COVID-19 Data Visualization

Using Tableau and Spark

Team 1:
Jacequai De Magalhaes, Brandon Cho, Jack Nguyen, Newyork Her, and Aaron Schomer

MVP

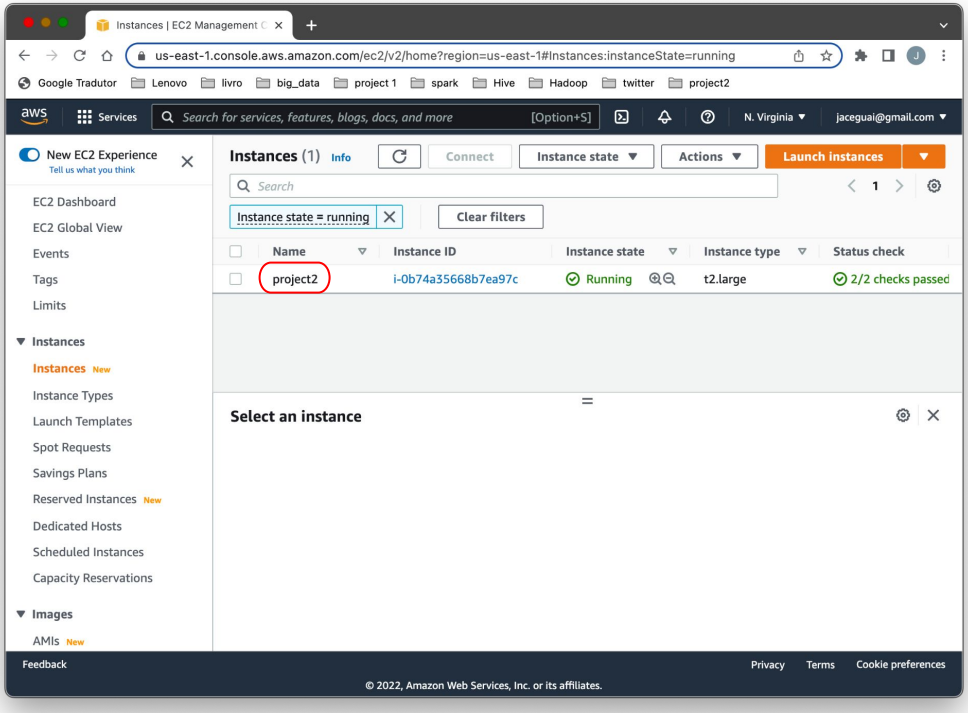
- Created a Spark Application, written in Scala, that uses the CLI for user interaction.
- Users can create an account and login to the program.
  - Accounts persists after sessions.
  - Passwords are automatically encrypted using **bCrypt API.**
  - Input validation for correct/present username and password.
  - Basic and Admin privileges. Admins have access to user management and log viewing.
- System will log each action done during sessions within a Spark SQL table.
  - Also set logging level to error to hide unnecessary log information during runtime
- Users are able to run queries on COVID data files provided by the program.
- Produced a .jar file for the project, allowing for the application to be ran in Ubuntu and AWS.
- Utilized Tableau and Zeppelin to visualize results and analyze trends.

Stretch Goals

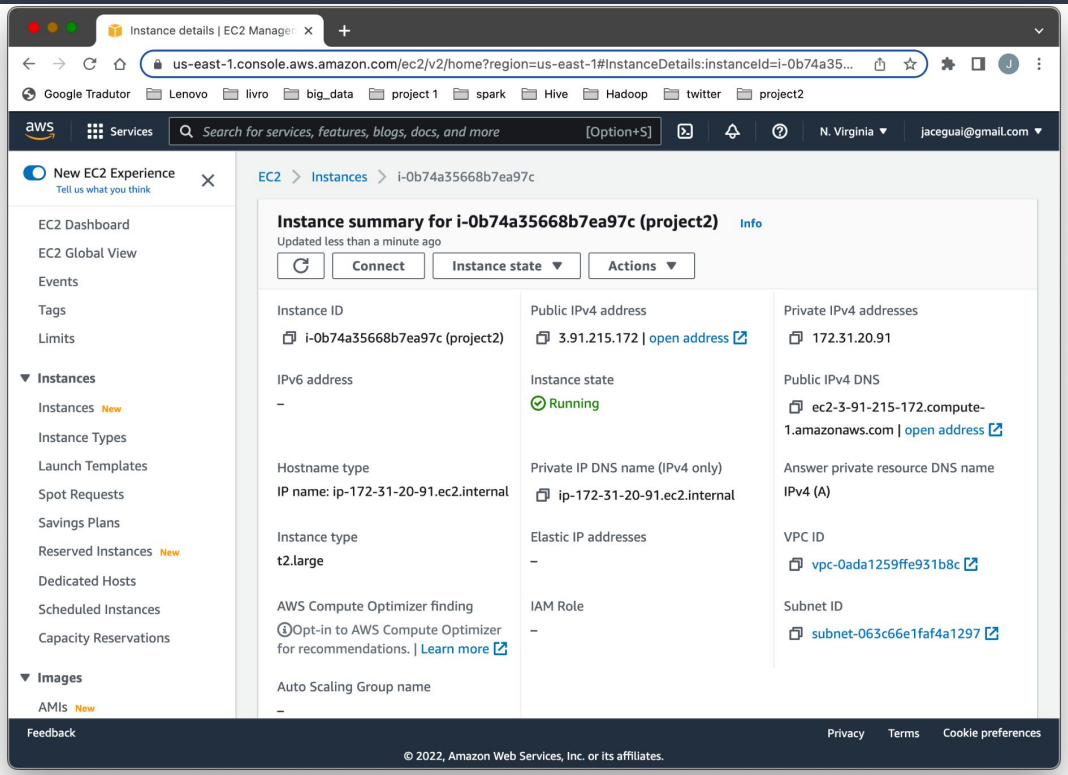
- Finding a relation between COVID data and other kinds of data, i.e., GDP and Population.
- Use Zeppelin for data visualization.
- Using AWS or Azure to run our .jar file.
- Visualizing Spatial Data in Tableau.

Here is our Demo!

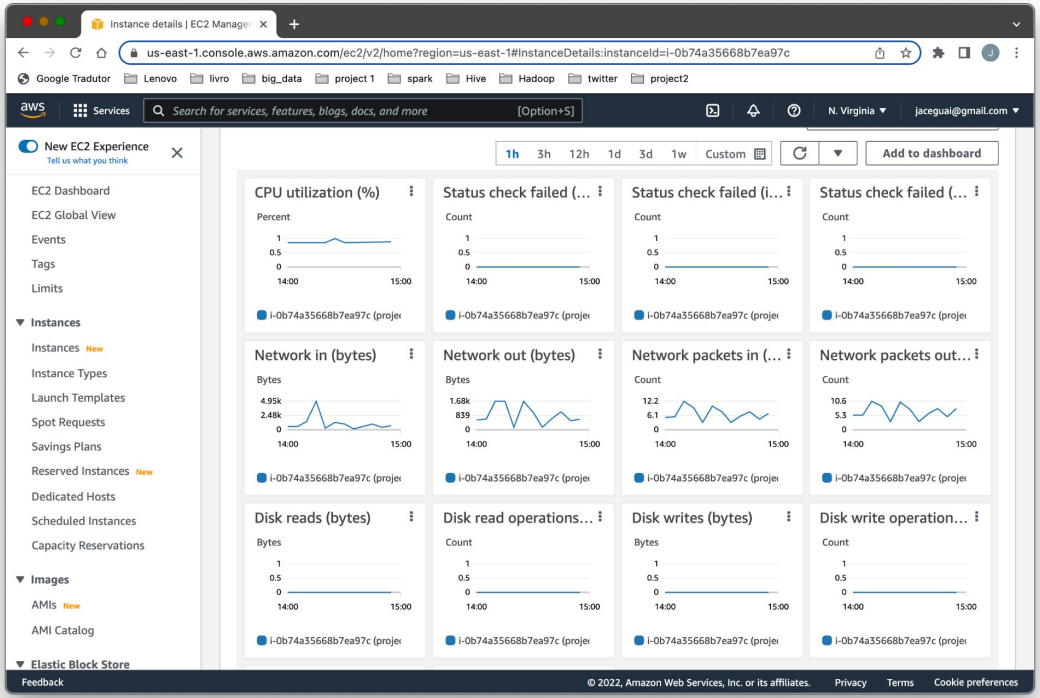
AWS EC2


Instances | EC2 Management
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:instanceState=running
Google Tradutor
Lenovo
livro
big_data
project 1
spark
Hive
Hadoop
twitter
project2
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
jaceguai@gmail.com
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Search
1
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
project2
i-0b74a35668b7ea97c
Running
t2.large
2/2 checks passed
Select an instance
Feedback
Privacy
Terms
Cookie preferences
© 2022, Amazon Web Services, Inc. or its affiliates.

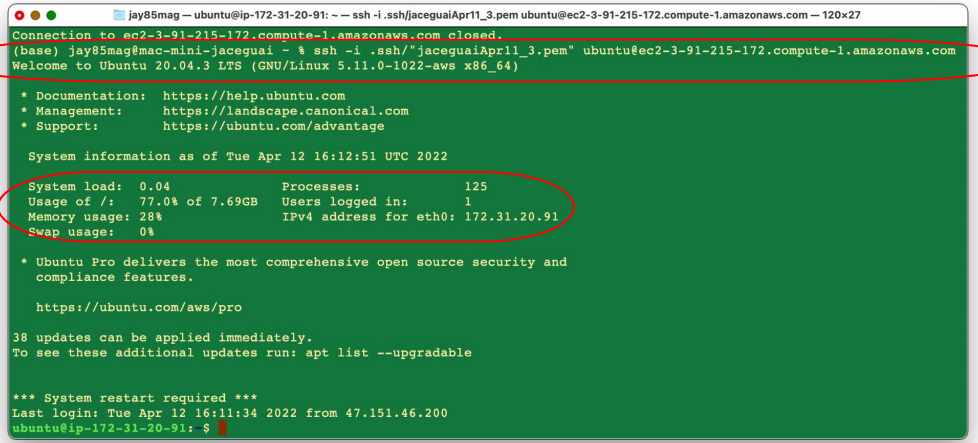
AWS EC2


Instance details | EC2 Manager
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-0b74a35...
Google Tradutor
Lenovo
livro
big_data
project 1
spark
Hive
Hadoop
twitter
project2
aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
jaceguai@gmail.com
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
EC2 > Instances > i-0b74a35668b7ea97c
Instance summary for i-0b74a35668b7ea97c (project2) Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-0b74a35668b7ea97c (project2)
IPv6 address
–
Hostname type
IP name: ip-172-31-20-91.ec2.internal
Instance type
t2.large
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
Public IPv4 address
3.91.215.172 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-20-91.ec2.internal
Elastic IP addresses
–
IAM Role
–
Private IPv4 addresses
172.31.20.91
Public IPv4 DNS
ec2-3-91-215-172.compute-1.amazonaws.com | open address
Answer private resource DNS name
IPv4 (A)
VPC ID
vpc-0ada1259ffe931b8c
Subnet ID
subnet-063c66e1faf4a1297
Feedback
Privacy
Terms
Cookie preferences
© 2022, Amazon Web Services, Inc. or its affiliates.

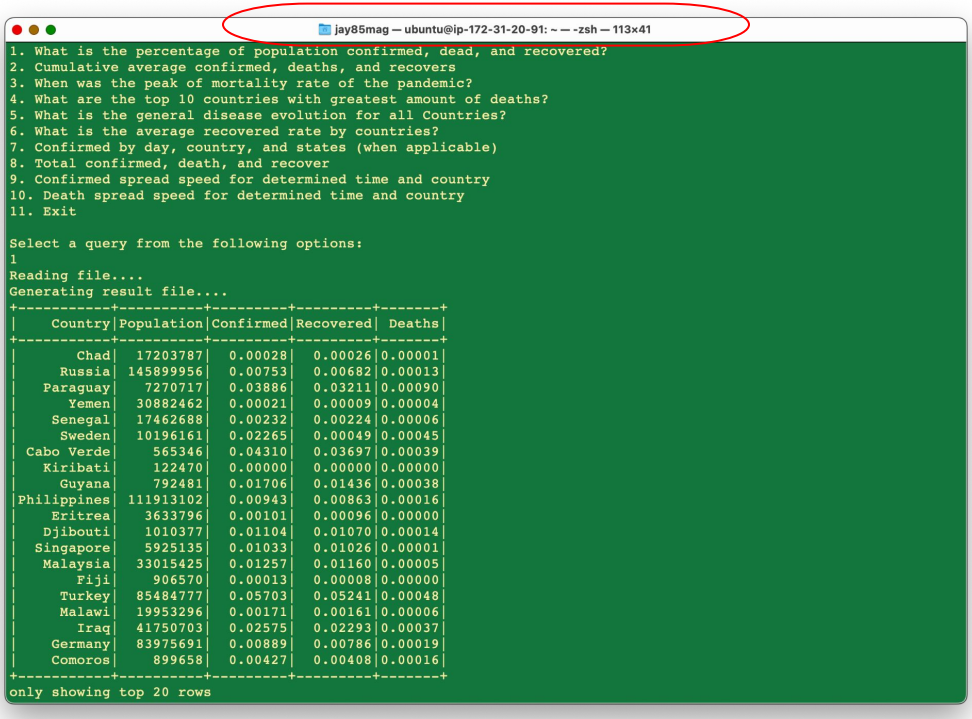
AWS EC2


Instance details | EC2 Manager
us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-0b74a35668b7ea97c
Google Tradutor
Lenovo
livro
big_data
project 1
spark
Hive
Hadoop
twitter
project2
aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia
jaceguai@gmail.com
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
1h
3h
12h
1d
3d
1w
Custom
Add to dashboard
CPU utilization (%)
Status check failed (...
Status check failed (i...
Status check failed (...
Network in (bytes)
Network out (bytes)
Network packets in (...
Network packets out...
Disk reads (bytes)
Disk read operations...
Disk writes (bytes)
Disk write operation...
Feedback
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

AWS EC2

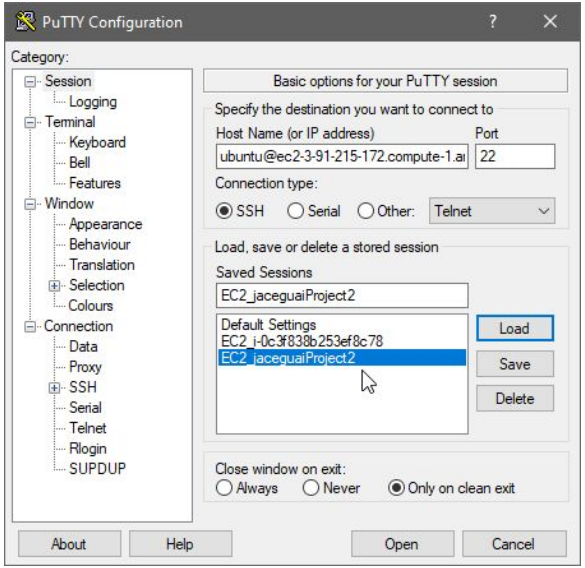

jay85mag — ubuntu@ip-172-31-20-91: ~ — ssh -i .ssh/jaceguaiApr11_3.pem ubuntu@ec2-3-91-215-172.compute-1.amazonaws.com — 120×27
Connection to ec2-3-91-215-172.compute-1.amazonaws.com closed.
(base) jay85mag@mac-mini-jaceguai ~ % ssh -i .ssh/"jaceguaiApr11_3.pem" ubuntu@ec2-3-91-215-172.compute-1.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Tue Apr 12 16:12:51 UTC 2022
System load: 0.04 Processes: 125
Usage of /: 77.0% of 7.69GB Users logged in: 1
Memory usage: 28% IPv4 address for eth0: 172.31.20.91
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.
https://ubuntu.com/aws/pro
38 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
*** System restart required ***
Last login: Tue Apr 12 16:11:34 2022 from 47.151.46.200
ubuntu@ip-172-31-20-91: $

AWS EC2

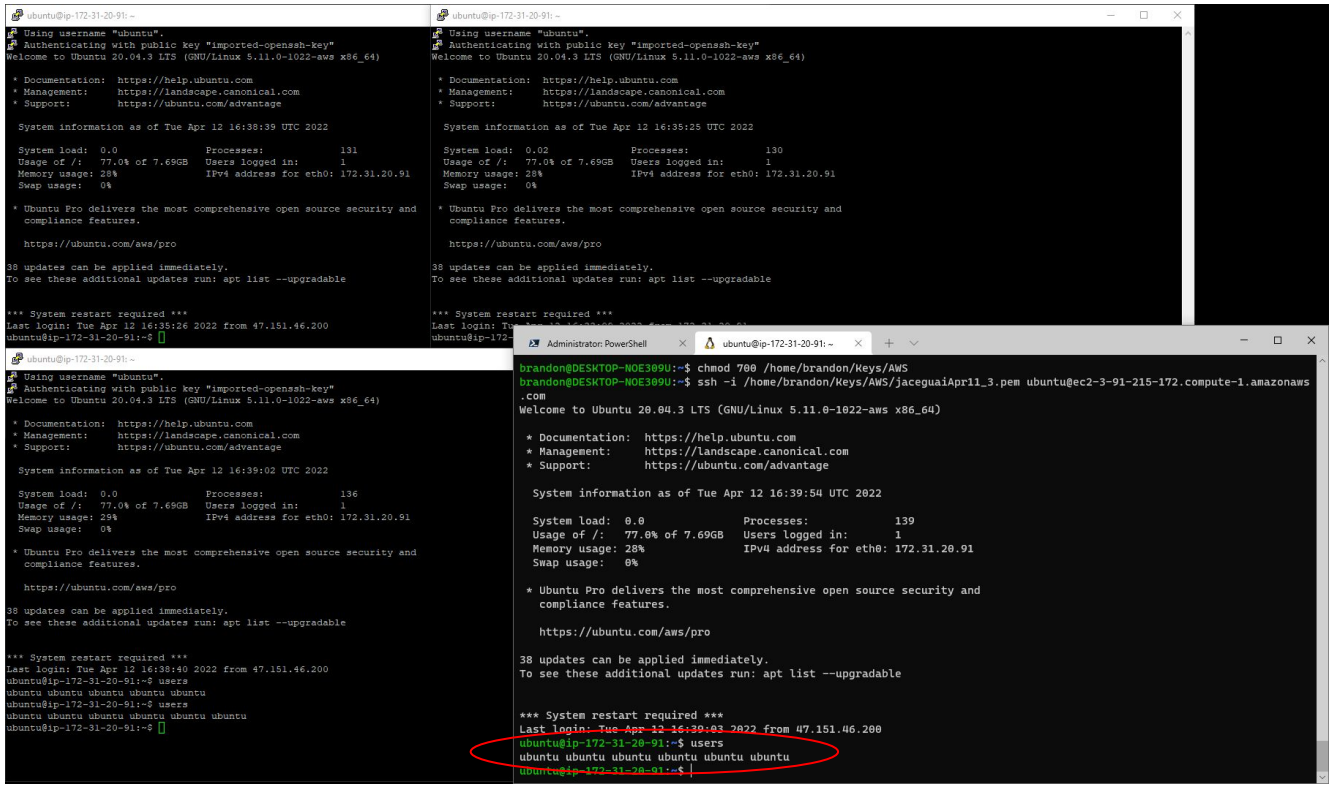

jay85mag — ubuntu@ip-172-31-20-91: ~ — -zsh — 113×41
1. What is the percentage of population confirmed, dead, and recovered?
2. Cumulative average confirmed, deaths, and recovers
3. When was the peak of mortality rate of the pandemic?
4. What are the top 10 countries with greatest amount of deaths?
5. What is the general disease evolution for all Countries?
6. What is the average recovered rate by countries?
7. Confirmed by day, country, and states (when applicable)
8. Total confirmed, death, and recover
9. Confirmed spread speed for determined time and country
10. Death spread speed for determined time and country
11. Exit
Select a query from the following options:
1
Reading file....
Generating result file....
+-----------+----------+---------+---------+-------+
| Country|Population|Confirmed|Recovered| Deaths|
+-----------+----------+---------+---------+-------+
Chad	17203787	0.00028	0.00026	0.00001
Russia	145899956	0.00753	0.00682	0.00013
Paraguay	7270717	0.03886	0.03211	0.00090
Yemen	30882462	0.00021	0.00009	0.00004
Senegal	17462688	0.00232	0.00224	0.00006
Sweden	10196161	0.02265	0.00049	0.00045
Cabo Verde	565346	0.04310	0.03697	0.00039
Kiribati	122470	0.00000	0.00000	0.00000
Guyana	792481	0.01706	0.01436	0.00038
Philippines	111913102	0.00943	0.00863	0.00016
Eritrea	3633796	0.00101	0.00096	0.00000
Djibouti	1010377	0.01104	0.01070	0.00014
Singapore	5925135	0.01033	0.01026	0.00001
Malaysia	33015425	0.01257	0.01160	0.00005
Fiji	906570	0.00013	0.00008	0.00000
Turkey	85484777	0.05703	0.05241	0.00048
Malawi	19953296	0.00171	0.00161	0.00006
Iraq	41750703	0.02575	0.02293	0.00037
Germany	83975691	0.00889	0.00786	0.00019
Comoros	899658	0.00427	0.00408	0.00016
+-----------+----------+---------+---------+-------+
only showing top 20 rows

AWS EC2 - Connecting

```
root@DESKTOP-NQE309U:~# ssh -i /home/brandon/jaceguaiApr11_3.pem ubuntu@ec2-3-91-215-172.compute-1.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Apr 12 16:15:16 UTC 2022

  System load:  0.04              Processes:             124
  Usage of /:   77.0% of 7.69GB   Users logged in:       0
  Memory usage: 28%               IPv4 address for eth0: 172.31.20.91
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

38 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Tue Apr 12 16:15:07 2022 from 47.151.46.200
ubuntu@ip-172-31-20-91:~$ ls
apache-hive-3.1.2-bin  apache-hive-3.1.2-bin.tar.gz  derby.log  hadoop  hadoop-3.3.2.tar.gz  metastore_db  project2  resources  spark  spark-3.2.1-bin-hadoop3.2.tgz
ubuntu@ip-172-31-20-91:~$ jps
52210 Jps
6819 SecondaryNameNode
6579 DataNode
6422 NameNode
7164 NodeManager
7005 ResourceManager
ubuntu@ip-172-31-20-91:~$ sudo su - brandon
brandon@ip-172-31-20-91:~$ jps
52236 Jps
brandon@ip-172-31-20-91:~$ logout
ubuntu@ip-172-31-20-91:~$
```



AWS EC2 - Connecting



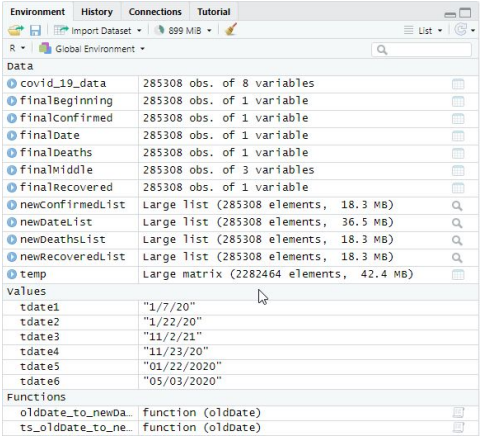
Data Cleaning/Sanitization

```
#####   Initial setup   #####
# Checks current working directory
getwd()

# Sets working directory
# Change the path to where your files are located
setwd("C:\\Users\\brany\\Documents\\Revature\\Training\\Projects\\P2\\Data_setP2")



#####   The files we are cleaning/changing    #####
# Covid-19 data file
covid_19_data = read.csv("covid_19_data.csv", header = F)

# Times Series files
ts_c19_confirmed = read.csv("time_series_covid_19_confirmed.csv", header = F)
ts_c19_confirmed_us = read.csv("time_series_covid_19_confirmed_US.csv", header = F)
ts_c19_deaths = read.csv("time_series_covid_19_deaths.csv", header = F)
ts_c19_deaths_us = read.csv("time_series_covid_19_deaths_US.csv", header = F)
ts_c19_recovered = read.csv("time_series_covid_19_recovered.csv", header = F)



#####   Functions   #####
# Convert date to format that is compatible with Hive (ONLY FOR TIMES SERIES)
# Potential formats: M/D/YY, MM/D/YY, M/DD/YY, MM/DD/YY
ts_oldDate_to_newDate <- function(oldDate) {
  tempDate <- strsplit(oldDate, split = "/")
  if (nchar(oldDate) == 6) {
    newDate <- paste("20", tempDate[[1]][[3]], "-0", tempDate[[1]][[1]], "-0", tempDate[[1]][[2]], sep = "")
    }
  else if (nchar(oldDate) == 7) {
      if (nchar(tempDate[[1]][[1]]) == 1) {
        newDate <- paste("20", tempDate[[1]][[3]], "-0", tempDate[[1]][[1]], "-", tempDate[[1]][[2]], sep = "")
      }
      else if (nchar(tempDate[[1]][[1]]) == 2){
        newDate <- paste("20", tempDate[[1]][[3]], "-", tempDate[[1]][[1]], "-0", tempDate[[1]][[2]], sep = "")
      }
    }
  else if (nchar(oldDate) == 8) {
    newDate <- paste("20", tempDate[[1]][[3]], "-", tempDate[[1]][[1]], "-", tempDate[[1]][[2]], sep = "")
  }
  else {
    return(oldDate)
  }
  return(newDate)
}

# same as above, but works for covid_19_data dates format MM/DD/YYYY
oldDate_to_newDate <- function(oldDate) {
  tempDate <- strsplit(oldDate, split = "/")
  newDate <- paste(tempDate[[1]][[3]], "-", tempDate[[1]][[1]], "-", tempDate[[1]][[2]], sep = "")
  return(newDate)
}
```

```
#####   Cleaning Data   #####
## Create csv that is a transpose of time series csv. One column is date, the rest are the countries
# Add up all Provinces/Regions so there is only 1 country.


# ts_19_confirmed
  # creates a transpose of data
transpose1 = t(ts_c19_confirmed)
  # removes columns, Province/State, Lat, Long
remCol1 = transpose1[-c(1,3,4),]
  # fixes row numbers after deleting rows
row.names(remCol1) <- NULL
  # creates a list of new dates from old dates
newDateList = lapply(remCol1[1:nrow(remCol1)], ts_oldDate_to_newDate)

  # All numbers are stored as characters, so we need to convert to numeric
df_clean = remCol1[,-1]
colnames(df_clean) <- NULL
  # now we have a transposed table with country then the rest being the data
transpose2 = data.frame(t(df_clean))

  # Fixing South Korea because it's formatted like: Korea, South
  # This causes problems later
  # Following statement gives us the row number and we already know the column
rowNum = which(apply(transpose2, 1, function(x) any(x %in% "Korea, South")))
transpose2[rowNum,1] = "Korea"

  # the columns we want to modify
i = c(2:ncol(transpose2))
  # converts the columns to numeric
transpose2[, i] = apply(transpose2[ , i], 2, function(x) as.numeric(as.character(x)))

  # convert into data.table
#install.packages("data.table", dependencies=TRUE)
library(data.table)
df_table = data.table(transpose2)
  # sum up values by country
df_sum = df_table[, lapply(.SD, sum), by=X1]

  # Set up the two data frames and combine
finalCountries = data.frame(t(df_sum))
finalDate = data.frame(t(data.frame(newDateList)))
final = cbind(finalDate, finalCountries)
row.names(final) <- NULL
colnames(final) <- NULL
  # Change Country/Region to Date
final[1,1] = "Date"

write.table(final, "C:\\Users\\brany\\Documents\\Revature\\Training\\Projects\\P2\\Test\\ts_confirmed_b
```

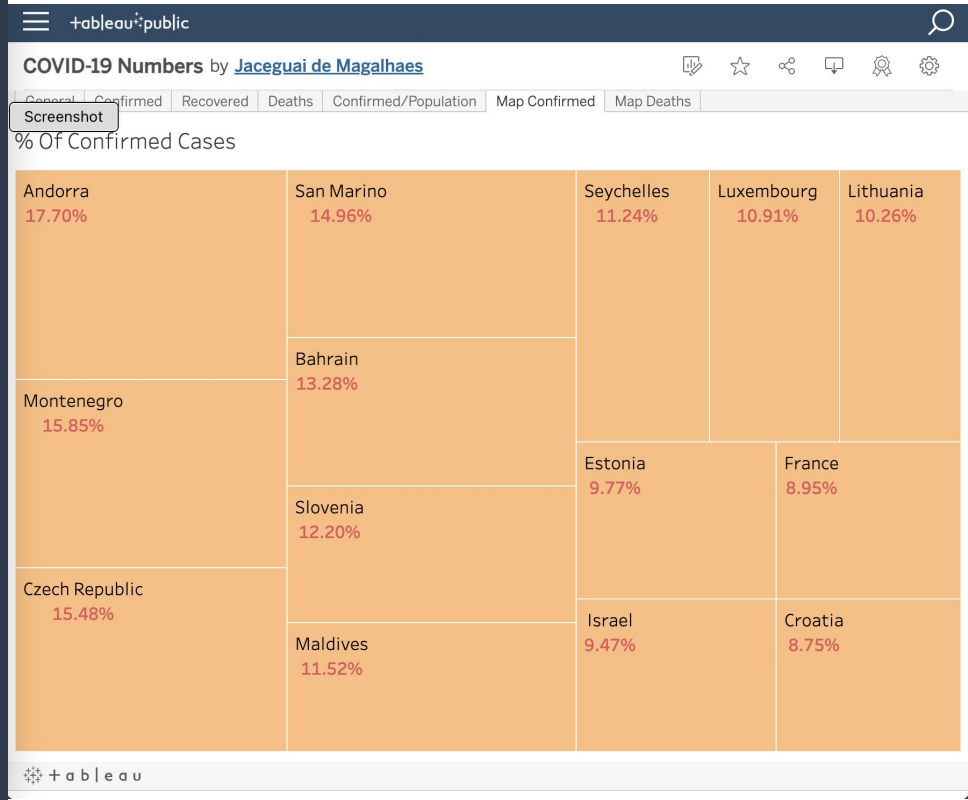
Data Cleaning/Sanitization

```
225 ## Create csv of covid_19_data, change date to Hive Format, remove Last Update (optional), remove decimals for last 3 columns
226 temp = t(t(covid_19_data))
227   # Create a list of converted dates
228 newDateList = lapply(temp[1:nrow(temp),2], oldDate_to_newDate)
229
230   # Gets rid of unnecessary decimal by converting to integer
231 newConfirmedList = lapply(temp[1:nrow(temp),6], as.integer)
232   # as.integer doesn't know what to do with "Confirmed" so it gets converted to NA. Adding it back here
233 newConfirmedList[1] = temp[[1,6]]
234   # Do the same for Deaths/Recovered
235 newDeathsList = lapply(temp[1:nrow(temp),7], as.integer)
236 newDeathsList[[1]] = temp[[1,7]]
237 newRecoveredList = lapply(temp[1:nrow(temp),8], as.integer)
238 newRecoveredList[[1]] = temp[[1,8]]
239
240   # If we store a list as a dataframe, it will appear as multiple columns instead of 1 column
241   # ObservationDate, column 2
242 finalDate = data.frame(t(data.frame(newDateList)))
243   # Confirmed, column 6
244 finalConfirmed = data.frame(t(data.frame(newConfirmedList)))
245   # Deaths, column 7
246 finalDeaths = data.frame(t(data.frame(newDeathsList)))
247   # Recovered, column 8
248 finalRecovered = data.frame(t(data.frame(newRecoveredList)))
249
250   # Store the columns that we didn't alter
251 finalBeginning = data.frame(temp[,1])
252 finalMiddle = data.frame(temp[,c(3,4,5)])
253
254   # Bind everything together
255 final = cbind(finalBeginning, finalDate, finalMiddle, finalConfirmed, finalDeaths, finalRecovered)
256 #row.names(final) <- NULL
257 #colnames(final) <- NULL  Don't need these, just to make final row/column headers look nice but we don't save them to csv
258
259   # Save to csv
260 write.table(final, "C:\\Users\\brany\\Documents\\Revature\\Training\\Projects\\P2\\Test\\covid_19_data_clean.csv", sep = ",", r
```



Q1: What % of the global population makes up the confirmed, death, and recovered cases?

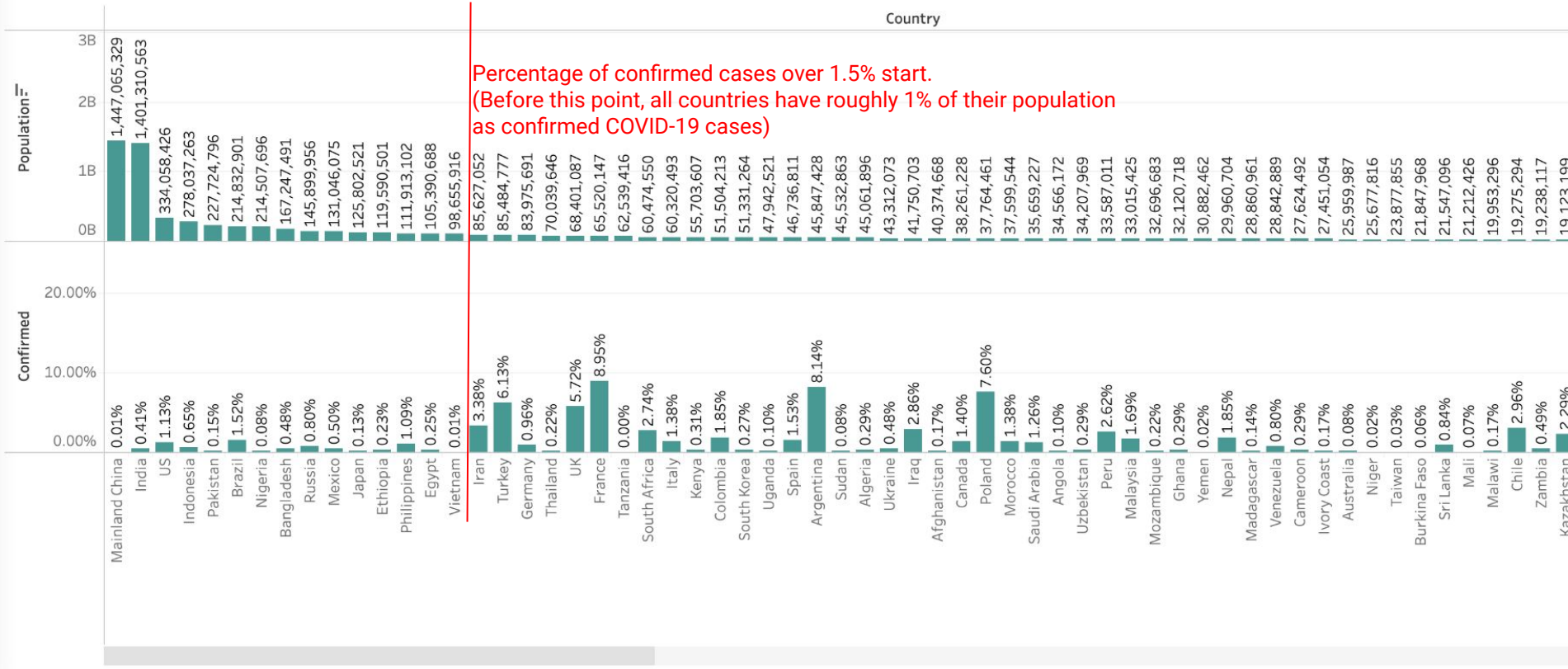
- It looks like the countries with the highest percentage of Confirmed Cases are not among the most populous countries.



COVID-19 Numbers

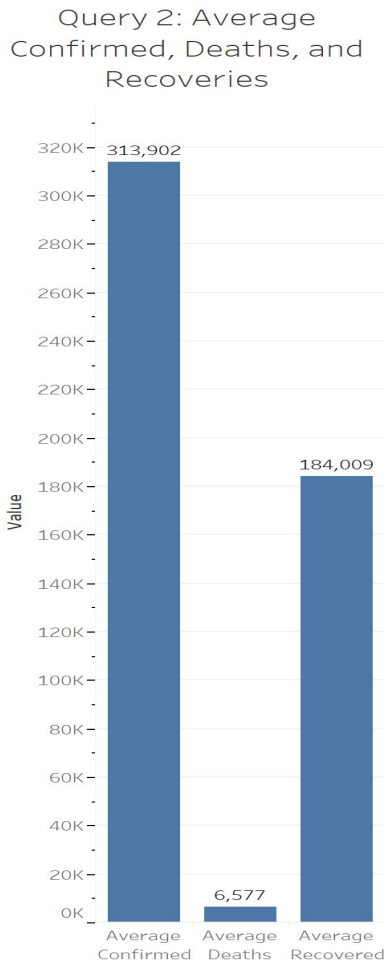
Screenshot | Confirmed | Recovered | Deaths | Confirmed/Population | Map Confirmed | Map Deaths

Confirmed/Population



Q2: What is the cumulative average of confirmed cases, death cases, and recovered cases?

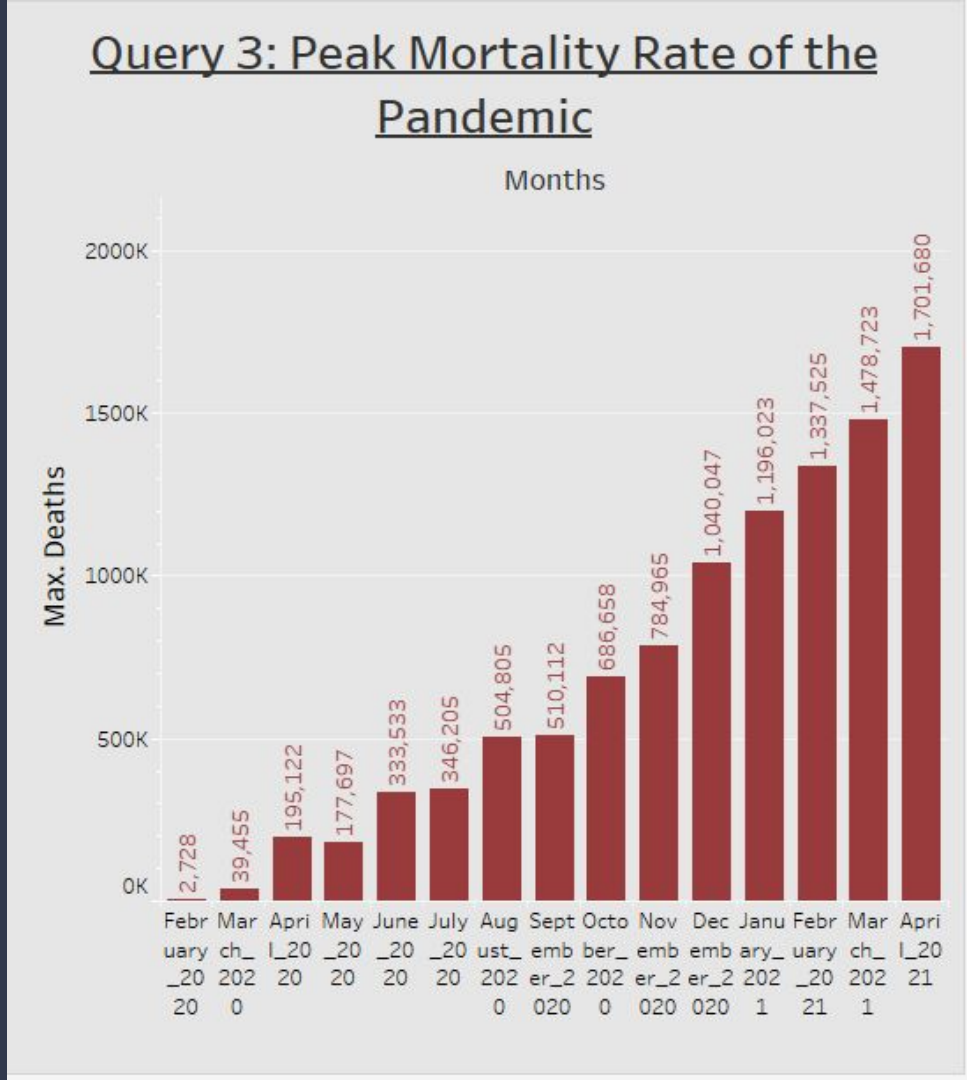
- It seems that the total cumulative average Confirmed Cases throughout the world does not reach even half a million people. While the cumulative average deaths is not even 5% of the total average confirmed cases. Also the average recovery rate is over 50% of the confirmed cases. It means that a person is far more likely to survive an infection, than he is to die from it.
- Average cases per day (World Wide)
- Date: 05/02/2021



Average Confirmed, Average Deaths and Average Recovered.

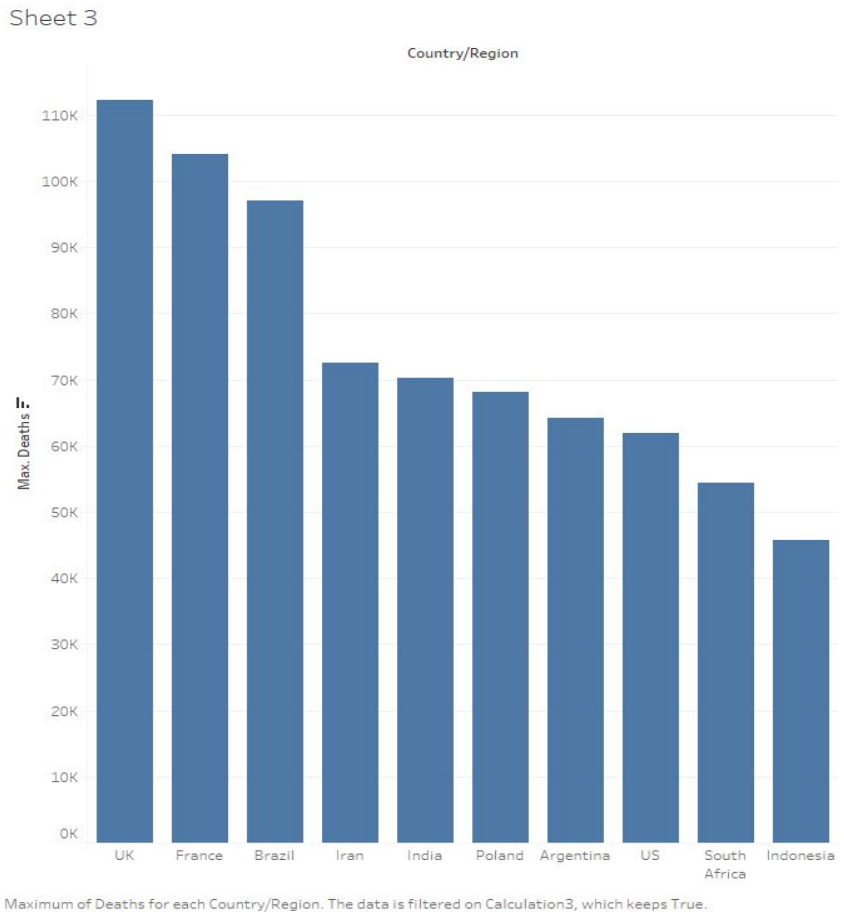
Q3: When was the peak of mortality rate during the pandemic?

- It's interesting to see that the peak of the pandemic is actually at the end of the last recorded month. We hypothesize that the peak mortality rate during the pandemic would be closer to the middle of the pandemic, but turns out that the mortality rate increases each month as it goes on



Q4: What are the top 10 countries with the greatest amount of deaths?

- Even though the source of the pandemic is from Asia. The death rate is increasingly more on other countries around the world.

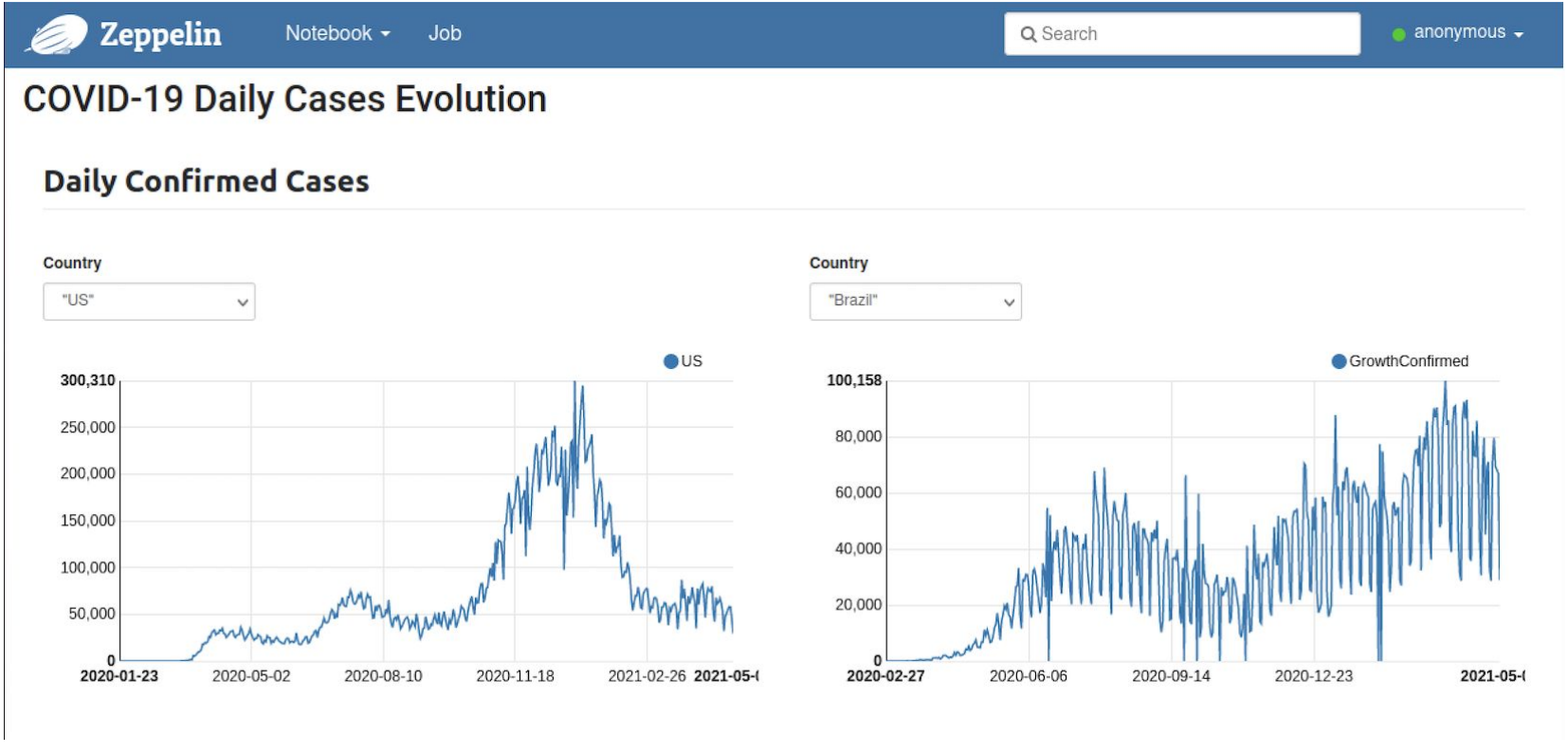


Q5: What is the general disease evolution for all countries?

For this query, we built a Zeppelin visualization where the user can select two countries side by side and compare the daily evolution of confirmed, recovered, and dead COVID-19 cases.

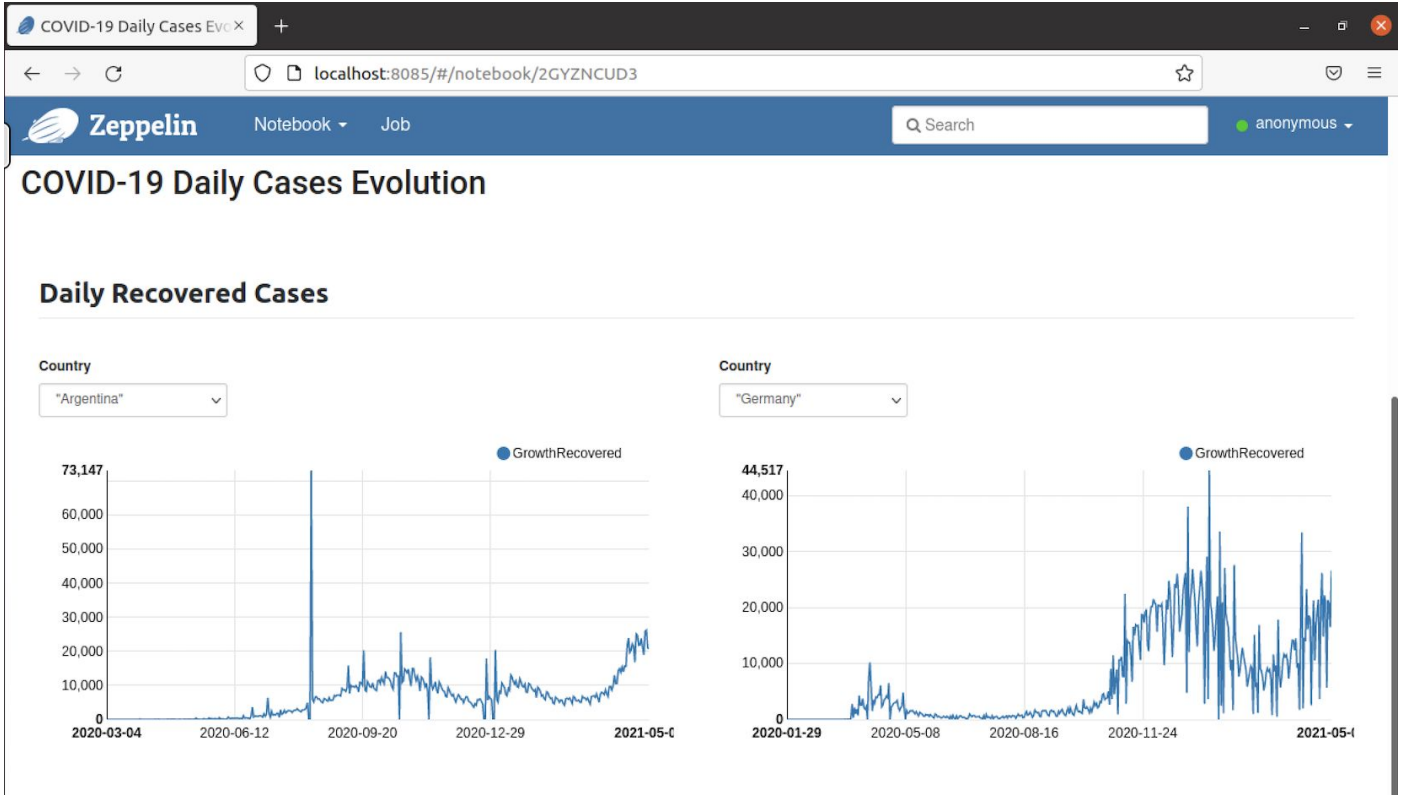
Q5: What is the general disease evolution for all countries?

- The US had a spike of confirmed cases between Nov/2020 and Feb/2021, while Brazil had a constant trend.



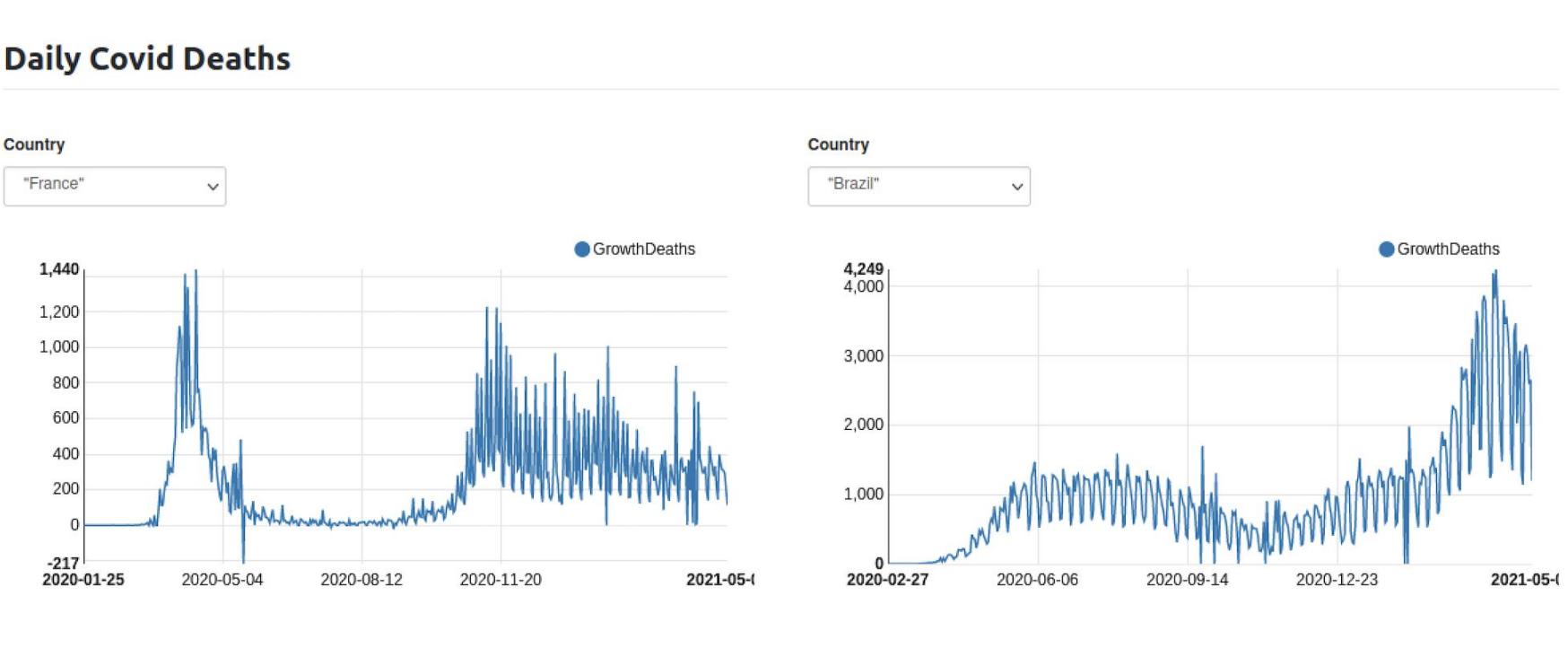
Q5: What is the general disease evolution for all countries?

- Argentina and Germany had similar trends regarding recovered cases. Argentina had a huge spike in a day in 2020. That looks like an outlier for Aug-11-2020.



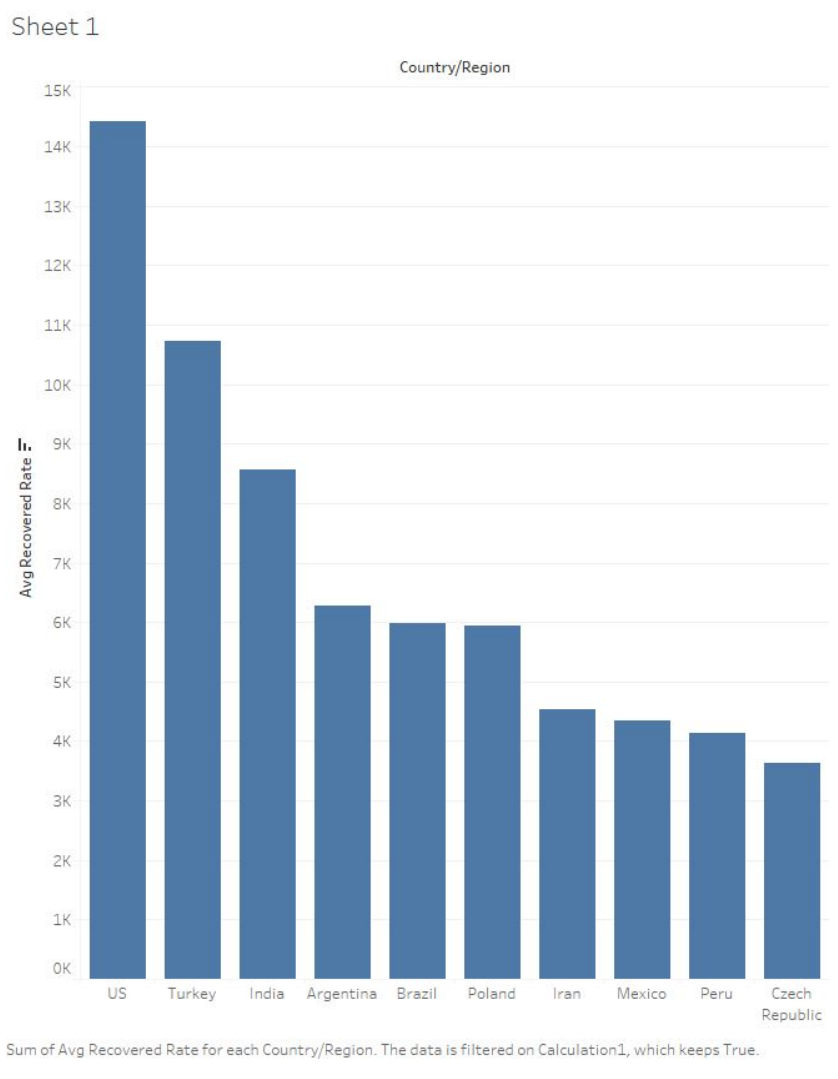
Q5: What is the general disease evolution for all countries?

- France and Brazil had different trends regarding deaths. France had an initial spike between Jan/2020 and Apr/2020. After that, deaths stabilized and increased in Oct/2020. Brazil had a steady rate, which increased dramatically in Apr/2021.



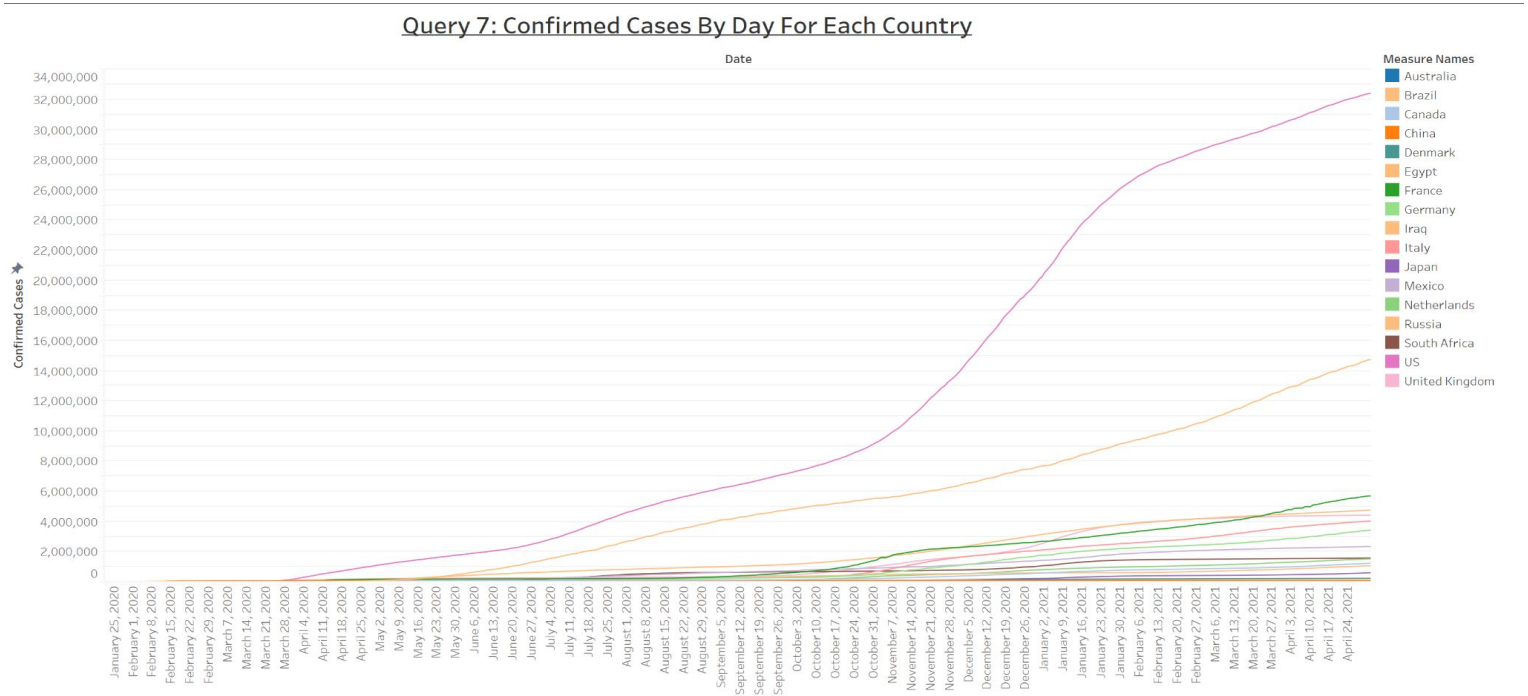
Q6: What is the average recovered rate by countries?

- Top 10 countries avg recovered rate per day
- Ordered by countries
- Pandemic recorded dates (01/22/2020-05/02/2021)



Q7: Total confirmed cases by countries

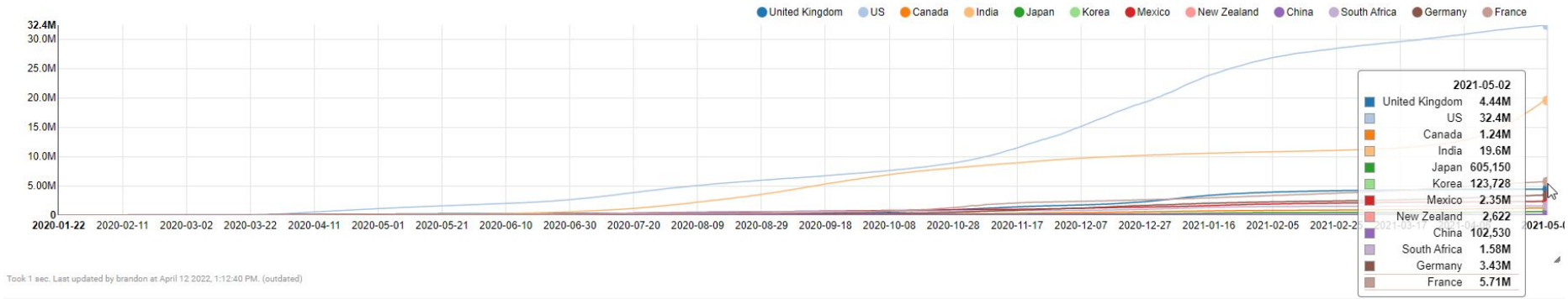
Tableau visualization:

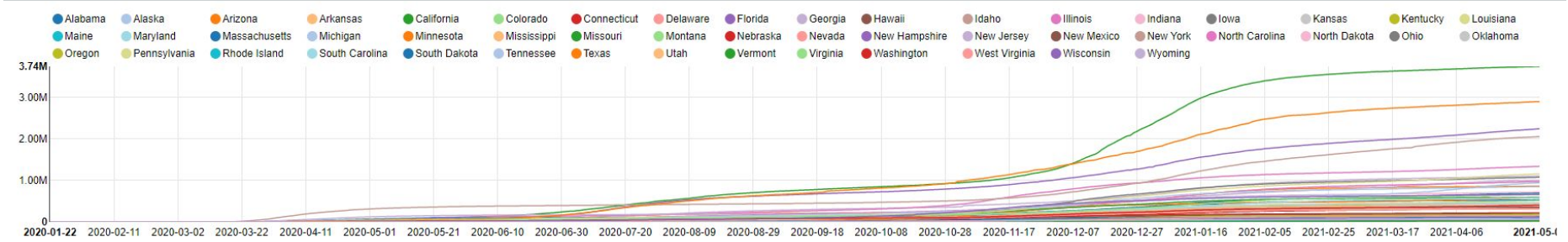


The Above Line Graph shows a visual from the results of running a query 7. Query 7 shows a comparison of the daily amount of confirmed casesfor the countries of Australia, Brazil, Canada, China, Denmark, Egypt, France, Germany, Iraq, Italy, Japan, Mexico, Netherlands, Russia, South Africa, United States, and United Kingdom for the whole duration of the pandemic.

Q7: Total confirmed cases by country and states

Zeppelin visualization:

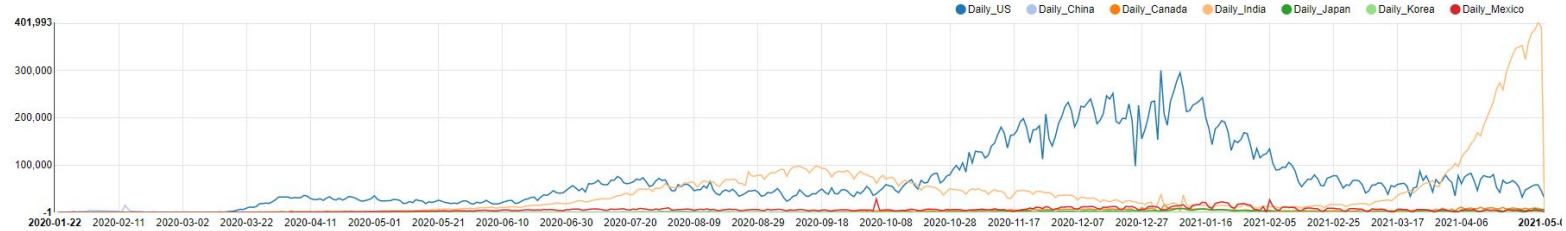




```
%spark.sql
SELECT
    Date,
    US - LAG(US)
        OVER(ORDER BY Date) AS Daily_US,
    China - LAG(China)
        OVER(ORDER BY Date) AS Daily_China,
    Canada - LAG(Canada)
        OVER(ORDER BY Date) AS Daily_Canada,
    India - LAG(India)
        OVER(ORDER BY Date) AS Daily_India,
    Japan - LAG(Japan)
        OVER(ORDER BY Date) AS Daily_Japan,
    Korea - LAG(Korea)
        OVER(ORDER BY Date) AS Daily_Korea,
    Mexico - LAG(Mexico)
        OVER(ORDER BY Date) AS Daily_Mexico
FROM tsConfirmedCountry
ORDER BY Date;
```

SPARK JOB FINISHED

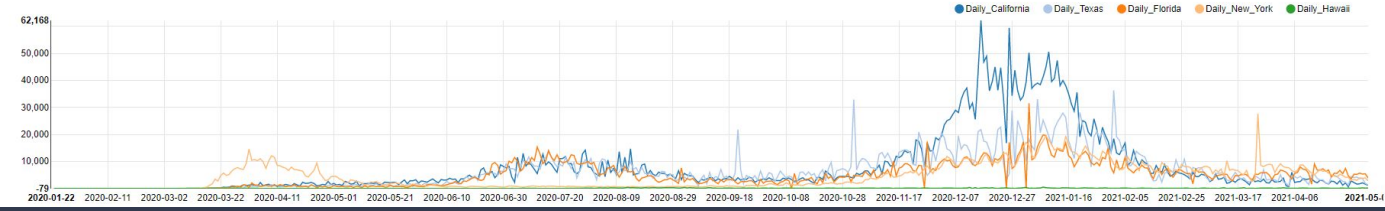
settings




```
%spark.sql
SELECT
    Date,
    California - LAG(California)
        OVER(ORDER BY Date) AS Daily_California,
    Texas - LAG(Texas)
        OVER(ORDER BY Date) AS Daily_Texas,
    Florida - LAG(Florida)
        OVER(ORDER BY Date) AS Daily_Florida,
    `New York` - LAG(`New York`)
        OVER(ORDER BY Date) AS Daily_New_York,
    Hawaii - LAG(Hawaii)
        OVER(ORDER BY Date) AS Daily_Hawaii
FROM tsConfirmedUS
ORDER BY Date;
```

SPARK JOB FINISHED

settings



Q7 Part 2: Daily Confirmed cases by Country/State

Q8: What is the total numbers of confirmed, death, and recovered cases?

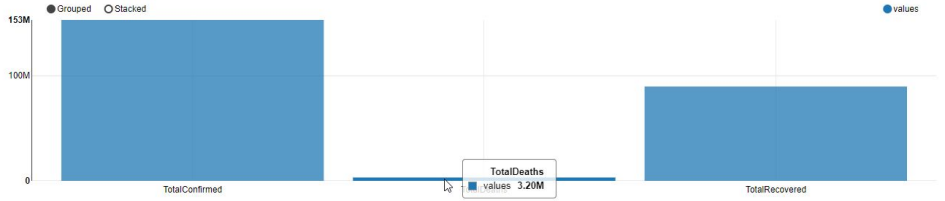
- Clean/sanitized COVID data is cumulative
- Found the sum of the last recorded date
- Showing the total worldwide
- Deaths is about 1% of all cases combined

```
%spark
import org.apache.spark.sql.Row
import org.apache.spark.sql.{DataFrame, SparkSession}
import org.apache.spark.sql.types.{ StructType, StructField, StringType };

val c19Clean = "Data/Project2/clean/covid_19_data_clean.csv"
val q8Path = "Data/Project2/results/q8_TotalCDR/part-00000-71caa67e-bffd-463f-b135-5284caeaf8e7-c000.csv"
val df_c19Clean: DataFrame = spark.read.options(Map("header"->"true", "inferSchema"->"true", "delimiter"->",")).csv(c19Clean).na.fill(0).na.fill("")
val totalCDR: DataFrame = spark.read.options(Map("header"->"true", "inferSchema"->"true", "delimiter"->",")).csv(q8Path).na.fill(0).na.fill("")

df_c19Clean.registerTempTable("Covid19Data")
```

```
warning: there was one deprecation warning (since 2.0.0); for details, enable `:setting -deprecation' or `:replay -deprecation'
import org.apache.spark.sql.Row
import org.apache.spark.sql.{DataFrame, SparkSession}
import org.apache.spark.sql.types.{StructType, StructField, StringType}
c19Clean: String = Data/Project2/clean/covid_19_data_clean.csv
q8Path: String = Data/Project2/results/q8_TotalCDR/part-00000-71caa67e-bffd-463f-b135-5284caeaf8e7-c000.csv
df_c19Clean: org.apache.spark.sql.DataFrame = [SNo: int, ObservationDate: string ... 6 more fields]
totalCDR: org.apache.spark.sql.DataFrame = [TotalConfirmed: int, TotalDeaths: int ... 1 more field]
```

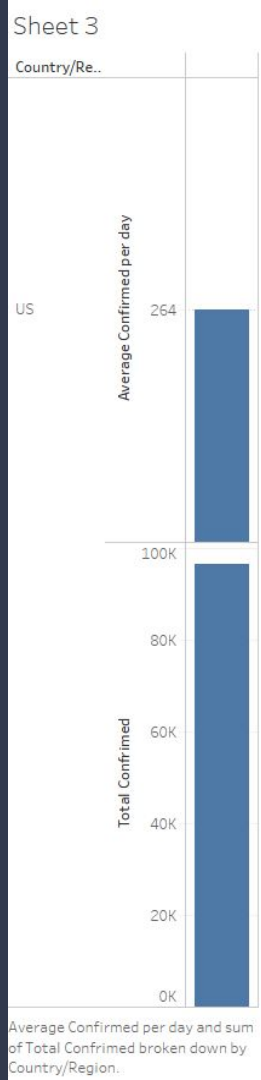
```
%spark.sql
SELECT sum(Confirmed) AS TotalConfirmed, sum(Deaths) AS TotalDeaths, sum(Recovered) AS TotalRecovered FROM Covid19Data WHERE ObservationDate = "2021
    -05-02";
```

SPARK JOB FINISHED



Q9: Spread speed for confirmed cases for a determined time and country.

- Implemented search feature
- Grabbed specific data from database
- Selected country and time period
- Show confirmed cases between time period




```
Type in the country: US
Start date. Earliest date recorded 2020-01-22 (yyyy-mm-dd): 2020-01-22
End date. Most recent date recorded 2021-05-02 (yyyy-mm-dd): 2021-01-22
+--------------+--------------+-----------------------+
|Country/Region|Total_Confrimed|Average_Confirmed_per_day|
+--------------+--------------+-----------------------+
|            US|         96565|                  264.0|
+--------------+--------------+-----------------------+
```

Total Confirmed: 96,565

Avg Confirmed per Day: 264

Country/Region: US

Date:
01/22/2020 - 01/22/2021

Q10: Spread speed for death cases for a determined time and country.

- Similar to Q9
- Implemented search feature
- Selected country and time period
- Show death cases between time period

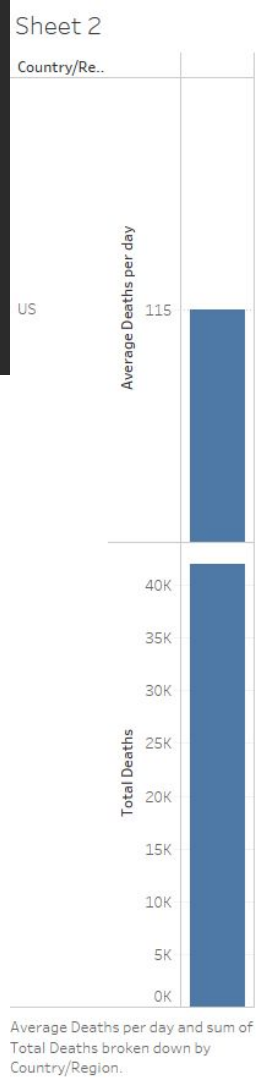
```
Confirmed spread speed for determined time and country
Type in the country: US
Start date. Earliest date recorded 2020-01-22 (yyyy-mm-dd): 2020-01-22
End date. Most recent date recorded 2021-05-02 (yyyy-mm-dd): 2021-01-22
+--------------+------------+--------------------+
|Country/Region|Total_Deaths|Average_Deaths_per_day|
+--------------+------------+--------------------+
|            US|       41974|               115.0|
+--------------+------------+--------------------+
```

Total Deaths : 41,974

Avg Deaths Per Day: 115

Country/Region: US

Date:
01/22/2020 - 01/22/2021



Questions?

GitHub

https://github.com/jaceguaidemagalhaes/Big-Data---Project-2