4825|     6072| 41.8719|
12.5674|
|          null|         Spain|2020-03-21 13:13:30|    25374|  1375|     2125| 40.4637|
-3.7492|
|          null|       Germany|2020-03-21 20:43:02|    22213|    84|      233| 51.1657|
10.4515|
|          null|          Iran|2020-03-21 11:13:12|    20610|  1556|     7635| 32.4279|
53.688|
|        France|        France|2020-03-21 20:43:02|    14282|   562|       12| 46.2276|
2.2137|
|      New York|            US|2020-03-21 22:43:04|    11710|    60|        0| 42.1657|
-74.9481|
|          null|  Korea, South|2020-03-21 11:13:12|     8799|   102|     1540| 35.9078|
127.7669|
|          null|   Switzerland|2020-03-21 20:43:02|     6575|    75|       15| 46.8182|
8.2275|
|United Kingdom|United Kingdom|2020-03-21 20:43:03|     5018|   233|       65| 55.3781|
-3.436|
|   Netherlands|   Netherlands|2020-03-21 14:43:04|     3631|   136|        2| 52.1326|
5.2913|
|          null|       Belgium|2020-03-21 11:13:12|     2815|    67|      263| 50.5039|
4.4699|
```

```
|          null|       Austria|2020-03-21 14:43:03|     2814|     8|         9| 47.5162|
14.5501|
|          null|        Norway|2020-03-21 17:13:07|     2118|     7|         1|  60.472|
8.4689|
|    Washington|            US|2020-03-21 22:43:04|     1793|    94|         0| 47.4009|-
121.4905|
|          null|        Sweden|2020-03-21 14:43:03|     1763|    20|        16| 60.1282|
18.6435|
|     Guangdong|         China|2020-03-21 12:43:08|     1400|     8|      1325| 23.3417|
113.4244|
|    California|            US|2020-03-21 22:43:04|     1364|    24|         0| 36.1162|-
119.6816|
|    New Jersey|            US|2020-03-21 19:43:03|     1327|    16|         0| 40.2989|
-74.521|
|       Denmark|       Denmark|2020-03-21 12:43:08|     1326|    13|         1| 56.2639|
9.5018|
+--------------+--------------+-------------------+---------+------+----------+--------+
---------+
only showing top 20 rows
```

Lastly, the remaining csv files contain the column Active being the active cases at the time of the Last_Update and Case_Fatality_Ratio being the ratio of deaths per cases so far. We calculate the Active values for the first csv files by subtracting Deaths and Recovered from Confirmed. Case_Fatality_Ratio is then calculated by dividing Deaths by Confirmed.

In [8]:

```
df_first = df_first.withColumn("Active", df_first["Confirmed"] - df_first["Deaths"] - df_
first["Recovered"])
df_first = df_first.withColumn("Case_Fatality_Ratio", df_first["Deaths"]/df_first["Confir
med"])
df_first.show()
```

```
+--------------+--------------+-------------------+---------+------+----------+--------+
---------+-------+--------------------+
|Province_State|Country_Region|        Last_Update|Confirmed|Deaths|Recovered|Latitude|Lo
ngitude| Active| Case_Fatality_Ratio|
+--------------+--------------+-------------------+---------+------+----------+--------+
---------+-------+--------------------+
|         Hubei|         China|2020-03-21 10:13:08|    67800|  3139|     58946| 30.9756|
112.2707| 5715.0|0.046297935103244835|
|          null|         Italy|2020-03-21 17:43:03|    53578|  4825|      6072| 41.8719|
12.5674|42681.0|  0.0900556198439658|
|          null|         Spain|2020-03-21 13:13:30|    25374|  1375|      2125| 40.4637|
-3.7492|21874.0| 0.05418932765823284|
|          null|       Germany|2020-03-21 20:43:02|    22213|    84|       233| 51.1657|
10.4515|21896.0|0.003781569351280...|
|          null|          Iran|2020-03-21 11:13:12|    20610|  1556|      7635| 32.4279|
53.688|11419.0|  0.0754973313925279|
|        France|        France|2020-03-21 20:43:02|    14282|   562|        12| 46.2276|
2.2137|13708.0| 0.03935023106007562|
|      New York|            US|2020-03-21 22:43:04|    11710|    60|         0| 42.1657|
-74.9481|11650.0|0.005123825789923143|
|          null|  Korea, South|2020-03-21 11:13:12|     8799|   102|      1540| 35.9078|
127.7669| 7157.0|0.011592226389362428|
|          null|   Switzerland|2020-03-21 20:43:02|     6575|    75|        15| 46.8182|
8.2275| 6485.0|0.011406844106463879|
|United Kingdom|United Kingdom|2020-03-21 20:43:03|     5018|   233|        65| 55.3781|
-3.436| 4720.0|0.046432841769629335|
|   Netherlands|   Netherlands|2020-03-21 14:43:04|     3631|   136|         2| 52.1326|
5.2913| 3493.0| 0.03745524648857064|
|          null|       Belgium|2020-03-21 11:13:12|     2815|    67|       263| 50.5039|
4.4699| 2485.0|0.023801065719360567|
|          null|       Austria|2020-03-21 14:43:03|     2814|     8|         9| 47.5162|
14.5501| 2797.0|0.002842928216062...|
|          null|        Norway|2020-03-21 17:13:07|     2118|     7|         1|  60.472|
8.4689| 2110.0|0.003305004721435...|
|    Washington|            US|2020-03-21 22:43:04|     1793|    94|         0| 47.4009|-
121.4905| 1699.0| 0.05242610150585611|
|          null|        Sweden|2020-03-21 14:43:03|     1763|    20|        16| 60.1282|
```

```
18.6435| 1727.0|0.011344299489506523|
|     Guangdong|         China|2020-03-21 12:43:08|     1400|      8|     1325| 23.3417|
113.4244|   67.0|0.005714285714285714|
|    California|            US|2020-03-21 22:43:04|     1364|     24|        0| 36.1162|-
119.6816| 1340.0|0.017595307917888565|
|    New Jersey|            US|2020-03-21 19:43:03|     1327|     16|        0| 40.2989|
-74.521| 1311.0|0.012057272042200452|
|       Denmark|       Denmark|2020-03-21 12:43:08|     1326|     13|        1| 56.2639|
9.5018| 1312.0| 0.00980392156862745|
+--------------+--------------+-------------------+---------+------+---------+--------+
---------+-------+--------------------+
only showing top 20 rows
```

1.2.2 Preprocessing the remaining csv files

We then append all the remaining csv files into one dataframe. It should be noted that the columns Incidence_rate and Case_Fatality_Ratio have only been added to the dataset since the 29th of May 2020. The second column can be calculated using the given data but since the population per country is not given, the first cannot. Therefore, if the column Incidence_rate is needed, only entries between May 29 and the current day will be heald into account.

In [9]:

```
df_second = spark.read.format("csv").option("header", "true").option("inferschema", "tru
e").load(type_2_file_names)
df_second.show()
```

```
+-----+---------+--------------+--------------+-------------------+-----------------+--
----------------+---------+------+---------+------+--------------------+
| FIPS|   Admin2|Province_State|Country_Region|        Last_Update|              Lat|
Long_|Confirmed|Deaths|Recovered|Active|        Combined_Key|
+-----+---------+--------------+--------------+-------------------+-----------------+--
----------------+---------+------+---------+------+--------------------+
|45001|Abbeville|South Carolina|            US|2020-05-29 02:32:50|      34.22333378|
-82.46170658|       37|     0|        0|    37|Abbeville, South ...|
|22001|   Acadia|     Louisiana|            US|2020-05-29 02:32:50|       30.2950649|
-92.41419698|      401|    22|        0|   379|Acadia, Louisiana...|
|51001| Accomack|      Virginia|            US|2020-05-29 02:32:50|      37.76707161|
-75.63234615|      807|    12|        0|   795|Accomack, Virgini...|
|16001|      Ada|         Idaho|            US|2020-05-29 02:32:50|       43.4526575|-11
6.24155159999998|      803|    22|        0|   781|      Ada, Idaho, US|
|19001|    Adair|          Iowa|            US|2020-05-29 02:32:50|      41.33075609|
-94.47105874|        8|     0|        0|     8|     Adair, Iowa, US|
|21001|    Adair|      Kentucky|            US|2020-05-29 02:32:50|      37.10459774|
-85.28129668|       96|    19|        0|    77| Adair, Kentucky, US|
|29001|    Adair|      Missouri|            US|2020-05-29 02:32:50|      40.19058551|
-92.60078167|       49|     0|        0|    49| Adair, Missouri, US|
|40001|    Adair|      Oklahoma|            US|2020-05-29 02:32:50|      35.88494195|
-94.65859267|       84|     3|        0|    81| Adair, Oklahoma, US|
| 8001|    Adams|      Colorado|            US|2020-05-29 02:32:50|      39.87432092|
-104.3362578|     3070|   118|        0|  2952| Adams, Colorado, US|
|16003|    Adams|         Idaho|            US|2020-05-29 02:32:50|      44.89333571|
-116.4545247|        3|     0|        0|     3|    Adams, Idaho, US|
|17001|    Adams|      Illinois|            US|2020-05-29 02:32:50|      39.98815591|
-91.18786813|       44|     1|        0|    43| Adams, Illinois, US|
|18001|    Adams|       Indiana|            US|2020-05-29 02:32:50|       40.7457653|
-84.93671406|       13|     1|        0|    12|  Adams, Indiana, US|
|19003|    Adams|          Iowa|            US|2020-05-29 02:32:50|      41.02903567|
-94.69932645|        7|     0|        0|     7|     Adams, Iowa, US|
|28001|    Adams|   Mississippi|            US|2020-05-29 02:32:50|      31.47669768|
-91.35326037|      190|    15|        0|   175|Adams, Mississipp...|
|31001|    Adams|      Nebraska|            US|2020-05-29 02:32:50|40.52449420000001|
-98.50117804|      265|    11|        0|   254| Adams, Nebraska, US|
|39001|    Adams|          Ohio|            US|2020-05-29 02:32:50|      38.84541072|
-83.4718964|        8|     1|        0|     7|     Adams, Ohio, US|
|42001|    Adams|  Pennsylvania|            US|2020-05-29 02:32:50|      39.87140411|
-77.21610347|      240|     7|        0|   233|Adams, Pennsylvan...|
|53001|    Adams|    Washington|            US|2020-05-29 02:32:50|      46.98299757|
```

```
-118.5601734|       54|     0|        0|    54|Adams, Washington...|
|55001|    Adams|     Wisconsin|            US|2020-05-29 02:32:50|      43.96974651|
-89.76782777|        4|     1|        0|     3|Adams, Wisconsin, US|
|50001|  Addison|       Vermont|            US|2020-05-29 02:32:50|      44.03217337|
-73.14130877|       62|     2|        0|    60|Addison, Vermont, US|
+-----+---------+--------------+--------------+--------------------+-----------------+--
-----------------+---------+------+---------+------+--------------------+
only showing top 20 rows
```

First the column Case-Fatality_Ratio is calculated and added to these entries, being entries between March 20th and May 28th.

In [10]:

```
df_second = df_second.withColumn("Case_Fatality_Ratio", df_second["Deaths"]/df_second["Co
nfirmed"])
df_second.show()
```

```
+-----+---------+--------------+--------------+--------------------+-----------------+--
-----------------+---------+------+---------+------+--------------------+-------------
------+
| FIPS|   Admin2|Province_State|Country_Region|        Last_Update|              Lat|
Long_|Confirmed|Deaths|Recovered|Active|        Combined_Key| Case_Fatality_Ratio|
+-----+---------+--------------+--------------+--------------------+-----------------+--
-----------------+---------+------+---------+------+--------------------+-------------
------+
|45001|Abbeville|South Carolina|            US|2020-05-29 02:32:50|      34.22333378|
-82.46170658|       37|     0|        0|    37|Abbeville, South ...|                0.0
|
|22001|   Acadia|     Louisiana|            US|2020-05-29 02:32:50|       30.2950649|
-92.41419698|      401|    22|        0|   379|Acadia, Louisiana...| 0.05486284289276808|
|51001| Accomack|      Virginia|            US|2020-05-29 02:32:50|      37.76707161|
-75.63234615|      807|    12|        0|   795|Accomack, Virgini...| 0.01486988847583643|
|16001|      Ada|         Idaho|            US|2020-05-29 02:32:50|       43.4526575|-11
6.24155159999998|      803|    22|        0|   781|      Ada, Idaho, US|  0.027397260273
9726|
|19001|    Adair|          Iowa|            US|2020-05-29 02:32:50|      41.33075609|
-94.47105874|        8|     0|        0|     8|     Adair, Iowa, US|                0.0
|
|21001|    Adair|      Kentucky|            US|2020-05-29 02:32:50|      37.10459774|
-85.28129668|       96|    19|        0|    77| Adair, Kentucky, US| 0.19791666666666666|
|29001|    Adair|      Missouri|            US|2020-05-29 02:32:50|      40.19058551|
-92.60078167|       49|     0|        0|    49| Adair, Missouri, US|                0.0
|
|40001|    Adair|      Oklahoma|            US|2020-05-29 02:32:50|      35.88494195|
-94.65859267|       84|     3|        0|    81| Adair, Oklahoma, US| 0.03571428571428571|
| 8001|    Adams|      Colorado|            US|2020-05-29 02:32:50|      39.87432092|
-104.3362578|     3070|   118|        0|  2952| Adams, Colorado, US|0.038436482084690554|
|16003|    Adams|         Idaho|            US|2020-05-29 02:32:50|      44.89333571|
-116.4545247|        3|     0|        0|     3|    Adams, Idaho, US|                0.0
|
|17001|    Adams|      Illinois|            US|2020-05-29 02:32:50|      39.98815591|
-91.18786813|       44|     1|        0|    43| Adams, Illinois, US|0.022727272727272728|
|18001|    Adams|       Indiana|            US|2020-05-29 02:32:50|       40.7457653|
-84.93671406|       13|     1|        0|    12|  Adams, Indiana, US| 0.07692307692307693|
|19003|    Adams|          Iowa|            US|2020-05-29 02:32:50|      41.02903567|
-94.69932645|        7|     0|        0|     7|     Adams, Iowa, US|                0.0
|
|28001|    Adams|   Mississippi|            US|2020-05-29 02:32:50|      31.47669768|
-91.35326037|      190|    15|        0|   175|Adams, Mississipp...| 0.07894736842105263|
|31001|    Adams|      Nebraska|            US|2020-05-29 02:32:50|40.52449420000001|
-98.50117804|      265|    11|        0|   254| Adams, Nebraska, US| 0.04150943396226415|
|39001|    Adams|          Ohio|            US|2020-05-29 02:32:50|      38.84541072|
-83.4718964|        8|     1|        0|     7|     Adams, Ohio, US|               0.125|
|42001|    Adams|  Pennsylvania|            US|2020-05-29 02:32:50|      39.87140411|
-77.21610347|      240|     7|        0|   233|Adams, Pennsylvan...|0.029166666666666667|
|53001|    Adams|    Washington|            US|2020-05-29 02:32:50|      46.98299757|
-118.5601734|       54|     0|        0|    54|Adams, Washington...|                0.0
|
|55001|    Adams|     Wisconsin|            US|2020-05-29 02:32:50|      43.96974651|
```

```
-89.76782777|        4|     1|        0|     3|Adams, Wisconsin, US|                0.25
|
|50001|  Addison|       Vermont|            US|2020-05-29 02:32:50|      44.03217337|
-73.14130877|       62|     2|        0|    60|Addison, Vermont, US| 0.03225806451612903|
+-----+---------+--------------+--------------+-------------------+-----------------+--
-----------------+---------+------+---------+------+--------------------+-------------
------+
only showing top 20 rows
```

The last type of csv files is then loaded into a dataframe.

In [11]:

```
df_third = spark.read.format("csv").option("header", "true").option("inferschema", "true
").load(type_3_file_names)
df_third.show()
```

```
+----+------+------------------+------------------+-------------------+---------+---
-------+---------+------+---------+------+--------------------+-----------------+-----
--------------+
|FIPS|Admin2|      Province_State|     Country_Region|        Last_Update|      Lat|
Long_|Confirmed|Deaths|Recovered|Active|        Combined_Key|     Incident_Rate|Case_Fata
lity_Ratio|
+----+------+------------------+------------------+-------------------+---------+---
-------+---------+------+---------+------+--------------------+-----------------+-----
--------------+
|null|  null|              null|       Afghanistan|2020-11-11 05:25:30| 33.93911| 67.
709953|    42463|  1577|    34954|  5932|         Afghanistan|109.07991172806463| 3.7138
214445517272|
|null|  null|              null|           Albania|2020-11-11 05:25:30|  41.1533|   2
0.1683|    25294|   579|    12353| 12362|             Albania| 878.9352977969282| 2.2890
804143275085|
|null|  null|              null|           Algeria|2020-11-11 05:25:30|  28.0339|
1.6596|    63446|  2077|    42626| 18743|             Algeria| 144.6852700858221| 3.2736
500330990133|
|null|  null|              null|           Andorra|2020-11-11 05:25:30|  42.5063|
1.5218|     5477|    75|     4405|   997|             Andorra|7088.5912120623825| 1.3693
627898484573|
|null|  null|              null|            Angola|2020-11-11 05:25:30| -11.2027|   1
7.8739|    12816|   308|     6036|  6472|              Angola| 38.99438780210762|  2.403
245942571785|
|null|  null|              null|Antigua and Barbuda|2020-11-11 05:25:30|  17.0608|  -61
.7964|      131|     3|      122|     6| Antigua and Barbuda|133.77175067396453| 2.290076
3358778624|
|null|  null|              null|         Argentina|2020-11-11 05:25:30| -38.4161|  -6
3.6167|  1262476| 34183|  1081897|146396|           Argentina| 2793.349475991972|  2.7076
15827944452|
|null|  null|              null|           Armenia|2020-11-11 05:25:30|  40.0691|   4
5.0382|   108687|  1609|    66835| 40243|             Armenia| 3667.850733354167| 1.48039
78396680375|
|null|  null|Australian Capita...|         Australia|2020-11-11 05:25:30| -35.4735|  149
.0124|      114|     3|      111|     0|Australian Capita...|26.629292221443592| 2.631578
9473684212|
|null|  null|     New South Wales|         Australia|2020-11-11 05:25:30| -33.8688|  151
.2093|     4469|    53|     3156|  1260|New South Wales, ...|55.050505050505045| 1.185947
6392929067|
|null|  null|  Northern Territory|         Australia|2020-11-11 05:25:30| -12.4634|  130
.8456|       41|     0|       33|     8|Northern Territor...|16.693811074918568|
0.0|
|null|  null|          Queensland|         Australia|2020-11-11 05:25:30| -27.4698|  15
3.0251|     1179|     6|     1163|    10|Queensland, Austr...| 23.04760043006549| 0.50890
58524173028|
|null|  null|     South Australia|         Australia|2020-11-11 05:25:30| -34.9285|  138
.6007|      517|     4|      495|    18|South Australia, ...|29.433532593225163| 0.773694
3907156673|
|null|  null|            Tasmania|         Australia|2020-11-11 05:25:30| -42.8821|  14
7.3272|      230|    13|      217|     0| Tasmania, Australia| 42.95051353874883| 5.65217
39130434785|
|null|  null|            Victoria|         Australia|2020-11-11 05:25:30| -37.8136|  14
4.9631|    20345|   819|    19522|     4| Victoria, Australia| 306.8673735652121|   4.025
```

```
55910543131|
|null|  null|   Western Australia|          Australia|2020-11-11 05:25:30| -31.9505|  115
.8605|      776|     9|      757|    10|Western Australia...|29.498973618185964| 1.159793
8144329898|
|null|  null|                null|            Austria|2020-11-11 05:25:30|  47.5162|   1
4.5501|   164866|  1499|    98663| 64704|             Austria|  1830.54272517321| 0.90922
32479710796|
|null|  null|                null|         Azerbaijan|2020-11-11 05:25:30|  40.1431|   4
7.5769|    67392|   867|    50009| 16516|          Azerbaijan|  664.669462752147|  1.286
502849002849|
|null|  null|                null|            Bahamas|2020-11-11 05:25:30|25.025885|-78.
035889|     7012|   154|     5035|  1823|             Bahamas|1783.0987061599806| 2.1962
350256702794|
|null|  null|                null|            Bahrain|2020-11-11 05:25:30|  26.0275|
50.55|    83811|   331|    81415|  2065|             Bahrain| 4925.472339580261|0.394936
22555511804|
+----+------+--------------------+-------------------+-------------------+---------+---
-------+---------+------+---------+------+--------------------+------------------+-----
--------------+
only showing top 20 rows
```

Only the columns listed in the first type of dataframe, excluding Longitude and Latitude, and Incidence_Rate from the third type are kept. A copy of the third dataframe is kept for when Incidence_Rate is needed.

In [12]:

```
df_first = df_first.select("Province_State", "Country_Region", "Last_Update", "Confirmed"
, "Deaths", "Recovered", "Active", "Case_Fatality_Ratio")
df_second = df_second.select("Province_State", "Country_Region", "Last_Update", "Confirme
d", "Deaths", "Recovered", "Active", "Case_Fatality_Ratio")
df_third_2 = df_third.select("Province_State", "Country_Region", "Last_Update", "Confirme
d", "Deaths", "Recovered", "Active", "Case_Fatality_Ratio")
df_with_incidence = df_third.select("Province_State", "Country_Region", "Last_Update", "
Confirmed", "Deaths", "Recovered", "Active", "Case_Fatality_Ratio", "Incident_Rate")
```

The last step of the preprocessing is combining all the dataframes to form a single one. We also rename 3 main countries such that we can later correctly access their data during a merge.

In [13]:

```
_df = df_first.union(df_second)
df = _df.union(df_third_2)

df = df.withColumn("Country_Region", F.when(df["Country_Region"] == "Mainland China", "C
hina").otherwise(df["Country_Region"]))
df = df.withColumn("Country_Region", F.when(df["Country_Region"] == "Korea, South", "Sou
th Korea").otherwise(df["Country_Region"]))
df = df.withColumn("Country_Region", F.when(df["Country_Region"] == "UK", "United Kingdo
m").otherwise(df["Country_Region"]))

df.show()
```

```
+--------------+--------------+-------------------+---------+------+---------+-------+-
--------------------+
|Province_State|Country_Region|        Last_Update|Confirmed|Deaths|Recovered| Active| Ca
se_Fatality_Ratio|
+--------------+--------------+-------------------+---------+------+---------+-------+-
--------------------+
|         Hubei|         China|2020-03-21 10:13:08|    67800|  3139|    58946| 5715.0|0.
046297935103244835|
|          null|         Italy|2020-03-21 17:43:03|    53578|  4825|     6072|42681.0|
0.0900556198439658|
|          null|         Spain|2020-03-21 13:13:30|    25374|  1375|     2125|21874.0| 0
.05418932765823284|
|          null|       Germany|2020-03-21 20:43:02|    22213|    84|      233|21896.0|0.
003781569351280...|
|          null|          Iran|2020-03-21 11:13:12|    20610|  1556|     7635|11419.0|
0.0754973313925279|
|        France|        France|2020-03-21 20:43:02|    14282|   562|       12|13708.0| 0
.03935023106007562|
```

```
.0395502510600752|
|      New York|            US|2020-03-21 22:43:04|    11710|    60|        0|11650.0|0.
005123825789923143|
|          null|   South Korea|2020-03-21 11:13:12|     8799|   102|     1540| 7157.0|0.
011592226389362428|
|          null|   Switzerland|2020-03-21 20:43:02|     6575|    75|       15| 6485.0|0.
011406844106463879|
|United Kingdom|United Kingdom|2020-03-21 20:43:03|     5018|   233|       65| 4720.0|0.0
46432841769629335|
|   Netherlands|   Netherlands|2020-03-21 14:43:04|     3631|   136|        2| 3493.0| 0.
03745524648857064|
|          null|       Belgium|2020-03-21 11:13:12|     2815|    67|      263| 2485.0|0.
023801065719360567|
|          null|       Austria|2020-03-21 14:43:03|     2814|     8|        9| 2797.0|0.
002842928216062...|
|          null|        Norway|2020-03-21 17:13:07|     2118|     7|        1| 2110.0|0.
003305004721435...|
|    Washington|            US|2020-03-21 22:43:04|     1793|    94|        0| 1699.0| 0
.05242610150585611|
|          null|        Sweden|2020-03-21 14:43:03|     1763|    20|       16| 1727.0|0.
011344299489506523|
|     Guangdong|         China|2020-03-21 12:43:08|     1400|     8|     1325|   67.0|0.
005714285714285714|
|    California|            US|2020-03-21 22:43:04|     1364|    24|        0| 1340.0|0.
017595307917888565|
|    New Jersey|            US|2020-03-21 19:43:03|     1327|    16|        0| 1311.0|0.
012057272042200452|
|       Denmark|       Denmark|2020-03-21 12:43:08|     1326|    13|        1| 1312.0| 0.
00980392156862745|
+--------------+--------------+-------------------+---------+------+---------+-------+-
------------------+
only showing top 20 rows
```

1.3 Assignment

We are to create a dashboard (textually, so not graphs) from our data. There are several assignments that we must complete:

- **We must track which countries are doing a good job of tracking cases (deaths/cases ratio). (section 1.3.1)**
- **We must determine which countries have good healthcare. (section 1.3.2)**
- **We must detect which countries are doing a good job of containing outbreaks (incidence). (section 1.3.3)**
- **We must analyze whether the situation between the final entry for a country and 7 days prior has improved or declined, based on a previous indicator. In this case, we have chosen to do so based on indidence, i.e. how good are they managing indicence vs. 7 days ago. (section 1.3.4)**

For each assignment, we will look at the 10 best and 10 worst countries. It is important to remain critical. Certain countries may appear in these top-10s due to underreported or statistically irrelevant figures. Thus, we will access these top 10s and make adjustments if needed.

1.3.1 Quality of tracking (deaths per cases ratio)

In this section, we determine the average and standard deviation of the *Case Fatality Ratio*. We will use both these metrics in our analysis of which countries are doing a good job of tracking cases and which are doing a poor job.

During EDA, we noticed that certain _Country *Regions* reported more than others. Thus, we will count the number of *updates* they have made, as well as how many _Province *States* each country has. We divide these into each other and what we are left with is a *Count* figure that tells us how frequently each country has made updates. The idea here is to detect which countries could be considered outliers that should be omitted. Indeed, the table below immediately shows us that certain _Country*Regions* have very low counts. One reason could be that they were originally counted separately, such as North Ireland, but were later incorporated into the figures of another _Country*Region*. We set a cut-off at 100 days, meaning every country with less than 100 updates (daily) will not be considered. Exploration of the excluded countries showed that these were indeed regions such as North Korea, Northern Ireland, Eritrea and Jersey. Given the low amount of information on these regions, we

cannot perform meaningful analysis on them. Thus, we excluded them.

To determine whether a country has good or poor tracking of the virus, we will use the standard deviation. A low standard deviation of the _Case_Fatality*Ratio* tells us that a country is good at tracking its cases: the deaths and cases grow (or decline) proportionally. A high standard deviation tells us that _Case_Fatality*Ratio* that there may be descrepancy between the deaths and cases, i.e. either underreporting or insufficient tracking of the virus' spread. One can easily visualize this by imagining that the increase in the number of detected cases remains stable, though the deaths grow fast. This may indicate to us that we are not suffiently testing people and discovering the true infection spread of our population.

In [14]:

```
from pyspark.sql.types import DoubleType, IntegerType

countries_dev_df = df.groupBy("Country_Region").agg(F.stddev(df.Case_Fatality_Ratio).ali
as('Stdev_Case_Fat'), F.count(df["Last_Update"]).alias("#_Updates"), F.countDistinct(df[
"Province_State"]).alias("#_Province"))

df = df.withColumn("Case_Fatality_Ratio", df["Case_Fatality_Ratio"].cast(DoubleType()))
countries_avg_df = df.groupBy("Country_Region").avg('Case_Fatality_Ratio')
countries_avg_df = countries_avg_df.withColumnRenamed("avg(Case_Fatality_Ratio)","Avg_Cas
e_Fat")

countries_df = countries_dev_df.join(countries_avg_df, "Country_Region")

countries_df = countries_df.withColumn("Count", F.when(countries_df["#_Province"] == 0,
countries_df["#_Updates"]).otherwise((countries_df["#_Updates"]/countries_df["#_Province
"]).cast(IntegerType())))

countries_df = countries_df.select('Country_Region', 'Stdev_Case_Fat', 'Avg_Case_Fat', '
Count')

countries_df.show()
```

```
+----------------+--------------------+--------------------+-----+
|  Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|
+----------------+--------------------+--------------------+-----+
|            Chad|  3.5695689858751494|   5.411630859189694|  237|
|        Paraguay|  0.8293239417888044|   1.005183763421402|  248|
|          Russia|   0.946508661445475|  1.2360474017945295|  164|
|   North Ireland|                 NaN|                 0.0|    1|
|           Yemen|  11.775450297939296|   21.52944755078509|  215|
|         Senegal|  0.9284426424828336|  1.2310155662310345|  254|
|      Cabo Verde|  0.4483307510272256|  0.7352040231301348|  236|
|          Sweden|  2.6224330317831352|   5.803851660766782|  164|
|Republic of Korea|                 NaN|0.007187541594569413|    1|
|          Guyana|  2.3892527692014673|  2.9850920146852618|  244|
|           Burma|  0.9446288145652882|   1.367181542886264|  229|
|         Eritrea|                 0.0|                 0.0|  235|
|          Jersey|                 0.0|                 0.0|    8|
|     Philippines|  1.3785307059296326|  1.4803186632474137|  283|
|        Djibouti|  0.5038305666274472|  0.7495083045927876|  238|
|        Malaysia|  0.6364586998732064|   0.736008707826929|  283|
|       Singapore|0.025860148802332693|0.031175630965621773|  283|
|            Fiji|  2.8814165663247078|  2.4249389413147795|  237|
|          Turkey|  1.1925938861856236|  1.7397240163174068|  245|
|          Malawi|  1.2911812326485566|  1.8685838469426101|  223|
+----------------+--------------------+--------------------+-----+
only showing top 20 rows
```

Below is a sorted table by *Count*. Recall, this column tells us how frequently this region has reported CoViD-19 cases. The Netherlands has made 144 daily reports, is a 1st world country and should thus be included in the analysis. All regions with a lower *Count* than the Netherlands, however, may lack sufficient frequent reporting to be statistically viable. The jump between the region before The Netherlands, i.e. Taiwan (with a mere 37 daily reports), is significant. Although we could have chosen 144 as our cut-off, this would be fairly arbitrary, especially as we add new daily data. Thus, we chose a nearby round number: 100.

In [15]:

```
countries_df.orderBy(countries_df["Count"]).show(100)
```

```
+--------------------+--------------------+--------------------+-----+
|      Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|
+--------------------+--------------------+--------------------+-----+
|         Ivory Coast|                null|                null|    0|
|Iran (Islamic Rep...|                 NaN| 0.03618502859985078|    1|
|       North Ireland|                 NaN|                 0.0|    1|
|          St. Martin|                 NaN|                 0.0|    1|
|          Azerbaijan|                 NaN|                 0.0|    1|
| Republic of Ireland|                 NaN|                 0.0|    1|
|   Republic of Korea|                 NaN|0.007187541594569413|    1|
|            Viet Nam|                 NaN|                 0.0|    1|
|     Channel Islands|                 NaN|                 0.0|    1|
|       Hong Kong SAR|                 NaN|               0.025|    1|
|           Macao SAR|                 NaN|                 0.0|    1|
|          Cape Verde|                 NaN|                 0.0|    1|
| Republic of Moldova|                 NaN|                 0.0|    1|
|  Russian Federation|                 NaN|                 0.0|    1|
|        Saint Martin|                 NaN|                 0.0|    1|
| Taipei and environs|                 NaN| 0.02127659574468085|    1|
|             Vanuatu|                 NaN|                 0.0|    1|
|          East Timor|                 NaN|                 0.0|    1|
|             Curacao|                 0.0|                 0.0|    2|
|        Bahamas, The|                 0.0|                 0.0|    3|
|      Cayman Islands|                 0.0|                 0.0|    3|
|        Vatican City|                 0.0|                 0.0|    4|
|         Gambia, The|                 0.0|                 0.0|    4|
|          The Gambia|                 NaN|                 0.0|    5|
|           Palestine|                 0.0|                 0.0|    5|
|                Guam|                 0.0|                 0.0|    6|
|           Greenland|                 0.0|                 0.0|    6|
|         The Bahamas|                 0.0|                 0.0|    6|
|Republic of the C...|                 NaN|                 0.0|    6|
|         Puerto Rico|                 NaN|                 0.0|    6|
|             Mayotte|                 0.0|                 0.0|    6|
|           Gibraltar|                 0.0|                 0.0|    7|
|       Faroe Islands|                 0.0|                 0.0|    7|
|occupied Palestin...|                 NaN|                 0.0|    7|
|    Saint Barthelemy|                 0.0|                 0.0|    7|
|               Aruba|                 0.0|                 0.0|    7|
|            Guernsey|                 0.0|                 0.0|    8|
|              Jersey|                 0.0|                 0.0|    8|
|          Guadeloupe|                 0.0|                 0.0|    9|
|      Czech Republic|                 0.0|                 0.0|   10|
|             Reunion|                 0.0|                 0.0|   11|
|         Cruise Ship|5.546507433399602E-4|0.010277272772429688|   13|
|       French Guiana|                 0.0|                 0.0|   14|
|    Marshall Islands|                 0.0|                 0.0|   14|
|          Martinique|0.025971468598461826| 0.01918510043224002|   15|
|              Others|0.003809275792086...|0.004135684676654204|   16|
|     Solomon Islands|                 0.0|                 0.0|   30|
|           Hong Kong|0.011643016005496932|0.023763030045685748|   37|
|               Macau|                 0.0|                 0.0|   37|
|              Taiwan|0.017182824823257177|  0.0192226584846501|   37|
|         Netherlands|    4.64978084432237|   4.803770956736284|  144|
|               India|    1.02634796800188|   1.199112105381121|  152|
|              Sweden|  2.6224330317831352|   5.803851660766782|  164|
|              Russia|   0.946508661445475|  1.2360474017945295|  164|
|             Ukraine|  1.0152665663177516|  2.1716710393971463|  166|
|            Colombia|   2.489603356893213|  3.2322042842000718|  167|
|               Japan|   2.672323311753206|  2.2765792521859733|  168|
|            Pakistan|  0.9434647612159119|  1.8934791664812924|  169|
|                Peru|   7.625936089962196|   3.987898312258953|  170|
|              Brazil|  1.6908518459160304|   2.732003489243206|  172|
|             Germany|   1.764620106625167|   3.018032296259767|  176|
|              Canada|   4.287902401221546|  1.7675164542669943|  176|
|              Mexico|   4.747268837379202|  10.408207243501932|  177|
|               Chile|  0.8390076024348666|  1.4613295502588748|  179|
|      United Kingdom|    4.83059583471635|  4.0362135494518885|  180|
|             Lesotho|  1.1825413118771857|  1.6238345304026895|  182|
```

```
|               Spain|   4.571133928549751|    6.07453437237955| 185|
|               Italy|   4.638040583002451|    8.08262907548739| 185|
|          Tajikistan| 0.30682960154604466|  0.7156363872447318| 195|
|             Comoros|  0.6773509392670114|   1.453387115650349| 195|
|           Australia|  1.6321824710641906|  1.1541926466004464| 197|
|               Yemen|  11.775450297939296|   21.52944755078509| 215|
|Sao Tome and Prin...|  0.7711777169544275|  1.3409364472316792| 219|
|      Western Sahara|  4.3932166852436865|   7.668181818181818| 220|
|         South Sudan|    0.81394128436488|  1.3779962652591853| 220|
|              France|   3.232075505707301|  1.5677666032705693| 222|
|              Malawi|  1.2911812326485566|  1.8685838469426101| 223|
|        Sierra Leone|  1.6739403772933732|  2.7188416610058663| 225|
|             Burundi| 0.37891604187011646| 0.32965066778410557| 225|
|            Botswana|  0.6189911342640264| 0.48427261666622073| 226|
|          MS Zaandam|    9.81695905684611 8|  16.235867446247564| 228|
|               Burma|  0.9446288145652882|    1.36718154288626 4| 229|
|  West Bank and Gaza| 0.33422986011119876|  0.4917106376959971| 230|
|Saint Kitts and N...|                 0.0|                 0.0| 231|
|    Diamond Princess|  0.8150778436437957|    1.31691108519359| 231|
|       Guinea-Bissau|  0.6980354111041228|   1.043974442668927| 231|
|                Mali|   2.159956821315007|   3.394416066268014| 231|
|                Laos|                 0.0|                 0.0| 232|
|               Libya|  0.9844895651116131|   1.408975302934538| 232|
|              Belize|   3.309778273300614|  2.7398813632776937| 233|
|             Bahamas|  3.8637210144927345|    3.41110412725333| 234|
|             Grenada|                 0.0|                 0.0| 234|
|          Mozambique|  0.3090379782498638| 0.47058102935681173| 234|
|              Gambia|  1.5898637266201248|   2.413470934627354| 234|
|            Dominica|                 0.0|                 0.0| 234|
|         Timor-Leste|                 0.0|                 0.0| 234|
|               Syria|   2.068820152144134|  3.1632871802896236| 234|
|              Uganda| 0.46513078895971216| 0.40204270435569534| 235|
|             Eritrea|                 0.0|                 0.0| 235|
|          Madagascar|  0.5661907104876923|  0.8063520765661805| 236|
+--------------------+--------------------+--------------------+-----+
only showing top 100 rows
```

Keep only the regions that have more than 100 daily reports.

In [16]:

```
countries_df = countries_df.filter(countries_df["Count"]>100)
```

Good tracking

We will now analyse which countries have a low standard deviation so that we can determine the top-10 of countries that are tracking the virus well. Notice that certain regions have a standard deviation (and average, for that matter) of 0, despite their large reporting. This is because these countries have reported 0 deaths. Thus, they are irrelevant to us, because they could be underreporting, not actually sharing truthful information or the region may just be too small to be statistically viable for comparisons.

In [17]:

```
countries_df.filter(countries_df["Stdev_Case_Fat"] == 0).show(40)
```

```
+--------------------+--------------+------------+-----+
|      Country_Region|Stdev_Case_Fat|Avg_Case_Fat|Count|
+--------------------+--------------+------------+-----+
|             Eritrea|           0.0|         0.0|  235|
|            Cambodia|           0.0|         0.0|  283|
|            Dominica|           0.0|         0.0|  234|
|         Timor-Leste|           0.0|         0.0|  234|
|                Laos|           0.0|         0.0|  232|
|              Bhutan|           0.0|         0.0|  250|
|            Holy See|           0.0|         0.0|  246|
|Saint Kitts and N...|           0.0|         0.0|  231|
|Saint Vincent and...|           0.0|         0.0|  242|
```

```
|            Mongolia|           0.0|         0.0|  246|
|             Grenada|           0.0|         0.0|  234|
|          Seychelles|           0.0|         0.0|  242|
+--------------------+--------------+------------+-----+
```

We filter the regions whose standard deviation is 0 out of our dataset, as well as the regions who standard deviation is NaN. Next, we order by the standard deviation.

In [18]:

```
countries_df_filtered = countries_df.filter(countries_df["Stdev_Case_Fat"] > 0).filter(~
F.isnan(countries_df["Stdev_Case_Fat"]))
countries_df_best = countries_df_filtered.orderBy(countries_df_filtered["Stdev_Case_Fat"
])
```

What we are left with are the countries, listed from excellently tracking the virus to poorly tracking the virus, that have a meaningful standard deviation.

There are several things we can notice, before applying a ranking:

- **Certain countries appear as tracking the virus really well. We must remain critical and assess whether this is truly the case. Note that, for example, Rwanda has a mere 41 deaths out of 5312 cases, despite having a large population size (13,066,896). Third world countries with dictorial regimes fall victim to underreporting or an inadequate assessment of the state of CoViD-19 is their countries. That is to say, that we cannot with confidence determine whether these kinds of countries should indeed be included in a list that assessess a country's performance on tracking the virus.**
- **Some countries, such as Saint Lucia, simply have a very low amount of cases and deaths compared to their total population. The country could be doing a really good job, could be underreporting or could be isolated better from other countries. It is difficult to determine this, though due to the fact that Saint Lucia appears to be the oddball in the list, we decided to remove it.**
- **In conjunction with the previous statement, we notice that the top of this list includes many countries with high average temperatures throughout the year, which may allow them to be less vulnerable to the virus. Thus, we must remain critical and assess whether they are at the top of this list due to good tracking or being less vulnerable to the virus. This is difficult to determine based on this dataset solely, so we decided to keep most of these countries on this list.**
- **Many of these countries are islands! This shows that island regions have their isolation factor has beneficiary to their exposure to CoViD-19, although that is not to say that they are immune to the virus. This merely tells us that influx of other countries' citizens is much more easy to control versus countries on the mainland that have physical borders that can be crossed.**

In [19]:

```
countries_df_best.show(25)
```

```
+--------------------+--------------------+--------------------+-----+
|      Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|
+--------------------+--------------------+--------------------+-----+
|           Singapore|0.025860148802332693|0.031175630965621773|  283|
|         Saint Lucia| 0.04343407746846115|0.002792048246593701|  242|
|               Qatar| 0.07525506314186811| 0.09706676018345682|  256|
|             Bahrain| 0.16734822061392865|  0.2141036358459203|  261|
|            Maldives|  0.1824294888247625| 0.25205215911179063|  248|
|           Sri Lanka| 0.21515182234282237| 0.23949510358940435|  283|
|             Iceland| 0.23131809992516325| 0.29552475769697606|  257|
|              Rwanda| 0.24384120413939261| 0.29868943036880685|  242|
|               Nepal| 0.24846209676491823| 0.25624908951815933|  283|
|               Ghana|  0.2736646093311702|   0.407322033696884|  242|
|United Arab Emirates| 0.27890160410244846|  0.3195480876213416|  283|
|              Guinea|  0.2816915090558841|   0.4148580788870 33|  243|
|               Gabon| 0.29540273001242784|  0.4656401966131654|  242|
|          Tajikistan| 0.30682960154604466|  0.7156363872447318|  195|
|          Mozambique|  0.3090379782498638| 0.47058102935681173|  234|
|              Kuwait|  0.3271174474480425|  0.4266638077859269|  261|
|       Cote d'Ivoire|  0.3322445163328404| 0.46323368706543266|  245|
|  West Bank and Gaza| 0.33422986011119876|  0.4917106376959971|  230|
```

```
|          Uzbekistan|  0.3371994476593011|  0.4404868661148826|  241|
|             Burundi| 0.37891604187011646| 0.32965066778410557|  225|
|                Oman| 0.39221495398251477|  0.4559117102460851|  261|
|           Venezuela| 0.39332660501566835|  0.6008977780844451|  242|
|              Jordan| 0.42964692809924426|  0.5713666525031367|  253|
|             Belarus|  0.4317817940171631|   0.553908079344052|  257|
|          Cabo Verde|  0.4483307510272256|  0.7352040231301348|  236|
+--------------------+--------------------+--------------------+-----+
only showing top 25 rows
```

Below, we remove certain selected countries based on criteria mentioned above. It should be noted that we have been very lenient here towards particular countries by keeping them on this list. In reality, perhaps we should have removed them, as well (e.g. Ghana and Guinea). We are thus left with the top 10 of best tracking countries based on the standard deviation of the case to fatality ratio being low.

In [20]:

```
countries_df_best = countries_df_best.filter((countries_df_best["Country_Region"] != 'Uganda') & (countries_df_best["Country_Region"] != 'Rwanda') & (countries_df_best["Country_Region"] != 'Saint Lucia'))

window = Window.orderBy(countries_df_best["Stdev_Case_Fat"])

countries_df_best = countries_df_best.withColumn("Rank", F.row_number().over(window))

countries_df_best.show(10)
```

```
+--------------------+--------------------+--------------------+-----+----+
|      Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|Rank|
+--------------------+--------------------+--------------------+-----+----+
|           Singapore|0.025860148802332693|0.031175630965621773|  283|   1|
|               Qatar| 0.07525506314186811| 0.09706676018345682|  256|   2|
|             Bahrain| 0.16734822061392865|  0.2141036358459203|  261|   3|
|            Maldives|  0.1824294888247625| 0.25205215911179063|  248|   4|
|           Sri Lanka| 0.21515182234282237| 0.23949510358940435|  283|   5|
|             Iceland| 0.23131809992516325| 0.29552475769697606|  257|   6|
|               Nepal| 0.24846209676491823| 0.25624908951815933|  283|   7|
|               Ghana|  0.2736646093311702|   0.407322033696884|  242|   8|
|United Arab Emirates| 0.27890160410244846|  0.3195480876213416|  283|   9|
|              Guinea|  0.2816915090558841|   0.414858078887033|  243|  10|
+--------------------+--------------------+--------------------+-----+----+
only showing top 10 rows
```

Poor tracking

In this scenario, we would like to create a top 10 of worst tracking countries around the globe. This top 10 will be much easier to establish than the previous (good tracking) list, because these countries will generally have a high reporting rate.

Again, several observations can be made:

- **Yemen is the 2nd worst tracking country of the virus according to this metric, however it should be noted that Yemen has 605 deaths out of 2071 total cases, with a population of over 30 million. Thus, several questions can be raised regarding the tracking of the virus in this third world country. Generally speaking, the total amount of cases is expected to be much higher on average, however this is not the case here. This could be due to an excellent containment of the virus, as well as the geographical location of the country (bottom part of the Arabian peninsula, which results in a low interaction rate with other countries safe for the UAE and Saudi-Arabia), however it must be noted that the country is currently experiencing genocide and thus could be considered inadequately equiped to sufficiently test for the presence of the virus. Although it could be argued that Yemen should be on this list, because it is clearly not tracking the virus adequately, we are not going to include it in this list due to the aformentioned reason. The comparison to other countries simply isn't adequate in our observation.**
- **MS Zaandam is a ship with extremely low deaths and cases (2 and 9 respectively) and thus is irrelevant in our list.**

- **We have chosen to exlude the Western Sahara. According to our metric, it's near the top of the list, but the amount of deaths and cases (1 and 10 respectively) on a total population of 602K is simply statistically not meaningful. Yes, it is to be expected that tracking the virus is going to be very difficult in this geographical region, but as previously stated, it can also be argued that the virus is simply less rampant in hotter regions. Thus, because it is not meaningful to compare these low figures to other countries, we have decided to exclude the country.**

This gives us a "truer" top 10 of poorly tracking countries, although this is of course merely our interpretation of the figures based on the cases stated and reasoning behind the exclusion of certain countries. It is not wrong to include the excluded countries, but perhaps it is fairer to include a different country in our top 10 that is statistically more viable versus these excluded ones.

In [21]:

```
countries_df_worst = countries_df_filtered.orderBy(countries_df_filtered["Stdev_Case_Fat"
].desc())

window = Window.orderBy(countries_df_worst["Stdev_Case_Fat"].desc())

countries_df_worst = countries_df_worst.withColumn("Rank", F.row_number().over(window))

countries_df_worst.show(25)
```

```
+------------------+------------------+------------------+-----+----+
|     Country_Region|    Stdev_Case_Fat|      Avg_Case_Fat|Count|Rank|
+------------------+------------------+------------------+-----+----+
|                US|18.217016354201867|1.9272010027526933| 3670|   1|
|             Yemen|11.775450297939296| 21.52944755078509|  215|   2|
|         MS Zaandam| 9.816959056846118|16.235867446247564|  228|   3|
|              Peru| 7.625936089962196| 3.987898312258953|  170|   4|
|           Belgium| 6.593688221334257|  6.79336859165134|  281|   5|
|           Hungary| 5.920368603310872| 6.059005824881824|  252|   6|
|     United Kingdom|  4.83059583471635|4.0362135494518885|  180|   7|
|            Mexico| 4.747268837379202|10.408207243501932|  177|   8|
|       Netherlands|  4.64978084432237| 4.803770956736284|  144|   9|
|             Italy| 4.638040583002451|  8.08262907548739|  185|  10|
|             Spain| 4.571133928549751|  6.07453437237955|  185|  11|
|     Western Sahara|4.3932166852436865| 7.668181818181818|  220|  12|
|            Canada| 4.287902401221546|1.7675164542669943|  176|  13|
|           Bahamas|3.8637210144927345|  3.41110412725333|  234|  14|
|           Ecuador| 3.706102374765361| 4.966120803362646|  255|  15|
|              Chad|3.5695689858751494| 5.411630859189694|  237|  16|
|Antigua and Barbuda|3.3366008420895397| 3.151075950169101|  243|  17|
|            Belize| 3.309778273300614|2.7398813632776937|  233|  18|
|            France| 3.232075505707301|1.5677666032705693|  222|  19|
|           Liberia|2.9385026151324656| 4.294330117361373|  240|  20|
|           Ireland|2.9257144330062297|3.7423908452522654|  256|  21|
|              Fiji|2.8814165663247078|2.4249389413147795|  237|  22|
|             Sudan|2.8218550089142815| 4.319131731494222|  243|  23|
|         San Marino|2.7573319931712295|  3.72994624946916|  258|  24|
|          Slovenia| 2.756891042687366|3.0294746685490606|  251|  25|
+------------------+------------------+------------------+-----+----+
only showing top 25 rows
```

This leaves us with the following top 10. Note that Belgium has done a very poor job of tracking the virus adequately according to our metric, as has the US. Other notable countries are the UK, the Netherlands, Italy and Spain, all of which have been severely hit by a second CoViD-19 wave recently, thus their apppearance on this list.

In [22]:

```
countries_df_worst = countries_df_worst.filter((countries_df_worst["Country_Region"] != '
MS Zaandam') & (countries_df_worst["Country_Region"] != 'Yemen') & (countries_df_worst["
Country_Region"] != 'Western Sahara'))

window = Window.orderBy(countries_df_worst["Stdev_Case_Fat"].desc())
```

```
countries_df_worst = countries_df_worst.withColumn("Rank", F.row_number().over(window))

countries_df_worst.show(10)
```

```
+--------------+------------------+------------------+-----+----+
|Country_Region|    Stdev_Case_Fat|      Avg_Case_Fat|Count|Rank|
+--------------+------------------+------------------+-----+----+
|            US|18.217016354201867|1.9272010027526933| 3670|   1|
|          Peru| 7.625936089962196| 3.987898312258953|  170|   2|
|       Belgium| 6.593688221334257|  6.79336859165134|  281|   3|
|       Hungary| 5.920368603310872| 6.059005824881824|  252|   4|
|United Kingdom|  4.830595834716351|4.0362135494518885|  180|   5|
|        Mexico| 4.747268837379202|10.408207243501932|  177|   6|
|   Netherlands|  4.64978084432237| 4.803770956736284|  144|   7|
|         Italy| 4.638040583002451|  8.08262907548739|  185|   8|
|         Spain| 4.571133928549751|  6.07453437237955|  185|   9|
|        Canada| 4.287902401221546|1.7675164542669943|  176|  10|
+--------------+------------------+------------------+-----+----+
only showing top 10 rows
```

1.3.2 Quality of Healthcare

We have to decided to use the average of the _Case_Fatality *Rate* to determine the quality of healthcare. We have selected this metric because the average can tell us something about the amount of people that have died proportionate to the amount of cases. A low average indicates to us that fewer people have died as the result of CoViD-19 due to better healthcare, whereas a high amount of deaths per cases tells us that the healthcare systems of a country may not be sufficiently up for this task (as was seen in Italy during the start of the pandemic). We note that this average does not take into account the age distribution of a country or the geographical spread (density) within a country and thus can only provide a limited scope on the situation. Whether or not a country appears in the list of good healthcare or poor healthcare is thus not just a function of its average. Our approach is a simplification due to the limited amount of data that we can work with. We would need more information on demographics as well as region density of these various countries in order for our metric to be more accurately describing of the quality of healthcare within a country.

Good Healthcare

Firstly, we will omit those regions whose averages are 0 (see 1.3.1 for the reasoning behind this).

In [23]:

```
countries_df.filter(countries_df["Avg_Case_Fat"] == 0).show(100)
```

```
+--------------------+--------------+------------+-----+
|      Country_Region|Stdev_Case_Fat|Avg_Case_Fat|Count|
+--------------------+--------------+------------+-----+
|             Eritrea|           0.0|         0.0|  235|
|            Cambodia|           0.0|         0.0|  283|
|            Dominica|           0.0|         0.0|  234|
|         Timor-Leste|           0.0|         0.0|  234|
|                Laos|           0.0|         0.0|  232|
|              Bhutan|           0.0|         0.0|  250|
|            Holy See|           0.0|         0.0|  246|
|Saint Kitts and N...|           0.0|         0.0|  231|
|Saint Vincent and...|           0.0|         0.0|  242|
|            Mongolia|           0.0|         0.0|  246|
|             Grenada|           0.0|         0.0|  234|
|          Seychelles|           0.0|         0.0|  242|
+--------------------+--------------+------------+-----+
```

As well as NaNs..

In [24]:

```
countries_df_filtered = countries_df.filter(countries_df["Avg_Case_Fat"] > 0).filter(~F.
isnan(countries_df["Avg_Case_Fat"]))
countries_df_best = countries_df_filtered.orderBy(countries_df_filtered["Avg_Case_Fat"])
```

Best healthcare

Below is an ordered table of countries based on the average of their _Case_Fatality *Rate*. A low average indeed should indicate that the quality of healthcare in this country is high(er). However, we must remain critical, especially when countries have low numbers of deaths and cases in comparison to their total population, when compared with other countries with similar population sizes. Their performance could indeed be significantly better due to better healthcare, however geographical region and climate, again, play a significant role in this metric and can not, as previously mentioned, be measured accurately.

We have decided to exclude Saint Lucia and Burundi from this top 10 list, due to the aforementioned reasons. While Saint Lucia's performance may indeed be worthy of a spot in this top 10, we would like to make more meaningful comparisons between countries, especially because the country is a very isolated island and low figures. This is not to say that these figures are inaccurate; they merely become statistically irrelevant due to their proportions being many times lower than other countries. Burundi was excluded because they have 1 death out of 620 cases on a total population of over 12 million. There are various possibilities for these low numbers: the virus has not spread in Africa at the same rate as first world countries and regions; the climate is much hotter, which means it is more difficult for the virus to survive in; underreporting.

Africa has generally been on the low end of figures of CoViD-19, which may be a good sign because the virus may not have spread there as widely as in Europe or North America. On the other hand, the various regimes and low accessibility to medical care, as well as the high probability for low testing rates, can all be contributing factors to why the figures in Africa may be lower than in reality. Thus, one can argue that these countries can not in a meaningful way be compared to other, first world countries and should therefore be exluded from the list.

Qatar has a very low amount of deaths (230), with their total cases being at over 135,000 (and a total population of nearly 3 million inhabitants). We should be critical of this and ask ourselves whether this is indeed possible. We have decided not to exclude Qatar from this list and assume that the country's healthcare is indeed sufficiently good to allow for reasonable treatment of CoViD-19 cases. We note that Singapore has 28 deaths out of 58,102 cases, with a total population of nearly 6 million. A similar question could be raised here namely whether this is indeed accurate. Singapore, however, is known as a very rich region and thus we assume that they can indeed provide their inhabitants with good healthcare.

In [25]:

```
countries_df_best.show(25)
```

```
+------------------+--------------------+--------------------+-----+
|      Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|
+------------------+--------------------+--------------------+-----+
|         Saint Lucia| 0.04343407746846115|0.002792048246593701|  242|
|           Singapore|0.025860148802332693|0.031175630965621773|  283|
|               Qatar| 0.07525506314186811| 0.09706676018345682|  256|
|             Bahrain| 0.16734822061392865|  0.2141036358459203|  261|
|           Sri Lanka| 0.21515182234282237| 0.23949510358940435|  283|
|            Maldives|  0.1824294888247625| 0.25205215911179063|  248|
|               Nepal| 0.24846209676491823| 0.25624908951815933|  283|
|             Iceland| 0.23131809992516325| 0.29552475769697606|  257|
|              Rwanda| 0.24384120413939261| 0.29868943036880685|  242|
|United Arab Emirates| 0.27890160410244846|  0.3195480876213416|  283|
|             Burundi| 0.37891604187011646| 0.32965066778410557|  225|
|              Uganda| 0.46513078895971216| 0.40204270435569534|  235|
|               Ghana|  0.2736646093311702|   0.407322033696884|  242|
|              Guinea|  0.2816915090558841|   0.414858078887033|  243|
|              Kuwait|  0.3271174474480425|  0.4266638077859269|  261|
|             Namibia|  0.4758002759710018|  0.4357711068353655|  242|
|          Uzbekistan|  0.3371994476593011|  0.4404868661148826|  241|
|                Oman| 0.39221495398251477|  0.4559117102460851|  261|
|       Cote d'Ivoire|  0.3322445163328404| 0.46323368706543266|  245|
|               Gabon| 0.29540273001242784|  0.4656401966131654|  242|
|          Mozambique|  0.3090379782498638| 0.47058102935681173|  234|
|            Botswana|  0.6189911342640264| 0.48427261666622073|  226|
```

```
|  West Bank and Gaza| 0.33422986011119876|  0.4917106376959971|  230|
|             Belarus|  0.4317817940171631|   0.553908079344052|  257|
|              Jordan| 0.42964692809924426|  0.5713666525031367|  253|
+--------------------+--------------------+--------------------+-----+
only showing top 25 rows
```

The final top 10 is thus:

In [26]:

```
countries_df_best = countries_df_best.filter((countries_df_best["Country_Region"] != 'Saint Lucia') & (countries_df_best["Country_Region"] != 'Burundi'))

window = Window.orderBy(countries_df_best["Avg_Case_Fat"])

countries_df_best = countries_df_best.withColumn("Rank", F.row_number().over(window))

countries_df_best.show(10)
```

```
+--------------------+--------------------+--------------------+-----+----+
|      Country_Region|      Stdev_Case_Fat|        Avg_Case_Fat|Count|Rank|
+--------------------+--------------------+--------------------+-----+----+
|           Singapore|0.025860148802332693|0.031175630965621773|  283|   1|
|               Qatar| 0.07525506314186811| 0.09706676018345682|  256|   2|
|             Bahrain| 0.16734822061392865|  0.2141036358459203|  261|   3|
|           Sri Lanka| 0.21515182234282237| 0.23949510358940435|  283|   4|
|            Maldives|  0.1824294888247625| 0.25205215911179063|  248|   5|
|               Nepal| 0.24846209676491823| 0.25624908951815933|  283|   6|
|             Iceland| 0.23131809992516325| 0.29552475769697606|  257|   7|
|              Rwanda| 0.24384120413939261| 0.29868943036880685|  242|   8|
|United Arab Emirates| 0.27890160410244846|  0.3195480876213416|  283|   9|
|              Uganda| 0.46513078895971216| 0.40204270435569534|  235|  10|
+--------------------+--------------------+--------------------+-----+----+
only showing top 10 rows
```

Not that this list, again, contains many geographically isolated countries, as well as very cold or very hot regions.

Poor Healthcare

As previously mentioned, we will exclude MS Zaandam from this list due to low figures. In fact, this is a ship and not a country/region. We have excluded many more countries here than usual to make a meaningful comparison. Several of these countries meet the critical thinking points that have been mentioned earlier: questionable regimes, low reporting rates, genocide in action, very hot regions or simply not enough data for meaningful statistical analysis.

In [27]:

```
countries_df_worst = countries_df_filtered.orderBy(countries_df_filtered["Avg_Case_Fat"].desc())

countries_df_worst.show(25)
```

```
+--------------+------------------+------------------+-----+
|Country_Region|    Stdev_Case_Fat|      Avg_Case_Fat|Count|
+--------------+------------------+------------------+-----+
|         Yemen|11.775450297939296| 21.52944755078509|  215|
|    MS Zaandam| 9.816959056846118|16.235867446247564|  228|
|        Mexico| 4.747268837379202|10.408207243501932|  177|
|         Italy| 4.638040583002451|  8.08262907548739|  185|
|Western Sahara|4.3932166852436865| 7.668181818181818|  220|
|       Belgium| 6.593688221334257|  6.79336859165134|  281|
|         Spain| 4.571133928549751|  6.07453437237955|  185|
|       Hungary| 5.920368603310872| 6.059005824881824|  252|
|        Sweden|2.6224330317831352| 5.803851660766782|  164|
```

```
|          Chad|3.569568956751494| 5.411630859189694|  237|
|       Ecuador| 3.706102374765361| 4.966120803362646|  255|
|   Netherlands|  4.64978084432237| 4.803770956736284|  144|
|         Sudan|2.8218550089142815| 4.319131731494222|  243|
|       Liberia|2.9385026151324656| 4.294330117361373|  240|
|         Niger| 2.750422743438422| 4.253447388815358|  236|
|United Kingdom|  4.83059583471635|4.0362135494518885|  180|
|          Peru| 7.625936089962196| 3.987898312258953|  170|
|       Ireland|2.9257144330062297|3.7423908452522654|  256|
|    San Marino|2.7573319931712295|  3.72994624946916|  258|
|      Barbados| 2.717925487181727|3.5452884342778272|  239|
|          Iran| 2.608878663734482| 3.425920452260099|  265|
|       Bahamas|3.8637210144927345|  3.41110412725333|  234|
|          Mali| 2.159956821315007| 3.394416066268014|  231|
|      Colombia| 2.489603356893213|3.2322042842000718|  167|
|         Syria| 2.068820152144134|3.1632871802896236|  234|
+--------------+------------------+------------------+-----+
only showing top 25 rows
```

Thus, we are left with the follow top 10, where we can say with certainty that the figures reported by these countries are indeed meaningful, accurate and significant. Notice that many countries from the *poor tracking* reappear in this list. Notably, Mexico is 1st here, Belgium remains 3rd and Peru has moved down. Thus, it can be said that Mexico's healthcare is even worse than its tracking. Peru does slightly better in terms of healthcare compared to tracking the virus. Belgium's performance is similarly poor and in fact shameful for a first world country. The same can be said about Spain, Italy and the Netherlands.

In [28]:

```
countries_df_worst = countries_df_worst.filter((countries_df_worst["Country_Region"] != '
MS Zaandam') & (countries_df_worst["Country_Region"] != 'Western Sahara') & (countries_d
f_worst["Country_Region"] != 'Yemen') & (countries_df_worst["Country_Region"] != 'Chad')
& (countries_df_worst["Country_Region"] != 'Sudan') & (countries_df_worst["Country_Regio
n"] != 'Liberia') & (countries_df_worst["Country_Region"] != 'Niger'))

window = Window.orderBy(countries_df_worst["Avg_Case_Fat"].desc())

countries_df_worst = countries_df_worst.withColumn("Rank", F.row_number().over(window))

countries_df_worst.show(10)
```

```
+--------------+------------------+------------------+-----+----+
|Country_Region|    Stdev_Case_Fat|      Avg_Case_Fat|Count|Rank|
+--------------+------------------+------------------+-----+----+
|        Mexico| 4.747268837379202|10.408207243501932|  177|   1|
|         Italy| 4.638040583002451|  8.08262907548739|  185|   2|
|       Belgium| 6.593688221334257|  6.79336859165134|  281|   3|
|         Spain| 4.571133928549751|  6.07453437237955|  185|   4|
|       Hungary| 5.920368603310872| 6.059005824881824|  252|   5|
|        Sweden|2.6224330317831352| 5.803851660766782|  164|   6|
|       Ecuador| 3.706102374765361| 4.966120803362646|  255|   7|
|   Netherlands|  4.64978084432237| 4.803770956736284|  144|   8|
|United Kingdom|  4.83059583471635|4.0362135494518885|  180|   9|
|          Peru| 7.625936089962196| 3.987898312258953|  170|  10|
+--------------+------------------+------------------+-----+----+
only showing top 10 rows
```

1.3.3 Incidence

The incidence tells us something about how good a country is containing outbreaks in its country.

We have used the following metrics to determine how well a country is containing outbreaks: the average and the standard deviation of the incidence.

If the average is low, we can state that a country is generally able to contain outbreaks, whereas a high average may indicate to us that the country is experiencing many spikes (outbreaks) and is thus not doing a good job of containing the virus.

If the standard deviation is low, this indicates to us that the country does not have many spikes (outliers) or in other words that the outbreaks are small when compared to the average of the current active cases per population. The country is good at stopping outbreaks. A high standard deviation, however, indicates to us that the country has experienced many spikes in active cases per total population and is not doing a good job of containing the spread of the CoViD-19 virus. It can thus be said that the situation's stability of the spread of the virus can be tracked using the standard deviation of the daily incidence.

First, we will determine the population from the *Confirmed* and _Incident*Rate* columns.

In [29]:

```
from pyspark.sql.types import IntegerType
df_with_incidence = df_with_incidence.withColumn("Population", (100000*df_with_incidence
["Confirmed"]/df_with_incidence["Incident_Rate"]).cast(IntegerType()))
df_with_incidence.show(25)
```

```
+--------------------+--------------------+--------------------+---------+------+--------
-+------+--------------------+------------------+----------+
|      Province_State|      Country_Region|         Last_Update|Confirmed|Deaths|Recovered|
Active|Case_Fatality_Ratio|      Incident_Rate|Population|
+--------------------+--------------------+--------------------+---------+------+--------
-+------+--------------------+------------------+----------+
|                null|         Afghanistan|2020-11-11 05:25:30|    42463|  1577|    34954
|  5932| 3.7138214445517272|109.07991172806463|  38928341|
|                null|             Albania|2020-11-11 05:25:30|    25294|   579|    12353
| 12362| 2.2890804143275085| 878.9352977969282|   2877800|
|                null|             Algeria|2020-11-11 05:25:30|    63446|  2077|    42626
| 18743| 3.2736500330990133| 144.6852700858221|  43851043|
|                null|             Andorra|2020-11-11 05:25:30|     5477|    75|     4405
|   997| 1.3693627898484573|7088.5912120623825|     77265|
|                null|              Angola|2020-11-11 05:25:30|    12816|   308|     6036
|  6472|  2.403245942571785| 38.99438780210762|  32866268|
|                null|Antigua and Barbuda|2020-11-11 05:25:30|      131|     3|      122
|     6| 2.2900763358778624|133.77175067396453|     97928|
|                null|           Argentina|2020-11-11 05:25:30|  1262476| 34183|  1081897
|146396|  2.707615827944452| 2793.349475991972|  45195776|
|                null|             Armenia|2020-11-11 05:25:30|   108687|  1609|    66835
| 40243| 1.4803978396680375| 3667.850733354167|   2963233|
|Australian Capita...|           Australia|2020-11-11 05:25:30|      114|     3|      111
|     0| 2.6315789473684212|26.629292221443592|    428099|
|     New South Wales|           Australia|2020-11-11 05:25:30|     4469|    53|     3156
|  1260| 1.1859476392929067|55.050505050505045|   8118000|
|  Northern Territory|           Australia|2020-11-11 05:25:30|       41|     0|       33
|     8|                 0.0|16.693811074918568|    245600|
|          Queensland|           Australia|2020-11-11 05:25:30|     1179|     6|     1163
|    10| 0.5089058524173028| 23.04760043006549|   5115499|
|     South Australia|           Australia|2020-11-11 05:25:30|      517|     4|      495
|    18| 0.7736943907156673|29.433532593225163|   1756500|
|            Tasmania|           Australia|2020-11-11 05:25:30|      230|    13|      217
|     0| 5.6521739130434785| 42.95051353874883|    535500|
|            Victoria|           Australia|2020-11-11 05:25:30|    20345|   819|    19522
|     4|   4.02555910543131| 306.8673735652121|   6629900|
|   Western Australia|           Australia|2020-11-11 05:25:30|      776|     9|      757
|    10| 1.1597938144329898|29.498973618185964|   2630600|
|                null|             Austria|2020-11-11 05:25:30|   164866|  1499|    98663
| 64704| 0.9092232479710796|  1830.54272517321|   9006400|
|                null|          Azerbaijan|2020-11-11 05:25:30|    67392|   867|    50009
| 16516|  1.286502849002849|  664.669462752147|  10139175|
|                null|             Bahamas|2020-11-11 05:25:30|     7012|   154|     5035
|  1823| 2.1962350256702794|1783.0987061599806|    393248|
|                null|             Bahrain|2020-11-11 05:25:30|    83811|   331|    81415
|  2065|0.39493622555511804| 4925.472339580261|   1701583|
|                null|          Bangladesh|2020-11-11 05:25:30|   423620|  6108|   341416
| 76096| 1.4418582692035316| 257.2236244275686| 164689383|
|                null|            Barbados|2020-11-11 05:25:30|      243|     7|      231
|     5|  2.880658436213992| 84.55968069151028|    287371|
|                null|             Belarus|2020-11-11 05:25:30|   108300|  1016|    91646
| 15638|  0.938134810710988|1146.1140964520093|   9449321|
|                null|             Belgium|2020-11-11 05:25:30|   507475| 13561|    30504
|463410| 2.6722498645253463| 4378.704177946879|  11589616|
```

```
|                null|             Belize|2020-11-11 05:25:30|     4414|    73|     2440
|  1901| 1.6538287267784322|1110.1023336292599|    397621|
+--------------------+-------------------+------------------+---------+------+--------
-+------+--------------------+------------------+----------+
only showing top 25 rows
```

This allows us to determine the Indicence on any given day for a country/region.

In [30]:

```
df_with_incidence = df_with_incidence.withColumn("Incidence", (df_with_incidence["Active
"]/df_with_incidence["Population"]/100000))
df_with_incidence.show()
```

```
+--------------------+--------------------+------------------+---------+------+--------
-+------+--------------------+------------------+----------+--------------------+
|      Province_State|      Country_Region|       Last_Update|Confirmed|Deaths|Recovered|
Active|Case_Fatality_Ratio|     Incident_Rate|Population|           Incidence|
+--------------------+--------------------+------------------+---------+------+--------
-+------+--------------------+------------------+----------+--------------------+
|                null|         Afghanistan|2020-11-11 05:25:30|    42463|  1577|    34954
|  5932| 3.7138214445517272|109.07991172806463|  38928341|1.523825533690223...|
|                null|             Albania|2020-11-11 05:25:30|    25294|   579|    12353
| 12362| 2.2890804143275085| 878.9352977969282|   2877800|4.295642504691084E-8|
|                null|             Algeria|2020-11-11 05:25:30|    63446|  2077|    42626
| 18743| 3.2736500330990133| 144.6852700858221|  43851043|4.274242690190972...|
|                null|             Andorra|2020-11-11 05:25:30|     5477|    75|     4405
|   997| 1.3693627898484573|7088.5912120623825|     77265|1.290364330550702...|
|                null|              Angola|2020-11-11 05:25:30|    12816|   308|     6036
|  6472|  2.403245942571785| 38.99438780210762|  32866268|1.969192242940391...|
|                null|Antigua and Barbuda|2020-11-11 05:25:30|      131|     3|      122
|     6| 2.2900763358778624|133.77175067396453|     97928|6.126950412547994...|
|                null|           Argentina|2020-11-11 05:25:30|  1262476| 34183|  1081897
|146396|  2.707615827944452| 2793.349475991972|  45195776|3.239152260600636...|
|                null|             Armenia|2020-11-11 05:25:30|   108687|  1609|    66835
| 40243| 1.4803978396680375| 3667.850733354167|   2963233|1.358077478213829...|
|Australian Capita...|           Australia|2020-11-11 05:25:30|      114|     3|      111
|     0| 2.6315789473684212|26.629292221443592|    428099|                0.0|
|     New South Wales|           Australia|2020-11-11 05:25:30|     4469|    53|     3156
|  1260| 1.1859476392929067|55.050505050505045|   8118000|1.552106430155210...|
|  Northern Territory|           Australia|2020-11-11 05:25:30|       41|     0|       33
|     8|                 0.0|16.693811074918568|    245600|3.257328990228012...|
|          Queensland|           Australia|2020-11-11 05:25:30|     1179|     6|     1163
|    10| 0.5089058524173028| 23.04760043006549|   5115499|1.954843505980550...|
|     South Australia|           Australia|2020-11-11 05:25:30|      517|     4|      495
|    18| 0.7736943907156673|29.433532593225163|   1756500|1.024765157984628...|
|            Tasmania|           Australia|2020-11-11 05:25:30|      230|    13|      217
|     0| 5.6521739130434785| 42.95051353874883|    535500|                0.0|
|            Victoria|           Australia|2020-11-11 05:25:30|    20345|   819|    19522
|     4|   4.02555910543131| 306.8673735652121|   6629900|6.033273503371091...|
|   Western Australia|           Australia|2020-11-11 05:25:30|      776|     9|      757
|    10| 1.1597938144329898|29.498973618185964|   2630600|3.801414126054892...|
|                null|             Austria|2020-11-11 05:25:30|   164866|  1499|    98663
| 64704| 0.9092232479710796|  1830.54272517321|   9006400|7.184224551430094E-8|
|                null|          Azerbaijan|2020-11-11 05:25:30|    67392|   867|    50009
| 16516|  1.286502849002849|  664.669462752147|  10139175|1.628929375417625E-8|
|                null|             Bahamas|2020-11-11 05:25:30|     7012|   154|     5035
|  1823| 2.1962350256702794|1783.0987061599806|    393248|4.635751485067947E-8|
|                null|             Bahrain|2020-11-11 05:25:30|    83811|   331|    81415
|  2065|0.39493622555511804| 4925.472339580261|   1701583|1.213575829095612...|
+--------------------+--------------------+------------------+---------+------+--------
-+------+--------------------+------------------+----------+--------------------+
only showing top 20 rows
```

We will now count the amount of updates a country has provided, as was explained earlier throughout section 1.3.1. Although the # of provinces here says 0, this should in fact be read as just 1. This indicates a country/region does not report for its internal regions separately.

In [31]:

```
df_count = df_with_incidence.groupBy("Country_Region").agg(F.count(df_with_incidence["La
st_Update"]).alias("#_Updates"), F.countDistinct(df_with_incidence["Province_State"]).al
ias("#_Province"))

df_count = df_count.withColumn("Count", F.when(df_count["#_Province"] == 0, df_count["#_
Updates"]).otherwise((df_count["#_Updates"]/df_count["#_Province"]).cast(IntegerType())))

df_average_incidence = df_with_incidence.groupBy("Country_Region").avg('Incidence')

df_average_incidence_with_count = df_average_incidence.join(df_count, "Country_Region")

df_average_incidence_with_count = df_average_incidence_with_count.filter(df_average_incid
ence_with_count["Count"]>100)

df_average_incidence_with_count.show()
```

```
+--------------+--------------------+---------+----------+-----+
|Country_Region|      avg(Incidence)|#_Updates|#_Province|Count|
+--------------+--------------------+---------+----------+-----+
|          Chad|3.549355065414161...|      166|         0|  166|
|      Paraguay|1.103201924426885...|      166|         0|  166|
|        Russia|1.586905100216670...|    13532|        83|  163|
|         Yemen|9.196320416728911...|      166|         0|  166|
|       Senegal|1.457960494848982E-9|      166|         0|  166|
|    Cabo Verde|1.266171434921538E-8|      166|         0|  166|
|        Sweden|7.115078199599006E-8|     3346|        21|  159|
|        Guyana|6.182183877730529E-9|      166|         0|  166|
|         Burma|8.745313857155528...|      166|         0|  166|
|       Eritrea|1.693882123285541...|      166|         0|  166|
|   Philippines|3.930907453607112E-9|      166|         0|  166|
|      Djibouti|3.213434436816673E-9|      166|         0|  166|
|      Malaysia|6.610481215555787...|      166|         0|  166|
|     Singapore|6.215113075161726E-9|      166|         0|  166|
|          Fiji|3.924475787663904...|      166|         0|  166|
|        Turkey|2.824312486081151...|      166|         0|  166|
|        Malawi|6.771755354092739...|      166|         0|  166|
|Western Sahara|1.734626719423523...|      166|         0|  166|
|          Iraq|1.011651233963927...|      166|         0|  166|
|       Germany|3.574203253015905...|     2822|        17|  166|
+--------------+--------------------+---------+----------+-----+
only showing top 20 rows
```

1.3.3.1 Average as a metric

We determine the average of the (daily) incidence per country.

In [32]:

```
df_average_incidence = df_with_incidence.groupBy("Country_Region").avg('Incidence')
df_average_incidence.show()
```

```
+--------------+--------------------+
|Country_Region|      avg(Incidence)|
+--------------+--------------------+
|          Chad|3.549355065414161...|
|      Paraguay|1.103201924426885...|
|        Russia|1.586905100216670...|
|         Yemen|9.196320416728911...|
|       Senegal|1.457960494848982E-9|
|    Cabo Verde|1.266171434921538E-8|
|        Sweden|7.115078199599006E-8|
|        Guyana|6.182183877730529E-9|
|         Burma|8.745313857155528...|
|       Eritrea|1.693882123285541...|
|   Philippines|3.930907453607112E-9|
|      Djibouti|3.213434436816673E-9|
|      Malaysia|6.610481215555787...|
```

```
|     Singapore|6.215113075161726E-9|
|          Fiji|3.924475787663904...|
|        Turkey|2.824312486081151...|
|        Malawi|6.771755354092739...|
|Western Sahara|1.734626719423523...|
|          Iraq|1.011651233963927...|
|       Germany|3.574203253015905...|
+--------------+--------------------+
only showing top 20 rows
```

Best control of outbreaks

We are solely interested in countries/regions whose averages are above 0. Next, we order by this average and establish a ranking. However, it should be noted that all of these countries have very low cases when compared to their total population, and as such have a very low average. It is thus meaningful to ask whether the reported figures are accurate and whether these figures are statistically relevant. Whilst this gives us a supposed list of countries that are the very best at containing outbreaks, several countries on this list may prove to be statistically irrelevant when compared to other countries. One country that does appear to be doing a very good job at containing the virus is China. China does indeed appear on this list, but only on rank 9. If we were to omit some of the countries that may be statistically irrelevant, China would rank much higher and this is probably more likely to reflect reality too.

In [33]:

```
df_average_incidence = df_average_incidence.filter(df_average_incidence["avg(incidence)"]
> 0)

df_average_incidence = df_average_incidence.orderBy(df_average_incidence["avg(incidence)"
])

window = Window.orderBy(df_average_incidence["avg(incidence)"])

df_average_incidence = df_average_incidence.withColumn("Rank", F.row_number().over(windo
w))

df_average_incidence.show(10)
```

```
+--------------+--------------------+----+
|Country_Region|      avg(Incidence)|Rank|
+--------------+--------------------+----+
|          Laos|1.357905650400272...|   1|
|   Timor-Leste|4.294957694624548...|   2|
|       Taiwan*|8.114155410638539...|   3|
|      Cambodia|8.784480327577467...|   4|
|         Niger|1.197520878296830...|   5|
|       Vietnam|1.261216846079179...|   6|
|      Thailand|1.518452513813174...|   7|
|Western Sahara|1.734626719423523...|   8|
|         China|1.787530889654422...|   9|
|        Brunei|2.051716787124127...|  10|
+--------------+--------------------+----+
only showing top 10 rows
```

Poor control of outbreaks

All of these countries have high reporting figures so we can assume that this list is accurate. Indeed, we see many familiar faces from previous lists in 1.3.1 and 1.3.2.

In [34]:

```
df_average_incidence_worst = df_average_incidence.orderBy(df_average_incidence["avg(incid
ence)"].desc())

window = Window.orderBy(df_average_incidence_worst["avg(incidence)"].desc())
```

```
df_average_incidence_worst = df_average_incidence_worst.withColumn("Rank", F.row_number(
).over(window))

df_average_incidence_worst.show(10)
```

```
+--------------+--------------------+----+
|Country_Region|      avg(Incidence)|Rank|
+--------------+--------------------+----+
|          Peru|1.487328567063268...|   1|
|            US|1.339846693743519...|   2|
|       Belgium|8.479823844210152E-8|   3|
|        Sweden|7.115078199599006E-8|   4|
|         Spain| 7.01229674135203E-8|   5|
|       Andorra|4.665618071302164E-8|   6|
|   Netherlands|4.620605209113434...|   7|
|        Panama|4.582580622308114E-8|   8|
|        Brazil|4.138256374735803E-8|   9|
|    Costa Rica|4.077470090601778E-8|  10|
+--------------+--------------------+----+
only showing top 10 rows
```

1.3.3.2 Standard deviation as a metric

We will now take a closer look at the standard deviation as a metric. As previously explained, if the standard deviation is low, this indicates to us that the country does not have many spikes (outliers) or in other words that the outbreaks are small when compared to the average of the current active cases per population. The country is good at stopping outbreaks. A high standard deviation, however, indicates to us that the country has experienced many spikes in active cases per total population and is not doing a good job of containing the spread of the CoViD-19 virus. It can thus be said that the situation's stability of the spread of the virus can be tracked using the standard deviation of the daily incidence.

In [35]:

```
df_stddev_incidence = df_with_incidence.groupBy("Country_Region").agg(F.stddev(df_with_i
ncidence.Incidence).alias('Stdev_Incidence'))

df_stddev_incidence.show()
```

```
+--------------+--------------------+
|Country_Region|     Stdev_Incidence|
+--------------+--------------------+
|          Chad|2.601529431204806...|
|      Paraguay|1.007300758254993...|
|        Russia|1.584646423741985...|
|         Yemen|4.088159624976144...|
|       Senegal|6.734693496780161...|
|    Cabo Verde|3.957833588376793E-9|
|        Sweden|3.362049973787039...|
|        Guyana|4.708790507188314E-9|
|         Burma|1.294039834779389...|
|       Eritrea|1.094388208206686...|
|   Philippines|1.542677368305759...|
|      Djibouti|6.265982039490021...|
|      Malaysia|1.053844833504068...|
|     Singapore|6.758799993807593E-9|
|          Fiji|3.552554158166786...|
|        Turkey|1.168173494759766...|
|        Malawi|3.572257715865910...|
|Western Sahara|6.355917168866697...|
|          Iraq|4.702848230714968E-9|
|       Germany|6.683170902395847E-9|
+--------------+--------------------+
only showing top 20 rows
```

Best control of outbreaks

Best control of outbreaks

A similar assessment to the one for the average as a metric can be made here.

In [36]:

```
df_stddev_incidence = df_stddev_incidence.filter(df_stddev_incidence["Stdev_Incidence"] > 0)

df_stddev_incidence = df_stddev_incidence.orderBy(df_stddev_incidence["Stdev_Incidence"])

window = Window.orderBy(df_stddev_incidence["Stdev_Incidence"])

df_stddev_incidence = df_stddev_incidence.withColumn("Rank", F.row_number().over(window))

df_stddev_incidence.show(25)
```

```
+---------------+--------------------+----+
|  Country_Region|     Stdev_Incidence|Rank|
+---------------+--------------------+----+
|            Laos|1.132280864467111...|   1|
|        Thailand|3.997763388508769...|   2|
|     Timor-Leste|5.240593899975499...|   3|
|         Taiwan*|5.283622296990616...|   4|
|  Western Sahara|6.355917168866697...|   5|
|           Niger|7.43027077444582E-12|   6|
|        Cambodia|1.178365320829944...|   7|
|         Vietnam|1.437298471295441...|   8|
|          Brunei|2.144712909107385...|   9|
|         Burundi|2.290990089679638...|  10|
|            Chad|2.601529431204806...|  11|
|            Fiji|3.552554158166786...|  12|
|            Mali|3.772245726035339...|  13|
|           Yemen|4.088159624976144...|  14|
| Solomon Islands|5.172224999911312...|  15|
|    Sierra Leone|6.61345445580143E-11|  16|
|     New Zealand|7.962386456914006...|  17|
|Marshall Islands|8.791751682212788...|  18|
|       Mauritius|1.005809241796792...|  19|
|    Burkina Faso|1.042767393999658...|  20|
|        Mongolia|1.068153354262428...|  21|
|         Eritrea|1.094388208206686...|  22|
|Papua New Guinea|1.114835415213979...|  23|
|           China|1.244183315786067...|  24|
|         Grenada|1.357051199197534...|  25|
+---------------+--------------------+----+
only showing top 25 rows
```

In [37]:

```
df_stddev_incidence.show(10)
```

```
+--------------+--------------------+----+
|Country_Region|     Stdev_Incidence|Rank|
+--------------+--------------------+----+
|          Laos|1.132280864467111...|   1|
|      Thailand|3.997763388508769...|   2|
|   Timor-Leste|5.240593899975499...|   3|
|       Taiwan*|5.283622296990616...|   4|
|Western Sahara|6.355917168866697...|   5|
|         Niger|7.43027077444582E-12|   6|
|      Cambodia|1.178365320829944...|   7|
|       Vietnam|1.437298471295441...|   8|
|        Brunei|2.144712909107385...|   9|
|       Burundi|2.290990089679638...|  10|
+--------------+--------------------+----+
only showing top 10 rows
```

Poor control of outbreaks

In [38]:

```
df_stddev_incidence_worst = df_stddev_incidence.orderBy(df_stddev_incidence["Stdev_Incide
nce"].desc())

window = Window.orderBy(df_stddev_incidence_worst["Stdev_Incidence"].desc())

df_stddev_incidence_worst = df_stddev_incidence_worst.withColumn("Rank", F.row_number().
over(window))

df_stddev_incidence_worst.show(25)
```

```
+--------------+--------------------+----+
|Country_Region|     Stdev_Incidence|Rank|
+--------------+--------------------+----+
|       Vanuatu|                 NaN|   1|
|            US|1.431052953129307E-7|   2|
|          Peru|1.196186149297552...|   3|
|       Belgium|9.624978980435268E-8|   4|
|         Spain|9.068253371816093E-8|   5|
|        Brazil|6.319886027859303E-8|   6|
|      Holy See|5.956235722471743E-8|   7|
|       Andorra|5.784162187653073E-8|   8|
|   Netherlands|5.400814626228451E-8|   9|
|       Czechia|5.132755992611442E-8|  10|
|    Luxembourg|3.852116935520802...|  11|
|United Kingdom|3.536383420356969E-8|  12|
|        France|3.420594959323419E-8|  13|
|        Sweden|3.362049973787039...|  14|
|   Switzerland|3.356861412921749...|  15|
|      Slovenia|3.067982465959057E-8|  16|
|    Costa Rica|3.044889309065979...|  17|
|       Armenia|2.972177155472856E-8|  18|
|         Chile|2.916688505249891E-8|  19|
|    Montenegro|2.781172377120435...|  20|
|         Qatar|2.678119681036311...|  21|
|      Slovakia|2.677220873184263E-8|  22|
|       Ireland|2.401748052982767E-8|  23|
|    San Marino|2.371511277212505...|  24|
|        Jordan|2.352156311809639E-8|  25|
+--------------+--------------------+----+
only showing top 25 rows
```

We will remove two regions: the Holy See, due to its low amount of inhabitants, as well as Vanuata due to NaN. We cannot say many meaningful things about either of these two. The other countries on this list, however, are indeed familiar faces. It can thus be established that these reappearing countries, such as the US, Belgium, Peru, Spain, Brazil, the Netherlands, the UK and several others have not only been hit hard by CoViD-19 but have also not adequately managed to contain the virus.

In [39]:

```
df_stddev_incidence_worst = df_stddev_incidence_worst.filter(~F.isnan(df_stddev_incidence
_worst["Stdev_Incidence"])).filter((df_stddev_incidence_worst["Country_Region"] != 'Holy
See'))

window = Window.orderBy(df_stddev_incidence_worst["Stdev_Incidence"].desc())

df_stddev_incidence_worst = df_stddev_incidence_worst.withColumn("Rank", F.row_number().
over(window))

df_stddev_incidence_worst.show(10)
```

```
+--------------+--------------------+----+
|Country_Region|     Stdev_Incidence|Rank|
+--------------+--------------------+----+
|            US|1.431052953129307E-7|   1|
|          Peru|1.196186149297552...|   2|
```

```
|       Belgium|9.624978980435268E-8|   3|
|         Spain|9.068253371816093E-8|   4|
|        Brazil|6.319886027859303E-8|   5|
|       Andorra|5.784162187653073E-8|   6|
|   Netherlands|5.400814626228451E-8|   7|
|       Czechia|5.132755992611442E-8|   8|
|    Luxembourg|3.852116935520802...|   9|
|United Kingdom|3.536383420356969E-8|  10|
+--------------+--------------------+----+
only showing top 10 rows
```

1.3.4 The situation during the past seven days

We have decided to look at the rolling average of the daily incidence rate to determine whether a country is doing better than seven days prior. This moving average allows us to contain random daily spikes that may skew the top 10. It could be that a country is doing well, but has a random spike on the day 7 days prior to this one. To smooth this out and reduce randomness, we use a moving average to make meaningful comparisons.

In [40]:

```
from datetime import timedelta, datetime

df_with_incidence_day_only = df_with_incidence.withColumn("Last_Update", F.to_date(df_wi
th_incidence["Last_Update"]))

df_with_incidence_day_only.show()
```

```
+------------------+------------------+-----------+---------+------+---------+------
+------------------+------------------+-----------+--------------------+
|      Province_State|     Country_Region|Last_Update|Confirmed|Deaths|Recovered|Active|C
ase_Fatality_Ratio|     Incident_Rate|Population|           Incidence|
+------------------+------------------+-----------+---------+------+---------+------
+------------------+------------------+-----------+--------------------+
|              null|       Afghanistan| 2020-11-11|    42463|  1577|    34954|  5932|
3.7138214445517272|109.07991172806463|  38928341|1.523825533690223...|
|              null|           Albania| 2020-11-11|    25294|   579|    12353| 12362|
2.2890804143275085| 878.9352977969282|   2877800|4.295642504691084E-8|
|              null|           Algeria| 2020-11-11|    63446|  2077|    42626| 18743|
3.2736500330990133| 144.6852700858221|  43851043|4.274242690190972...|
|              null|           Andorra| 2020-11-11|     5477|    75|     4405|   997|
1.3693627898484573|7088.5912120623825|     77265|1.290364330550702...|
|              null|            Angola| 2020-11-11|    12816|   308|     6036|  6472|
2.403245942571785| 38.99438780210762|  32866268|1.969192242940391...|
|              null|Antigua and Barbuda| 2020-11-11|      131|     3|      122|     6|
2.2900763358778624|133.77175067396453|     97928|6.126950412547994...|
|              null|         Argentina| 2020-11-11|  1262476| 34183|  1081897|146396|
2.707615827944452| 2793.349475991972|  45195776|3.239152260600636...|
|              null|           Armenia| 2020-11-11|   108687|  1609|    66835| 40243|
1.4803978396680375| 3667.850733354167|   2963233|1.358077478213829...|
|Australian Capita...|         Australia| 2020-11-11|      114|     3|      111|     0|
2.6315789473684212|26.629292221443592|    428099|                 0.0|
|     New South Wales|         Australia| 2020-11-11|     4469|    53|     3156|  1260|
1.1859476392929067|55.050505050505045|   8118000|1.552106430155210...|
|  Northern Territory|         Australia| 2020-11-11|       41|     0|       33|     8|
0.0|16.693811074918568|    245600|3.257328990228012...|
|          Queensland|         Australia| 2020-11-11|     1179|     6|     1163|    10|
0.5089058524173028| 23.04760043006549|   5115499|1.954843505980550...|
|     South Australia|         Australia| 2020-11-11|      517|     4|      495|    18|
0.7736943907156673|29.433532593225163|   1756500|1.024765157984628...|
|            Tasmania|         Australia| 2020-11-11|      230|    13|      217|     0|
5.6521739130434785| 42.95051353874883|    535500|                 0.0|
|            Victoria|         Australia| 2020-11-11|    20345|   819|    19522|     4|
4.02555910543131| 306.8673735652121|   6629900|6.033273503371091...|
|   Western Australia|         Australia| 2020-11-11|      776|     9|      757|    10|
1.1597938144329898|29.498973618185964|   2630600|3.801414126054892...|
|              null|           Austria| 2020-11-11|   164866|  1499|    98663| 64704|
0.9092232479710796|  1830.54272517321|   9006400|7.184224551430094E-8|
|              null|        Azerbaijan| 2020-11-11|    67392|   867|    50009| 16516|
1.286502849002849|  664.669462752147|  10139175|1.628929375417625E-8|
```

```
|                null|            Bahamas| 2020-11-11|     7012|   154|     5035|  1823|
2.1962350256702794|1783.0987061599806|    393248|4.635751485067947E-8|
|                null|            Bahrain| 2020-11-11|    83811|   331|    81415|  2065|
0.39493622555511804| 4925.472339580261|   1701583|1.213575829095612...|
+-------------------+------------------+-----------+---------+------+---------+------
+------------------+-----------------+----------+--------------------+
only showing top 20 rows
```

We will first group by _Country *Region* and _Last *Update*.

In [41]:

```
df_with_avg_incidence_day_only = df_with_incidence_day_only.groupBy('Country_Region', 'La
st_Update').avg('Incidence')
df_with_avg_incidence_day_only.show()
```

```
+--------------------+-----------+--------------------+
|      Country_Region|Last_Update|      avg(Incidence)|
+--------------------+-----------+--------------------+
|      Western Sahara| 2020-11-11|1.674116485025028...|
|             Georgia| 2020-11-09|3.686477451221731...|
|          Madagascar| 2020-11-09|1.653965858027832...|
|             Lesotho| 2020-11-10|4.196518430138004...|
|              Greece| 2020-11-07|2.730964891678602E-8|
|              France| 2020-11-06|7.003145590225364E-8|
|             Morocco| 2020-11-05|1.014398102570001...|
|            Djibouti| 2020-11-04|4.757075390535647...|
|             Burundi| 2020-11-03|6.475605469111362...|
|         Switzerland| 2020-11-03|1.078880786983410...|
|            Eswatini| 2020-11-01|1.327398540206384...|
|              Monaco| 2020-10-30|2.064009784935276...|
|       Liechtenstein| 2020-10-31|4.300285811678947E-8|
|            Dominica| 2020-10-29|1.250156269533691...|
|           Nicaragua| 2020-10-29|1.710303818189118...|
|      United Kingdom| 2020-10-29|4.432900478908612E-8|
|           Venezuela| 2020-10-29|1.649321072278137...|
|Central African R...| 2020-10-27|5.956812796650106E-9|
|            Paraguay| 2020-10-27|2.597815922260404E-8|
|             Liberia| 2020-10-28|1.166543454633421...|
+--------------------+-----------+--------------------+
only showing top 20 rows
```

Next, we will add a dummy time of 00:00:00 to our date, which will allow us to make calculations on it during a later step.

In [42]:

```
df_with_incidence_with_time = df_with_avg_incidence_day_only.withColumn("Last_Update", F.
to_timestamp(df_with_avg_incidence_day_only["Last_Update"]))
df_with_incidence_with_time.show()
```

```
+--------------------+-------------------+--------------------+
|      Country_Region|        Last_Update|      avg(Incidence)|
+--------------------+-------------------+--------------------+
|      Western Sahara|2020-11-11 00:00:00|1.674116485025028...|
|             Georgia|2020-11-09 00:00:00|3.686477451221731...|
|          Madagascar|2020-11-09 00:00:00|1.653965858027832...|
|             Lesotho|2020-11-10 00:00:00|4.196518430138004...|
|              Greece|2020-11-07 00:00:00|2.730964891678602E-8|
|              France|2020-11-06 00:00:00|7.003145590225364E-8|
|             Morocco|2020-11-05 00:00:00|1.014398102570001...|
|            Djibouti|2020-11-04 00:00:00|4.757075390535647...|
|             Burundi|2020-11-03 00:00:00|6.475605469111362...|
|         Switzerland|2020-11-03 00:00:00|1.078880786983410...|
|            Eswatini|2020-11-01 00:00:00|1.327398540206384...|
|              Monaco|2020-10-30 00:00:00|2.064009784935276...|
|       Liechtenstein|2020-10-31 00:00:00|4.300285811678947E-8|
|            Dominica|2020-10-29 00:00:00|1.250156269533691...|
```

```
|           Nicaragua|2020-10-29 00:00:00|1.710303818189118...|
|      United Kingdom|2020-10-29 00:00:00|4.432900478908612E-8|
|           Venezuela|2020-10-29 00:00:00|1.649321072278137...|
|Central African R...|2020-10-27 00:00:00|5.956812796650106E-9|
|            Paraguay|2020-10-27 00:00:00|2.597815922260404E-8|
|             Liberia|2020-10-28 00:00:00|1.166543454633421...|
+--------------------+-------------------+--------------------+
only showing top 20 rows
```

We calculate the 7-day rolling (also known as moving) average over our average incidence rate. This average incidence rate was determined as the average of a country's regions on a particular day. We partition by _Country*Region* so that a moving average is calculated over each country individually.

In [43]:

```
days = lambda i: i * 86400

window = (Window().partitionBy('Country_Region').orderBy(F.col('Last_Update').cast('long
')).rangeBetween(-days(7),0))

df_rolling_average = df_with_incidence_with_time.withColumn('Rolling_Average', F.avg('avg
(Incidence)').over(window))

df_rolling_average.show()
```

```
+--------------+-------------------+--------------------+--------------------+
|Country_Region|        Last_Update|      avg(Incidence)|     Rolling_Average|
+--------------+-------------------+--------------------+--------------------+
|          Chad|2020-05-30 00:00:00|1.521990417670089...|1.521990417670089...|
|          Chad|2020-05-31 00:00:00|1.363703414232400...|1.442846915951245...|
|          Chad|2020-06-01 00:00:00|1.351527490891039...|1.412407107597843...|
|          Chad|2020-06-02 00:00:00|1.126272909075866...|1.340873557967349...|
|          Chad|2020-06-03 00:00:00|1.065393292369062...|1.285777504847691...|
|          Chad|2020-06-04 00:00:00|9.984257139915788...|1.237885539705006...|
|          Chad|2020-06-05 00:00:00|7.853470555177663...|1.173237184821115...|
|          Chad|2020-06-06 00:00:00|6.757637454455199...|1.111053004899165...|
|          Chad|2020-06-07 00:00:00|5.78356358714634E-11|9.930987475297334...|
|          Chad|2020-06-08 00:00:00|5.844443559660628...|8.956913652464412...|
|          Chad|2020-06-09 00:00:00|5.113888114703049...|7.906740303188495...|
|          Chad|2020-06-10 00:00:00|4.07893431935584E-11|7.008765956763142...|
|          Chad|2020-06-11 00:00:00|4.139813936062643...|6.194501083309643...|
|          Chad|2020-06-12 00:00:00|4.261573169476250...|5.479165587004701...|
|          Chad|2020-06-13 00:00:00|3.957175085942233...|4.992128653350273...|
|          Chad|2020-06-14 00:00:00|3.531017768994608...|4.588801192667699...|
|          Chad|2020-06-15 00:00:00|3.470138152287804...|4.299623013310382...|
|          Chad|2020-06-16 00:00:00|3.470138152287804...|4.002834837388779...|
|          Chad|2020-06-17 00:00:00|3.591897385701411...|3.812585996263575...|
|          Chad|2020-06-18 00:00:00|3.591897385701411...|3.751706379556771...|
+--------------+-------------------+--------------------+--------------------+
only showing top 20 rows
```

In [44]:

```
last_date = df_rolling_average.select(F.max("Last_Update")).first()
today = last_date[0]

seven_days_ago = (last_date[0] - timedelta(days=7))
```

We determine the rolling average figures at the most recent day in our dataset and add these as a column.

In [45]:

```
df_today = df_rolling_average.filter(df_rolling_average["Last_Update"] == today)

df_today.show()
```

```
+--------------+-------------------+--------------------+--------------------+
|Country_Region|        Last_Update|      avg(Incidence)|     Rolling_Average|
```

```
+--------------+-------------------+--------------------+--------------------+
|          Chad|2020-11-11 00:00:00|2.983101400243445...|4.231133446368392...|
|      Paraguay|2020-11-11 00:00:00|2.515516934383732...|2.575278427966839...|
|        Russia|2020-11-11 00:00:00|3.343745085812556...|3.182147173010132...|
|         Yemen|2020-11-11 00:00:00|2.414003796959 75E-11|2.854056609121471...|
|       Senegal|2020-11-11 00:00:00|1.672247793678067...|3.949192407743567...|
|    Cabo Verde|2020-11-11 00:00:00|1.187078115135785...|1.299938702367800...|
|        Sweden|2020-11-11 00:00:00|1.369752580132391...|1.221074994228177...|
|        Guyana|2020-11-11 00:00:00|1.042516890044980...|1.058567001619932E-8|
|         Burma|2020-11-11 00:00:00|2.640884837755497E-9|2.665604658068318...|
|       Eritrea|2020-11-11 00:00:00|1.494461890798823...|1.515610153086249...|
|   Philippines|2020-11-11 00:00:00|2.753121149997019...|2.878770093799857E-9|
|      Djibouti|2020-11-11 00:00:00|7.591077750854756...|6.123470141210984...|
|      Malaysia|2020-11-11 00:00:00|3.536427333400935E-9|3.422418810982488...|
|     Singapore|2020-11-11 00:00:00|1.025580893291213...|1.038400677831050...|
|          Fiji|2020-11-11 00:00:00|1.115518649240778...|1.115518649240777...|
|        Turkey|2020-11-11 00:00:00|5.430460832581892...|5.266820772394838E-9|
|        Malawi|2020-11-11 00:00:00|2.127553358071150...|2.165452078487820...|
|Western Sahara|2020-11-11 00:00:00|1.674116485025028...|1.674116485025028...|
|          Iraq|2020-11-11 00:00:00|1.472136132353573...|1.464463188678238...|
|       Germany|2020-11-11 00:00:00|2.591412499990955...|2.383970790520691...|
+--------------+-------------------+--------------------+--------------------+
only showing top 20 rows
```

We determine the rolling average figures at 7 days ago and add these as a column.

In [46]:

```
df_seven_days_ago = df_rolling_average.filter(df_rolling_average["Last_Update"] == seven_
days_ago)
df_seven_days_ago = df_seven_days_ago.withColumnRenamed('Rolling_Average', 'Rolling_Aver
age_7_days_ago')
df_seven_days_ago.show()
```

```
+--------------+-------------------+--------------------+--------------------------+
|Country_Region|        Last_Update|      avg(Incidence)|Rolling_Average_7_days_ago|
+--------------+-------------------+--------------------+--------------------------+
|          Chad|2020-11-04 00:00:00|5.113887803371501...|      4.428992214564233...|
|      Paraguay|2020-11-04 00:00:00|2.616743287444988E-8|      2.629326526607789E-8|
|        Russia|2020-11-04 00:00:00|3.053571826788404E-8|      2.896740519915526...|
|         Yemen|2020-11-04 00:00:00|2.916921352457742...|      3.168380048989396...|
|       Senegal|2020-11-04 00:00:00|9.854317944133662...|      2.889853269901766...|
|    Cabo Verde|2020-11-04 00:00:00|1.248228379029763...|      1.294767153247911...|
|        Sweden|2020-11-04 00:00:00|1.127623383534503...|      1.035629186877423...|
|        Guyana|2020-11-04 00:00:00|1.013274274402810...|      1.068101877844223...|
|         Burma|2020-11-04 00:00:00|2.836621656755399...|      3.308503618909683E-9|
|       Eritrea|2020-11-04 00:00:00|1.522659714315200...|      1.512085369490173...|
|   Philippines|2020-11-04 00:00:00|2.817639558870949...|      3.278725064640489...|
|      Djibouti|2020-11-04 00:00:00|4.757075390535647...|      5.921041503631781...|
|      Malaysia|2020-11-04 00:00:00|3.131372718102891E-9|      3.116503670571159E-9|
|     Singapore|2020-11-04 00:00:00|1.093953139833534...|      1.162325035770405...|
|          Fiji|2020-11-04 00:00:00|1.115518649240778...|      9.760788180856806...|
|        Turkey|2020-11-04 00:00:00|5.076295188325950...|      4.839260908589372E-9|
|        Malawi|2020-11-04 00:00:00|2.179827510301384...|      2.202697302344853...|
|Western Sahara|2020-11-04 00:00:00|1.674116485025028...|      1.674116485025028...|
|          Iraq|2020-11-04 00:00:00|1.553160428628720...|       1.54248854659493E-8|
|       Germany|2020-11-04 00:00:00|2.021246378871420...|       1.68903789289067E-8|
+--------------+-------------------+--------------------+--------------------------+
only showing top 20 rows
```

We join both of these together.

In [47]:

```
df_incidence_joined = df_today.join(df_seven_days_ago, "Country_Region").select('Country_
Region', 'Rolling_Average', 'Rolling_Average_7_days_ago')
df_incidence_joined.show()
```

```
+--------------+--------------------+--------------------------+
```

```
|Country_Region|     Rolling_Average|Rolling_Average_7_days_ago|
+--------------+--------------------+--------------------------+
|          Chad|4.231133446368392...|      4.428992214564233...|
|      Paraguay|2.575278427966839...|      2.629326526607789E-8|
|        Russia|3.182147173010132...|      2.896740519915526...|
|         Yemen|2.854056609121471...|      3.168380048989396...|
|       Senegal|3.949192407743567...|      2.889853269901766...|
|    Cabo Verde|1.299938702367800...|      1.294767153247911...|
|        Sweden|1.221074994228177...|      1.035629186877423...|
|        Guyana|1.058567001619932E-8|      1.068101877844223...|
|         Burma|2.665604658068318...|      3.308503618909683E-9|
|       Eritrea|1.515610153086249...|      1.512085369490173...|
|   Philippines|2.878770093799857E-9|      3.278725064640489...|
|      Djibouti|6.123470141210984...|      5.921041503631781...|
|      Malaysia|3.422418810982488...|      3.116503670571159E-9|
|     Singapore|1.038400677831050...|      1.162325035770405...|
|          Fiji|1.115518649240777...|      9.760788180856806...|
|        Turkey|5.266820772394838E-9|      4.839260908589372E-9|
|        Malawi|2.165452078487820...|      2.202697302344853...|
|Western Sahara|1.674116485025028...|      1.674116485025028...|
|          Iraq|1.464463188678238...|       1.54248854659493E-8|
|       Germany|2.383970790520691...|       1.68903789289067E-8|
+--------------+--------------------+--------------------------+
only showing top 20 rows
```

Lastly, we take the difference of both and add these values as a separate column.

In [48]:

```
df_incidence_joined = df_incidence_joined.withColumn("Difference", df_incidence_joined['R
olling_Average'] - df_incidence_joined['Rolling_Average_7_days_ago'])
df_incidence_joined.show()
```

```
+--------------+--------------------+--------------------------+--------------------+
|Country_Region|     Rolling_Average|Rolling_Average_7_days_ago|          Difference|
+--------------+--------------------+--------------------------+--------------------+
|          Chad|4.231133446368392...|      4.428992214564233...|-1.97858768195841...|
|      Paraguay|2.575278427966839...|      2.629326526607789E-8|-5.40480986409493...|
|        Russia|3.182147173010132...|      2.896740519915526...|2.854066530946061E-9|
|         Yemen|2.854056609121471...|      3.168380048989396...|-3.14323439867925...|
|       Senegal|3.949192407743567...|      2.889853269901766...|-2.49493402912741...|
|    Cabo Verde|1.299938702367800...|      1.294767153247911...|5.171549119888898...|
|        Sweden|1.221074994228177...|      1.035629186877423...|1.854458073507546...|
|        Guyana|1.058567001619932E-8|      1.068101877844223...|-9.53487622429179...|
|         Burma|2.665604658068318...|      3.308503618909683E-9|-6.42898960841364...|
|       Eritrea|1.515610153086249...|      1.512085369490173...|3.524783596075687...|
|   Philippines|2.878770093799857E-9|      3.278725064640489...|-3.99954970840632...|
|      Djibouti|6.123470141210984...|      5.921041503631781...|2.024286375792024...|
|      Malaysia|3.422418810982488...|      3.116503670571159E-9|3.059151404113296...|
|     Singapore|1.038400677831050...|      1.162325035770405...|-1.23924357939354...|
|          Fiji|1.115518649240777...|      9.760788180856806...|1.394398311550971...|
|        Turkey|5.266820772394838E-9|      4.839260908589372E-9|4.275598638054658...|
|        Malawi|2.165452078487820...|      2.202697302344853...|-3.72452238570328...|
|Western Sahara|1.674116485025028...|      1.674116485025028...|                 0.0|
|          Iraq|1.464463188678238...|       1.54248854659493E-8|-7.80253579166919...|
|       Germany|2.383970790520691...|       1.68903789289067E-8|6.949328976300217E-9|
+--------------+--------------------+--------------------------+--------------------+
only showing top 20 rows
```

We note during ranking that two entries have null values, so we drop these. These are the two ships in the dataset.

In [49]:

```
df_incidence_joined_best = df_incidence_joined.orderBy(df_incidence_joined["Difference"])

window = Window.orderBy(df_incidence_joined_best["Difference"])
```

```
df_incidence_joined_best = df_incidence_joined_best.withColumn("Rank", F.row_number().over(window))

df_incidence_joined_best.show(10)
```

```
+---------------+--------------------+--------------------------+--------------------+----+
|  Country_Region|     Rolling_Average|Rolling_Average_7_days_ago|          Difference|Rank|
+---------------+--------------------+--------------------------+--------------------+----+
|      MS Zaandam|                null|                      null|                null|   1|
|Diamond Princess|                null|                      null|                null|   2|
|         Tunisia| 2.71178073540617E-8|      4.385210366036887E-8|-1.67342963063071...|   3|
|         Andorra|1.478032375460512...|      1.619596201882013...|-1.41563826421500...|   4|
|         Czechia|1.583998619380861...|      1.664982040587470...|-8.09834212066094...|   5|
|         Iceland|2.084249084249084...|      2.853480978414385E-8|-7.69231894165300...|   6|
|         Bahamas|4.864298454394061E-8|      5.520378159317915E-8|-6.56079704923853...|   7|
|        Colombia|1.162321861087899...|      1.702845139868463E-8|-5.40523278780563...|   8|
|         Georgia|3.789129682385098E-8|      4.254670721555529E-8|-4.65541039170431...|   9|
|          Panama| 4.39592001238537E-8|      4.745736176181264E-8|-3.49816163795893...|  10|
+---------------+--------------------+--------------------------+--------------------+----+
only showing top 10 rows
```

Doing better in comparison to 7 days ago

The top 10 of countries that have been doing better at containing outbreaks in their countries during these past 7 days, based on the moving averages.

In [50]:

```
df_incidence_joined_best = df_incidence_joined_best.filter((df_incidence_joined_best["Country_Region"] != 'MS Zaandam') & (df_incidence_joined_best["Country_Region"] != 'Diamond Princess'))

window = Window.orderBy(df_incidence_joined_best["Difference"])

df_incidence_joined_best = df_incidence_joined_best.withColumn("Rank", F.row_number().over(window))

df_incidence_joined_best.show(10)
```

```
+-------------+--------------------+--------------------------+--------------------+----+
|Country_Region|     Rolling_Average|Rolling_Average_7_days_ago|          Difference|Rank|
+-------------+--------------------+--------------------------+--------------------+----+
|      Tunisia| 2.71178073540617E-8|      4.385210366036887E-8|-1.67342963063071...|   1|
|      Andorra|1.478032375460512...|      1.619596201882013...|-1.41563826421500...|   2|
|      Czechia|1.583998619380861...|      1.664982040587470...|-8.09834212066094...|   3|
|      Iceland|2.084249084249084...|      2.853480978414385E-8|-7.69231894165300...|   4|
|      Bahamas|4.864298454394061E-8|      5.520378159317915E-8|-6.56079704923853...|   5|
```

```
|      Colombia|1.162321861087899...|      1.702845139868463E-8|-5.40523278780563...|   6
|
|       Georgia|3.789129682385098E-8|      4.254670721555529E-8|-4.65541039170431...|   7
|
|        Panama| 4.39592001238537E-8|      4.745736176181264E-8|-3.49816163795893...|   8
|
|       Bahrain|1.294529950200285...|      1.595793804050625...|-3.01263853850339...|   9
|
|          Oman|1.653563787087882...|      1.945929268613634...|-2.92365481525752...|  10
|
+--------------+--------------------+--------------------------+--------------------+--
--+
only showing top 10 rows
```

Doing worse in comparison to 7 days ago

The top 10 of countries that have been doing worse at containing outbreaks in their countries during these past 7 days, based on the moving averages. We note many familiar countries, such as Belgium, Spain, the US and Italy. There are several newcomers, such as Switzerland and Poland, who have indeed been getting hit hard recently. Thus, we can say that this metric has given us an adequate measure to determine how well we are doing today as compared to x amount of days ago.

In [51]:

```
df_incidence_joined_worst = df_incidence_joined.orderBy(df_incidence_joined["Difference"]
.desc())

window = Window.orderBy(df_incidence_joined_worst["Difference"].desc())

df_incidence_joined_worst = df_incidence_joined_worst.withColumn("Rank", F.row_number().
over(window))

df_incidence_joined_worst.show(10)
```

```
+--------------+--------------------+--------------------------+--------------------+--
--+
|Country_Region|     Rolling_Average|Rolling_Average_7_days_ago|          Difference|Rank
|
+--------------+--------------------+--------------------------+--------------------+--
--+
|       Belgium|3.841182140148855...|      3.219893645734710...|6.212884944141449E-8|   1
|
|   Switzerland|1.337647138209913E-7|      9.509331000732001E-8|3.867140381367128E-8|   2
|
|    Montenegro|1.012758487548012...|      6.437266351645077E-8|3.690318523835044...|   3
|
|         Spain|2.489461864479453E-7|      2.153328351851541...|3.361335126279118E-8|
4|
|            US|2.991811261316338...|      2.686670251456715...|3.051410098596231E-8|
5|
|    Luxembourg|1.484441895535931...|      1.184550206397689...|2.998916891382419E-8|   6
|
|        Jordan|9.151288230877946E-8|      6.180817953491727E-8|2.970470277386218...|   7
|
|    San Marino|7.359007602097943E-8|      4.548735930225705E-8|2.810271671872237...|   8
|
|        Poland|8.028388588457731E-8|      5.419617611558314...|2.608770976899416...|   9
|
|         Italy|7.946798056134401E-8|      5.405079375312444...|2.541718680821956...|  10
|
+--------------+--------------------+--------------------------+--------------------+--
--+
only showing top 10 rows
```