BITS Pilani

Covid 19 data Analytics

Big Data Systems (CCZG522)

Assignment - 1

Submitted By:
PRASHANT SINGH
GOUTHAM V
JALAMANCHILI RAMA SURYAM
KAKANI VARSHITHA
SRUTHI KRISHNAMURTHY

Submitted To:
PROF. ASHISH NARANG

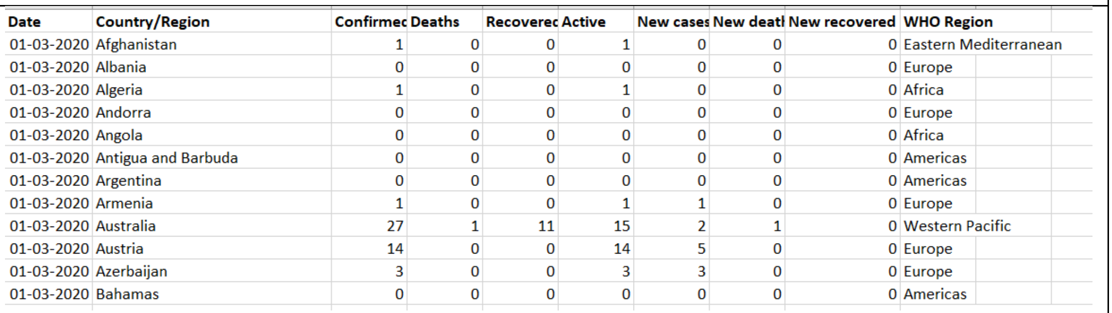
Oct 01, 2023

Contents

Table of Contents

S1-23-BDS Group 5 – Assignment 1

Prerequisite and Problem Statements for Analysis

Covid19 Data Set	https://drive.google.com/file/d/1UmigpsKC_Lwx-xrS_s6iFTOWuFchRr1F/view?usp=drive_link
Sample Data of Covid 19 Dataset (Column Values)	
Source of Covid 19 Data Set	https://www.kaggle.com/datasets/imdevskp/corona-virus-report

Date	Country/Region	Confirmec	Deaths	Recoverec	Active	New cases	New deatl	New recovered	WHO Region
01-03-2020	Afghanistan	1	0	0	1	0	0	0	Eastern Mediterranean
01-03-2020	Albania	0	0	0	0	0	0	0	Europe
01-03-2020	Algeria	1	0	0	1	0	0	0	Africa
01-03-2020	Andorra	0	0	0	0	0	0	0	Europe
01-03-2020	Angola	0	0	0	0	0	0	0	Africa
01-03-2020	Antigua and Barbuda	0	0	0	0	0	0	0	Americas
01-03-2020	Argentina	0	0	0	0	0	0	0	Americas
01-03-2020	Armenia	1	0	0	1	1	0	0	Europe
01-03-2020	Australia	27	1	11	15	2	1	0	Western Pacific
01-03-2020	Austria	14	0	0	14	5	0	0	Europe
01-03-2020	Azerbaijan	3	0	0	3	3	0	0	Europe
01-03-2020	Bahamas	0	0	0	0	0	0	0	Americas

Problem Statements and Analysis on Covid 19 Dataset

S.No	Analysis	Student Name	Student Id
1.	Provide top 10 countries for each category i.e. Recovered, deaths and confirmed cases which can be useful for WHO for there resource movements, new channels to show high level stats for bigger impact countries.	PRASHANT SINGH	2023MT03125
2.	Analyzing the recovery rate of all the unique combinations of WHO regions and Countries	GOUTHAM V	2023MT03149
3.	Group the data by dates, country/region and calculate the total number of deaths and recoveries based on date and Country	KAKANI VARSHITHA	2023MT03002
4.	Analysis on highest increase in % for confirmed cases for every country	JALAMANCHILI RAMA SURYAM	2023MT03101
5.	Temporal Analysis of COVID-19 Confirmed Cases: Tracking the Pandemic's Progression Over Time	SRUTHI KRISHNAMURTHY	2023MT03003

1. **Problem No.1 :** Provide top 10 countries for each category i.e. Recovered, deaths and confirmed cases daily changes which can be useful for WHO for there resource movements, new channels to show high level stats for bigger impact countries.

1.1 **Problem Statement**: It is very important for WHO to maintain resources to stop wider spread of Covid virus and restrict them in limit it in high impacting areas using isolation and travel bans.

1.1.1 WHO needs reports on daily basis of Top 10 countries where deaths are more so that they can ask other nations to support them with financial aids and medical facilities. (**Predictive analysis)**

1.1.2 WHO needs reports on daily basis for Top 10 countries where confirmed cases are more so that they can influence there vaccination plan to speed vaccination to reduce confirmed cases. **(Predictive analysis)**

1.1.3 WHO needs reports on daily basis for Top 10 countries where recovered cases are more so that they can notify other countries to adopt similar measures which these top 10 countries are taking. **(Prescriptive analysis)**

1.2 **Map and Reduce Diagrams**:

The diagram isn't fitting or visible in document and making it unreadable hence created top down chart steps below:

Input → Splitting → 2 Mappers→ Shuffle and Sort → Reducer Ouput

Input

```
2023-09-01,Country1,500,23,1,100,5,50,region1
2023-09-01,Country1,500,23,1,100,5,50,region2
```

Splitting (Split input into 2 Sets and two map processes can be run in parallel)

```
2023-09-30 09:56:55,843 INFO mapred.FileInputFormat: Total input files to process : 1
2023-09-30 09:56:55,930 INFO mapreduce.JobSubmitter: number of splits:2
```

Each **Mapper** will provide below Key value pairs

```
        Key: "2023-09-01, Country1"   Value: {'confirmed': 100, 'deaths': 5,
        'recovered': 50}
        Key: "2023-09-01, Country2"   Value: {'confirmed': 200, 'deaths': 10,
        'recovered': 100}
        Key: "2023-09-02, Country1"   Value: {'confirmed': 120, 'deaths': 6,
        'recovered': 60}
        Key: "2023-09-02, Country2"   Value: {'confirmed': 220, 'deaths': 11,
        'recovered': 110}
        Key: "2023-09-03, Country1"   Value: {'confirmed': 150, 'deaths': 7,
        'recovered': 75}
Key: "2023-09-03, Country2"   Value: {'confirmed': 250, 'deaths': 12,
'recovered': 125}
```

Shuffle and Sort

```
        Key: "2023-09-01"
        Value: [
          ('Country1', {'confirmed': 100, 'deaths': 5, 'recovered': 50}),
            ('Country2', {'confirmed': 200, 'deaths': 10, 'recovered': 100})
        ]

        Key: "2023-09-02"
        Value: [
            ('Country1', {'confirmed': 120, 'deaths': 6, 'recovered': 60}),
            ('Country2', {'confirmed': 220, 'deaths': 11, 'recovered': 110})
        ]

        Key: "2023-09-03"
        Value: [
            ('Country1', {'confirmed': 150, 'deaths': 7, 'recovered': 75}),
            ('Country2', {'confirmed': 250, 'deaths': 12, 'recovered': 125})
]
```

Reducer Output

```
Top 10 Countries with Highest Confirmed Cases on 2020-
09-01: Country2 (200)
Top 10 Countries with Highest Death Cases on 2020-09-01:
country 2 (10)
Top 10 Countries with Highest Recovered Cases on 2020-
09-01: country 2 (100)
```

1.3 **Map and Reduce Pseudo Code**:

1.3.1 **Mapper**: The mapper will emit the values:

The **keys** are made up of date and country, which represent the date and country respectively.

The **values** are the accumulated counts of confirmed cases, deaths, and recovered cases (confirmed, deaths, and recovered variables) for the given date and country.

```
1. Initialize variables to store the current date, current country, and accumulated counts for confirmed cases, deaths, and recovered cases.
2. Read input lines one by one and split them into fields.
3. Check if the date or country has changed compared to the previous line. If it has, we emit the accumulated data for the previous date and country as a key-value pair.
4. Reset the variables for the new date and country and start accumulating data again.
5. Accumulate the data (confirmed, deaths, and recovered) for the current date and country.
6. After processing all input lines, we emit the accumulated data for the last date and country.
```

The emit_key_value_pair function is used to format and output the key-value pair, where the key is a combination of the current date and current country, and the value is the accumulated counts of confirmed cases, deaths, and recovered cases.

1.3.2 **Reducer**: The Mapper will output data in the format expected by the Reducer (date, country, confirmed, deaths, recovered). The Reducer can then calculate the daily percentage changes based on this data.

1. Initialize variables, including current_date to keep track of the current date and country_data to store data for each country.
2. We iterate through input lines, which are assumed to be in CSV format, containing date, country, confirmed cases, deaths, and recovered cases.
3. We check if the date has changed. If it has, we perform the following steps:
    a. Calculate and print the top 10 countries with the highest death cases for the previous date.
    b. Calculate and print the top 10 countries with the highest recovered cases for the previous date.
    c. Calculate and print the top 10 countries with the highest confirmed cases for the previous date.
4. Reset the data for the new date.
5. For each input line, we update the data for the current country in the country_data dictionary.

After processing all input lines, we repeat the same calculations and printing for the last date to ensure all data is accounted for.

This program processes data and finds the top 10 countries with the highest counts of deaths, recoveries, and confirmed cases for each date.

1.4 **Map and Reduce Code**:

1.4.1 **Mapper**:

```
#!/usr/bin/env python

import sys

# Initialize variables
current_date = None
current_country = None
confirmed = 0
deaths = 0
recovered = 0

# Read data from HDFS streaming
for line in sys.stdin:
    line = line.strip()
    date, country, confirmed, deaths, recovered, active, new_cases, new_deaths,
new_recovered, who_region = line.split(',')

    # Check if the date or country has changed
    if current_date is None:
        current_date = date
        current_country = country
    if date != current_date or country != current_country:
        # Output the combined data for the previous date and country
        if current_date and current_country:

print(f"{current_date},{current_country},{confirmed},{deaths},{recovered}")

        # Reset data
        current_date = date
        current_country = country
        confirmed = 0
        deaths = 0
        recovered = 0

    # Add the data for the current date and country
    confirmed += int(new_confirmed)
    deaths += int(new_deaths)
    recovered += int(new_recovered)

# Emit key value pair
if current_date and current_country:
    print(f"{current_date},{current_country},{confirmed},{deaths},{recovered}")
```

Sample Output of Mapper:

```
2020-05-15,Andorra,761,0,8
2020-05-15,Angola,48,0,3
2020-05-15,Antigua and Barbuda,25,0,0
2020-05-15,Argentina,7479,3,112
2020-05-15,Armenia,4044,3,94
2020-05-15,Australia,7035,0,25
2020-05-15,Austria,16109,2,66
2020-05-15,Azerbaijan,2980,1,53
2020-05-15,Bahamas,96,0,0
2020-05-15,Bahrain,6583,2,287
2020-05-15,Bangladesh,20065,15,521
2020-05-15,Barbados,85,0,0
2020-05-15,Belarus,27730,5,639
2020-05-15,Belgium,54644,56,190
2020-05-15,Belize,18,0,0
2020-05-15,Benin,339,0,0
2020-05-15,Bhutan,21,0,0
2020-05-15,Bolivia,3577,12,78
2020-05-15,Bosnia and Herzegovina,2236,6,64
2020-05-15,Botswana,24,0,0
2020-05-15,Brazil,220291,963,5491
```

Full output at link: https://drive.google.com/file/d/12GR_iBodyEqqD90PBJ8jR0OLBp4Hxajc/view?usp=drive_link

1.4.2 **Reducer**:

```
#!/usr/bin/env python

import sys
```

```
# Initialize variables
current_date = None
country_data = {}

# Read Mapper's output
for line in sys.stdin:
    line = line.strip()
    date, country, confirmed, deaths, recovered = line.split(',')

    if current_date is None:
        current_date = date

    if date != current_date:
        # Print the top 10 countries with the highest death cases
        top_deaths = sorted(country_data.items(), key=lambda x: x[1]['deaths'],
reverse=True)[:10]
        for country, data in top_deaths:
            print(f"Top 10 Countries with Highest Death Cases on {current_date}:
{country} ({data['deaths']})")

        # Print the top 10 countries with the highest recovered cases
        top_recovered = sorted(country_data.items(), key=lambda x:
x[1]['recovered'], reverse=True)[:10]
        for country, data in top_recovered:
            print(f"Top 10 Countries with Highest Recovered Cases on {current_date}:
{country} ({data['recovered']})")

        # Print the top 10 countries with the highest confirmed cases
        top_confirmed = sorted(country_data.items(), key=lambda x:
x[1]['confirmed'], reverse=True)[:10]
        for country, data in top_confirmed:
            print(f"Top 10 Countries with Highest Confirmed Cases on {current_date}:
{country} ({data['confirmed']})")

        # Reset
        current_date = date
        country_data = {}

        if country not in country_data:
        country_data[country] = {'confirmed': 0, 'deaths': 0, 'recovered': 0}
    country_data[country]['confirmed'] += int(confirmed)
    country_data[country]['deaths'] += int(deaths)
    country_data[country]['recovered'] += int(recovered)

if current_date:
    top_deaths = sorted(country_data.items(), key=lambda x: x[1]['deaths'],
reverse=True)[:10]
    for country, data in top_deaths:
        print(f"Top 10 Countries with Highest Death Cases on {current_date}:
{country} ({data['deaths']})")

    top_recovered = sorted(country_data.items(), key=lambda x: x[1]['recovered'],
reverse=True)[:10]
    for country, data in top_recovered:
        print(f"Top 10 Countries with Highest Recovered Cases on {current_date}:
{country} ({data['recovered']})")
```

```
    top_confirmed = sorted(country_data.items(), key=lambda x: x[1]['confirmed'],
reverse=True)[:10]
    for country, data in top_confirmed:
        print(f"Top 10 Countries with Highest Confirmed Cases on {current_date}:
{country} ({data['confirmed']})")
```

Sample output of Reducer:

```
Top 10 Countries with Highest Death Cases on 2020-05-15: US (1661)
Top 10 Countries with Highest Death Cases on 2020-05-15: Brazil (963)
Top 10 Countries with Highest Death Cases on 2020-05-15: United Kingdom (385)
Top 10 Countries with Highest Death Cases on 2020-05-15: Mexico (290)
Top 10 Countries with Highest Death Cases on 2020-05-15: Ecuador (256)
Top 10 Countries with Highest Death Cases on 2020-05-15: Italy (242)
Top 10 Countries with Highest Death Cases on 2020-05-15: Spain (138)
Top 10 Countries with Highest Death Cases on 2020-05-15: Peru (125)
Top 10 Countries with Highest Death Cases on 2020-05-15: Sweden (117)
Top 10 Countries with Highest Death Cases on 2020-05-15: Russia (113)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Brazil (5491)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Italy (4917)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Russia (4696)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: US (4333)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Saudi Arabia (2818)

Top 10 Countries with Highest Recovered Cases on 2020-05-15: India (2289)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Turkey (2103)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Peru (1996)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Mexico (1976)
Top 10 Countries with Highest Recovered Cases on 2020-05-15: Spain (1409)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: US (1449027)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Russia (262843)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Spain (230183)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: United Kingdom
(227334)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Italy (223885)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Brazil (220291)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: France (179630)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Germany (175233)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Turkey (146457)
Top 10 Countries with Highest Confirmed Cases on 2020-05-15: Iran (122688)
```

Full Output at link:
https://drive.google.com/file/d/10PqisGljjYtWg5iEkSMf4RQMPgNBt4iQ/view?usp=drive_link

1.5 Statistics of Map reduce task

```
2023-09-30 09:56:55,843 INFO mapred.FileInputFormat: Total input files to
process : 1
2023-09-30 09:56:55,930 INFO mapreduce.JobSubmitter: number of splits:2
2023-09-30 09:57:03,602 INFO mapreduce.Job:  map 0% reduce 0%
2023-09-30 09:57:11,826 INFO mapreduce.Job:  map 50% reduce 0%
2023-09-30 09:57:12,831 INFO mapreduce.Job:  map 100% reduce 0%
2023-09-30 09:57:19,909 INFO mapreduce.Job:  map 100% reduce 100%
```

```
File System Counters
                FILE: Number of bytes read=1123225
                FILE: Number of bytes written=2974387
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1861208
                HDFS: Number of bytes written=421410
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
Job Counters
                Killed map tasks=1
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=25672
                Total time spent by all reduces in occupied slots (ms)=13959
                Total time spent by all map tasks (ms)=12836
                Total time spent by all reduce tasks (ms)=4653
                Total vcore-milliseconds taken by all map tasks=12836
                Total vcore-milliseconds taken by all reduce tasks=4653
                Total megabyte-milliseconds taken by all map tasks=26288128
                Total megabyte-milliseconds taken by all reduce tasks=14294016
        Map-Reduce Framework
                Map input records=35156
                Map output records=35156
                Map output bytes=1052907
                Map output materialized bytes=1123231
                Input split bytes=174
                Combine input records=0
                Combine output records=0
                Reduce input groups=35156
                Reduce shuffle bytes=1123231
                Reduce input records=35156
                Reduce output records=5640
                Spilled Records=70312
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=289
                CPU time spent (ms)=4330
                Physical memory (bytes) snapshot=1624854528
                Virtual memory (bytes) snapshot=10727583744
                Total committed heap usage (bytes)=1474822144
                Peak Map Physical memory (bytes)=716967936
                Peak Map Virtual memory (bytes)=3006988288
                Peak Reduce Physical memory (bytes)=191455232
                Peak Reduce Virtual memory (bytes)=4713611264
                File Input Format Counters
                        Bytes Read=1861034
                File Output Format Counters
                        Bytes Written=421410
2023-09-30 09:57:20,006 INFO streaming.StreamJob: Output directory:
/cov1/output13
```

2. Not submitted the documentation of analysis until 11PM on 01st Oct – Problem No 2 (Goutham V)

3. **Problem No.3:** Group the data by dates, country/region and calculate the total number of deaths and recoveries based on date and Country.

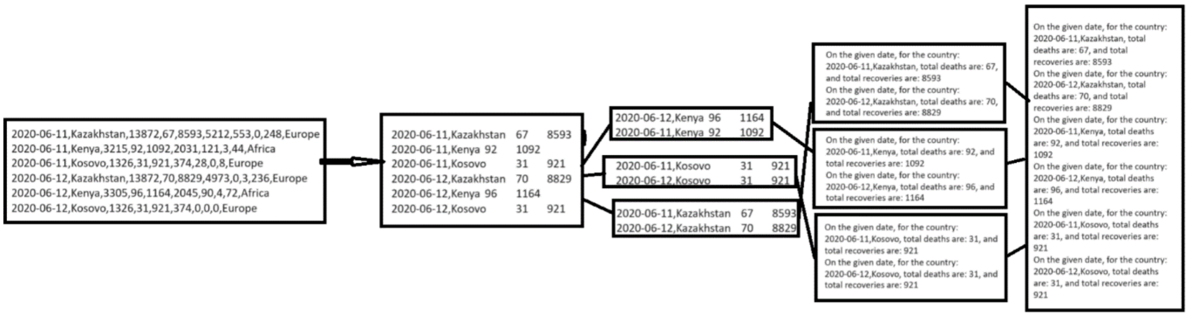
3.1 Problem Statement: It is very important for WHO to maintain resources to stop wider spread of Covid virus and restrict them in limit it in high impacting areas using isolation and travel bans.

3.1.1 WHO needs reports on daily basis of deaths and recoveries so that they can predict the requirement for medical resources like beds, staffing, medicines, etc.(**Predictive analysis)**

3.1.2 WHO needs reports on daily basis of deaths and recoveries where deadths are increasing so that they can prioritize the vaccine supply. **(Predictive analysis)**

3.1.3 WHO needs reports on daily basis of deaths and recoveries where recovered cases are more so that they can notify other countries to adopt similar measures which are being followed by the recovering countries. **(Prescriptive analysis)**

3.2 Map and Reduce Diagrams:



```
2020-06-11,Kazakhstan,13872,67,8593,5212,553,0,248,Europe
2020-06-11,Kenya,3215,92,1092,2031,121,3,44,Africa
2020-06-11,Kosovo,1326,31,921,374,28,0,8,Europe
2020-06-12,Kazakhstan,13872,70,8829,4973,0,3,236,Europe
2020-06-12,Kenya,3305,96,1164,2045,90,4,72,Africa
2020-06-12,Kosovo,1326,31,921,374,0,0,0,Europe
```

```
2020-06-11,Kazakhstan   67    8593
2020-06-11,Kenya  92    1092
2020-06-11,Kosovo       31    921
2020-06-12,Kazakhstan   70    8829
2020-06-12,Kenya  96    1164
2020-06-12,Kosovo       31    921
```

```
2020-06-12,Kenya  96      1164
2020-06-11,Kenya  92      1092
```

```
2020-06-11,Kosovo         31     921
2020-06-12,Kosovo         31     921
```

```
2020-06-11,Kazakhstan   67     8593
2020-06-12,Kazakhstan   70     8829
```

On the given date, for the country:
2020-06-11,Kazakhstan, total deaths are: 67,
and total recoveries are: 8593
On the given date, for the country:
2020-06-12,Kazakhstan, total deaths are: 70,
and total recoveries are: 8829

On the given date, for the country:
2020-06-11,Kenya, total deaths are: 92, and
total recoveries are: 1092
On the given date, for the country:
2020-06-12,Kenya, total deaths are: 96, and
total recoveries are: 1164

On the given date, for the country:
2020-06-11,Kosovo, total deaths are: 31, and
total recoveries are: 921
On the given date, for the country:
2020-06-12,Kosovo, total deaths are: 31, and
total recoveries are: 921

On the given date, for the country:
2020-06-11,Kazakhstan, total
deaths are: 67, and total
recoveries are: 8593
On the given date, for the country:
2020-06-12,Kazakhstan, total
deaths are: 70, and total
recoveries are: 8829
On the given date, for the country:
2020-06-11,Kenya, total deaths
are: 92, and total recoveries are:
1092
On the given date, for the country:
2020-06-12,Kenya, total deaths
are: 96, and total recoveries are:
1164
On the given date, for the country:
2020-06-11,Kosovo, total deaths
are: 31, and total recoveries are:
921
On the given date, for the country:
2020-06-12,Kosovo, total deaths
are: 31, and total recoveries are:
921

3.3 Map and Reduce Pseudo Code:

3.3.1 **Mapper**: The mapper will emit the values:
The **keys** are made up of date and country.
The **values** are the accumulated counts of confirmed cases, deaths, and recovered cases (deaths, and recovered variables) for the given date and country.

1. Initialize variables to store the current date, current country, and accumulated counts for confirmed cases, deaths, and recovered cases.
2. Read input lines one by one and split them into fields.
3. Check if the date or country has changed compared to the previous line. If it has, we emit the accumulated data for the previous date and country as a key-value pair.
4. Reset the variables for the new date and country and start accumulating data again.
5. Accumulate the data (country, deaths, and recovered) for the current date and country.
6. After processing all input lines, we emit the accumulated data for the last date and country.

The emit_key_value_pair function is used to format and output the key-value pair, where the key is a combination of the current date and current country, and the value is the accumulated counts of deaths and recovered cases for particular date and country.

3.3.2 Reducer: The Mapper will output data in the format expected by the Reducer (date, country, deaths, recovered). The Reducer can then calculate the daily percentage changes based on this data.

1. Initialize variables, including current_date to keep track of the current date and country_data to store data for each country.
2. We iterate through input lines, which are assumed to be in CSV format, containing date, country, confirmed cases, deaths, and recovered cases.
3. We check if the date has changed. If it has, we perform the following steps:
    a. Calculate and print the Who region with the highest death cases for the previous date.
    b. Calculate and print the who region with the highest recovered cases for the previous date.
4. Reset the data for the new date.
5. For each input line, we update the data for the current country in the country_data dictionary.

After processing all input lines, we repeat the same calculations and printing for the last date to ensure all data is accounted for.

This program processes data and finds the region with the highest counts of deaths, recoveries for each date.

3.4 Map and Reduce Code:

3.4.1 Mapper:

```

#!/usr/bin/env python

import sys

```

```
# Input comes from STDIN (standard input)
for lines in sys.stdin:
    # Remove leading and trailing whitespace and split the line into fields
    lines = lines.strip()
    input_fields = lines.split(',')

    # Check if the line has the expected number of fields
    if len(input_fields) == 10:
        date, country, confirmed, deaths, recovered, active, new_cases, new_deaths, new_recovered, who_region = input_fields

        # Emit key-value pairs for grouping by date and country/region
        # Key: Date, Country/Region
        # Value: Deaths,Recovered
        print(f"{date},{country}\t{deaths}\t{recovered}")

```

Sample Output of Mapper:

```
2020-01-22,Armenia    0       0
2020-01-22,Australia  0       0
2020-01-22,Austria    0       0
2020-01-22,Azerbaijan 0       0
2020-01-22,Bahamas    0       0
2020-01-22,Bahrain    0       0
2020-01-22,Bangladesh 0       0
2020-01-22,Barbados   0       0
2020-01-22,Belarus    0       0
2020-01-22,Belgium    0       0
2020-01-22,Belize     0       0
2020-01-22,Benin      0       0
2020-01-22,Bhutan     0       0
```

Full Mapper output at link:
https://drive.google.com/file/d/17BazZ1-bg7KgQqIGQ6v0YHnPBE9Nueke/view?usp=drive_link

3.4.2 **Reducer**:

```
#!/usr/bin/env python

import sys

current_date_country = None
total_deaths = 0
total_recoveries = 0

# Input comes from STDIN (standard input)
for lines in sys.stdin:
    # Remove leading and trailing whitespace
    lines = lines.strip()

    # Split the line into key and values
    date_country, deaths, recoveries = lines.split('\t')
    # Convert deaths and recoveries to integers
    deaths = int(deaths)
    recoveries = int(recoveries)

    # If the date and country change (new date and country)
    if current_date_country != date_country:
        # Print the total deaths and recoveries for the previous date and country
        if current_date_country:
            print(f"On the given date, for the country: {current_date_country}, total deaths are: {total_deaths}, and total recoveries are: {total_recoveries}")

        # Reset the totals and update the current date and country
        current_date_country = date_country
        total_deaths = 0
        total_recoveries = 0

    # Update the totals
    total_deaths += deaths
    total_recoveries += recoveries

# Print the totals for the last date and country
if current_date_country:
    print(f"On the given date, for the country: {current_date_country}, total deaths are: {total_deaths}, and total recoveries are: {total_recoveries}")
```

Sample output of Reducer:

```
On the given date, for the country: 2020-06-04,Netherlands, total deaths are: 6009, and total recoveries are: 173

On the given date, for the country: 2020-06-04,New Zealand, total deaths are: 22, and total recoveries are: 1481
On the given date, for the country: 2020-06-04,Nicaragua, total deaths are: 46, and total recoveries are: 370
On the given date, for the country: 2020-06-04,Niger, total deaths are: 65, and total recoveries are: 860
On the given date, for the country: 2020-06-04,Nigeria, total deaths are: 323, and total recoveries are: 3535
On the given date, for the country: 2020-06-04,North Macedonia, total deaths are: 147, and total recoveries are: 1621

On the given date, for the country: 2020-06-04,Norway, total deaths are: 238, and total recoveries are: 8138
On the given date, for the country: 2020-06-04,Oman, total deaths are: 67, and total recoveries are: 3451
On the given date, for the country: 2020-06-04,Pakistan, total deaths are: 1838, and total recoveries are: 31198
On the given date, for the country: 2020-06-04,Panama, total deaths are: 363, and total recoveries are: 9619
On the given date, for the country: 2020-06-04,Papua New Guinea, total deaths are: 0, and total recoveries are: 8

On the given date, for the country: 2020-06-04,Paraguay, total deaths are: 11, and total recoveries are: 511
On the given date, for the country: 2020-06-04,Peru, total deaths are: 5031, and total recoveries are: 76228
On the given date, for the country: 2020-06-04,Philippines, total deaths are: 984, and total recoveries are: 4248
On the given date, for the country: 2020-06-04,Poland, total deaths are: 1117, and total recoveries are: 12227
```

Full Output at link:
https://drive.google.com/file/d/1NpO3l8BMTTi0x63hd9mE-ska256dRUhg/view?usp=drive_link

3.5 Statistics of Map reduce task

```
                2023-10-01 07:34:53,971 WARN streaming.StreamJob: -file
option is deprecated, please use generic option -files instead.
packageJobJar: [mapper.py, reducer.py, /tmp/hadoop-
unjar4065329102241534669/] [] /tmp/streamjob3045648253766261495.jar
tmpDir=null
2023-10-01 07:34:54,981 INFO client.RMProxy: Connecting to ResourceManager
at master/172.31.6.106:8032
2023-10-01 07:34:55,251 INFO client.RMProxy: Connecting to ResourceManager
at master/172.31.6.106:8032
2023-10-01 07:34:55,456 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path: /tmp/hadoop-
yarn/staging/centos/.staging/job_1696144539938_0004
2023-10-01 07:34:55,796 INFO mapred.FileInputFormat: Total input files to
process : 1
2023-10-01 07:34:55,872 INFO mapreduce.JobSubmitter: number of splits:2
2023-10-01 07:34:56,048 INFO mapreduce.JobSubmitter: Submitting tokens for
job: job_1696144539938_0004
2023-10-01 07:34:56,050 INFO mapreduce.JobSubmitter: Executing with tokens:
[]
2023-10-01 07:34:56,261 INFO conf.Configuration: resource-types.xml not found
2023-10-01 07:34:56,261 INFO resource.ResourceUtils: Unable to find 'resource-
types.xml'.
2023-10-01 07:34:56,330 INFO impl.YarnClientImpl: Submitted application
application_1696144539938_0004
2023-10-01 07:34:56,373 INFO mapreduce.Job: The url to track the job:
http://master:8088/proxy/application_1696144539938_0004/
2023-10-01 07:34:56,375 INFO mapreduce.Job: Running job:
job_1696144539938_0004
2023-10-01 07:35:02,568 INFO mapreduce.Job: Job job_1696144539938_0004
running in uber mode : false
2023-10-01 07:35:02,569 INFO mapreduce.Job:  map 0% reduce 0%
2023-10-01 07:35:09,678 INFO mapreduce.Job:  map 100% reduce 0%
2023-10-01 07:35:16,716 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-01 07:35:17,731 INFO mapreduce.Job: Job job_1696144539938_0004
completed successfully
2023-10-01 07:35:17,820 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=1007519
                FILE: Number of bytes written=2743086
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1861234
                HDFS: Number of bytes written=3855149
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=16092
                Total time spent by all reduces in occupied slots (ms)=15180
                Total time spent by all map tasks (ms)=8046
                Total time spent by all reduce tasks (ms)=5060
                Total vcore-milliseconds taken by all map tasks=8046
                Total vcore-milliseconds taken by all reduce tasks=5060
                Total megabyte-milliseconds taken by all map tasks=16478208
                Total megabyte-milliseconds taken by all reduce tasks=15544320
        Map-Reduce Framework
                Map input records=35156
                Map output records=35156
                Map output bytes=937201
                Map output materialized bytes=1007525
                Input split bytes=200
                Combine input records=0
                Combine output records=0
                Reduce input groups=35156
                Reduce shuffle bytes=1007525
                Reduce input records=35156
                Reduce output records=35156
                Spilled Records=70312
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=251
                CPU time spent (ms)=4960
                Physical memory (bytes) snapshot=1657139200
                Virtual memory (bytes) snapshot=10748768256
                Total committed heap usage (bytes)=1597505536
                Peak Map Physical memory (bytes)=734896128
                Peak Map Virtual memory (bytes)=3018330112
                Peak Reduce Physical memory (bytes)=187535360
                Peak Reduce Virtual memory (bytes)=4713205760
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1861034
        File Output Format Counters
```

```
                Bytes Written=3855149
2023-10-01 07:35:17,820 INFO streaming.StreamJob: Output directory:
/mydata/assignment/output_final
```

4. Not submitted the document of analysis until 11 PM on 01st Oct – Problem no 4 (JALAMANCHILI RAMA SURYAM)

5. **Problem no. 5:** Temporal Analysis of COVID-19 Confirmed Cases: Tracking the Pandemic's Progression Over Time

   a. **Problem Statement:** The COVID-19 pandemic had significantly impacted societies, economies, and healthcare systems worldwide. Timely and data-driven analysis is crucial for understanding the dynamics of the pandemic, assessing its severity, and informing public health decisions. In this context, the problem at hand is to perform a comprehensive temporal analysis of COVID-19 confirmed cases, with a focus on tracking the pandemic's progression over time.

      Scope of Analysis:

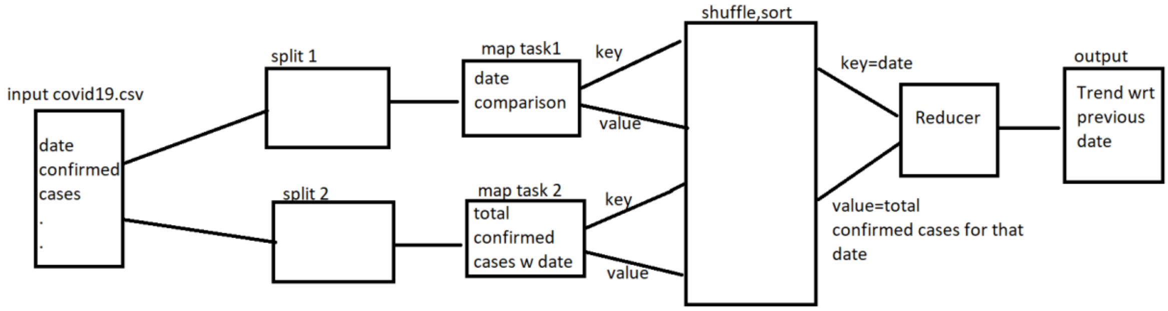
      i. **Data Source:** The analysis will use a dataset containing daily COVID-19 statistics, including the date, country, and the number of new confirmed cases.

      ii. **Temporal Trend Analysis:** The analysis will focus on understanding the temporal trends in COVID-19 confirmed cases. It will involve tracking the daily and cumulative confirmed cases over time.

      iii. **Key Metrics:**

         1. Daily Confirmed Cases: Tracking the daily increase in confirmed cases.
         2. Total Confirmed Cases: Calculating the cumulative total of confirmed cases over time.
         3. Daily Trend: Identifying whether the number of confirmed cases is increasing, decreasing, or remaining stable on a daily basis.

   b. **Map and Reduce Diagram:**



   c. **Map and Reduce Pseudo Code**:

      i. **Mapper:**

         1. input_data represents the input data stream containing lines of COVID-19 data.
         2. split(each_line, ',') is a function that takes each line and splits it into individual columns using a comma as the delimiter.

3. The script iterates through each line of data, skipping the header row.
4. It checks if the date has changed. If it has, it means that we have completed processing data for the current date, so we emit the total confirmed cases for that date.
5. The script then resets the data for the new date and continues accumulating confirmed cases.
6. Finally, after processing all the data, it emits the accumulated data for the last date.

   The output of this mapper will contain key-value pairs where the key is the date, and the value is the total confirmed cases for that date. This data can be further processed by the reducer or used for temporal analysis of COVID-19 confirmed cases.

ii. Reducer:

1. input_data represents the input data stream containing key-value pairs where the key is the date, and the value is the total confirmed cases for that date.
2. split(each_line, '\t') is a function that takes each line and splits it into two parts using a tab character as the delimiter.
3. The script iterates through each line of data and checks if the date has changed. If it has, it emits a line containing the date and the total confirmed cases for that date.
4. The script then resets the data for the new date and continues accumulating the total confirmed cases.
5. Finally, after processing all the data, it emits the accumulated data for the last date.

   The output of this reducer script will contain lines with the date and the total confirmed cases for that date, providing a temporal summary of COVID-19 confirmed cases over time.

d. Map and Reduce Code:

i. Mapper:

link: https://drive.google.com/file/d/16DlkTL9L-7ddsOnuoECGdCTlAvSNciDF/view?usp=drive_link

```
#!/usr/bin/env python

import sys

# Initialize variables to store data
current_date = None
previous_confirmed = None

# Read data from standard input (HDFS streaming)
for line in sys.stdin:
    line = line.strip()
    date, _, new_confirmed, _, _, _, _, _, _, _ =
line.split(',')
```

```
    # Skip the header row
    if date == "date":
        continue

    # Check if the date has changed
    if current_date is None:
        current_date = date
        previous_confirmed = int(new_confirmed)
        continue

    # Calculate the daily confirmed cases
    daily_confirmed = int(new_confirmed) -
previous_confirmed

    # Emit key-value pairs with date as the key and
daily confirmed cases as the value
    print(f"{current_date}\t{daily_confirmed}")

    # Update variables for the next iteration
    current_date = date
    previous_confirmed = int(new_confirmed)
```

ii. Reducer:

link: https://drive.google.com/file/d/1e5_1N06DRQg6U8u-AirRp0BiAI4m3vf-/view?usp=drive_link

```
#!/usr/bin/env python

import sys

# Initialize variables to store data
current_date = None
previous_confirmed = None

# Read data from standard input (HDFS streaming)
for line in sys.stdin:
    line = line.strip()
    date, _, new_confirmed, _, _, _, _, _, _, _ =
line.split(',')

    # Skip the header row
    if date == "date":
        continue

    # Check if the date has changed
    if current_date is None:
        current_date = date
        previous_confirmed = int(new_confirmed)
        continue

    # Calculate the daily confirmed cases
    daily_confirmed = int(new_confirmed) -
previous_confirmed

    # Emit key-value pairs with date as the key and
daily confirmed cases as the value
    print(f"{current_date}\t{daily_confirmed}")

    # Update variables for the next iteration
    current_date = date
    previous_confirmed = int(new_confirmed)
```

iii. **Output:**

link:
https://drive.google.com/file/d/1qSTpFbKImCDjuhOj5LV4EKG4F8tzrDTk/view?usp=drive_link

Sample output:

```
Date: 2020-04-03, Total Daily Confirmed Cases: 18,
Daily Trend: -272
Date: 2020-04-04, Total Daily Confirmed Cases: 50,
Daily Trend: -290
Date: 2020-04-05, Total Daily Confirmed Cases: 18,
Daily Trend: -340
Date: 2020-04-06, Total Daily Confirmed Cases: 56,
Daily Trend: -357
Date: 2020-04-07, Total Daily Confirmed Cases: 21,
Daily Trend: -412
Date: 2020-04-08, Total Daily Confirmed Cases: 40,
Daily Trend: -433
Date: 2020-04-09, Total Daily Confirmed Cases: 37,
Daily Trend: -473
Date: 2020-04-10, Total Daily Confirmed Cases: 34,
Daily Trend: -508
Date: 2020-04-11, Total Daily Confirmed Cases: 52,
Daily Trend: -541
```

e. **Statistics of Map Reduce Task:**

```
2023-09-30 13:41:53,154 INFO mapred.FileInputFormat: Total
input files to process : 1
2023-09-30 13:41:53,300 INFO mapreduce.JobSubmitter:
number of splits:2
2023-09-30 13:41:53,893 INFO mapreduce.JobSubmitter:
Submitting tokens for job: job_1696077914905_0005
2023-09-30 13:41:53,895 INFO mapreduce.JobSubmitter:
Executing with tokens: []
2023-09-30 13:41:54,107 INFO conf.Configuration: resource-
types.xml not found
2023-09-30 13:41:54,107 INFO resource.ResourceUtils:
Unable to find 'resource-types.xml'.
2023-09-30 13:41:54,194 INFO impl.YarnClientImpl:
```

```
Submitted application application_1696077914905_0005
2023-09-30 13:41:54,238 INFO mapreduce.Job: The url to
track the job:
http://master:8088/proxy/application_1696077914905_0005/
2023-09-30 13:41:54,239 INFO mapreduce.Job: Running job:
job_1696077914905_0005
2023-09-30 13:42:01,357 INFO mapreduce.Job: Job
job_1696077914905_0005 running in uber mode : false
2023-09-30 13:42:01,358 INFO mapreduce.Job:  map 0% reduce
0%
2023-09-30 13:42:07,473 INFO mapreduce.Job:  map 100%
reduce 0%
2023-09-30 13:42:14,510 INFO mapreduce.Job:  map 100%
reduce 100%
2023-09-30 13:42:15,526 INFO mapreduce.Job: Job
job_1696077914905_0005 completed successfully
2023-09-30 13:42:15,620 INFO mapreduce.Job: Counters: 54
      File System Counters
            FILE: Number of bytes read=618047
            FILE: Number of bytes written=1964475
            FILE: Number of read operations=0
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=1861250
            HDFS: Number of bytes written=13195
            HDFS: Number of read operations=11
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=2
            HDFS: Number of bytes read erasure-coded=0
      Job Counters
            Launched map tasks=2
            Launched reduce tasks=1
            Data-local map tasks=2
            Total time spent by all maps in occupied slots
(ms)=16304
            Total time spent by all reduces in occupied
slots (ms)=13827
            Total time spent by all map tasks (ms)=8152
            Total time spent by all reduce tasks (ms)=4609
            Total vcore-milliseconds taken by all map
tasks=8152
            Total vcore-milliseconds taken by all reduce
tasks=4609
            Total megabyte-milliseconds taken by all map
tasks=16695296
            Total megabyte-milliseconds taken by all reduce
tasks=14158848
      Map-Reduce Framework
            Map input records=35156
            Map output records=35154
            Map output bytes=547733
            Map output materialized bytes=618053
            Input split bytes=216
            Combine input records=0
            Combine output records=0
            Reduce input groups=188
            Reduce shuffle bytes=618053
```

```
			Reduce input records=35154
			Reduce output records=188
			Spilled Records=70308
			Shuffled Maps =2
			Failed Shuffles=0
			Merged Map outputs=2
			GC time elapsed (ms)=261
			CPU time spent (ms)=4810
			Physical memory (bytes) snapshot=1657196544
			Virtual memory (bytes) snapshot=10751795200
			Total committed heap usage (bytes)=1606942720
			Peak Map Physical memory (bytes)=735891456
			Peak Map Virtual memory (bytes)=3018297344
			Peak Reduce Physical memory (bytes)=189288448
			Peak Reduce Virtual memory (bytes)=4715679744
		Shuffle Errors
			BAD_ID=0
			CONNECTION=0
			IO_ERROR=0
			WRONG_LENGTH=0
			WRONG_MAP=0
			WRONG_REDUCE=0
		File Input Format Counters
			Bytes Read=1861034
		File Output Format Counters
			Bytes Written=13195
2023-09-30 13:42:15,620 INFO streaming.StreamJob: Output
directory: /user/skrishnamurthy/output4
```