

COVID-19 Dataset Analysis

Harsh Vardhan Rai

Introduction

The year 2020, with no uncertainty, should be scratched from the memory of medical care experts around the globe for a long time to come. Covid (SARS-CoV-2) was first recognized in December 2019 in Wuhan, a city in China. The infection has now spread to more than 200 nations over the globe. The reactions of nations to the COVID-19 pandemic have been disparate. Different nations are endeavoring to stifle transmission of the serious intense respiratory condition by again confining organizations, enterprises, and schools. It was declared as a pandemic by the World Health Organization (WHO) on 11 March 2020. We have now surpassed over a million deaths and the human family is enduring under a practically terrible weight of misfortune.

The pandemic is significantly more than a wellbeing emergency, it's additionally a phenomenal financial emergency. Focusing on all of the nations it contacts, it can possibly make decimating social, monetary and political impacts that will leave profound and longstanding scars.

Data Description

COVID-19 Dataset

Link: <https://www.kaggle.com/imdevskp/corona-virus-report>

We are using 4 different CSV files for our analysis. These files, to automate the process without using the local machine dataset are uploaded to the repository in my GitHub and are fetched to the data frames through URL. The below figure displays the same.

```
a= 'https://raw.githubusercontent.com/harshvr15/DataVisualisation2/main/country_wise_latest.csv'
country_wise=read.csv(a,na.strings = "")

b= 'https://raw.githubusercontent.com/harshvr15/DataVisualisation2/main/covid_19_clean_complete.csv'
covid=read.csv(b,na.strings = "")

c= 'https://raw.githubusercontent.com/harshvr15/DataVisualisation2/main/full_grouped.csv'
full_grouped=read.csv(c,na.strings = "")

d= 'https://raw.githubusercontent.com/harshvr15/DataVisualisation2/main/worldometer_data.csv'
world=read.csv(d,na.strings = "")
```

The dimensions of the data frames are calculated with the help of *DIM* function. It will give us the Rows * Columns for that particular data frame.

```

> dim(country_wise)
[1] 187 15
> dim(covid)
[1] 49068 8
> dim(full_grouped)
[1] 35156 5
> dim(world)
[1] 209 14

```

Data Preprocessing

Being a raw data from Kaggle some major processing needs to be carried out before visualizing the data.

To get a general overview of the data we will use the *STR* function.

```

> str(country_wise)
'data.frame': 187 obs. of 15 variables:
 $ Country      : chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
 $ Confirmed    : int  36263 4880 27973 907 950 86 167416 37390 15303 20558 ...
 $ Deaths      : int  1269 144 1163 52 41 3 3059 711 167 713 ...
 $ Recovered    : int  25198 2745 18837 803 242 65 72575 26665 9311 18246 ...
 $ Active       : int  9796 1991 7973 52 667 18 91782 10014 5825 1599 ...
 $ New.cases    : int  106 117 616 10 18 4 4890 73 368 86 ...
 $ New.deaths   : int  10 6 8 0 1 0 120 6 6 1 ...
 $ New.recovered : int  18 63 749 0 0 5 2057 187 137 37 ...
 $ DeathPer100  : num  3.5 2.95 4.16 5.73 4.32 3.49 1.83 1.9 1.09 3.47 ...
 $ RecoverPER100 : num  69.5 56.2 67.3 88.5 25.5 ...
 $ DeathPer100Recover: num  5.04 5.25 6.17 6.48 16.94 ...
 $ ConfirmedPrevWeek : int  35526 4171 23691 884 749 76 130774 34981 12428 19743 ...
 $ WeekChange    : int  737 709 4282 23 201 10 36642 2409 2875 815 ...
 $ WeekChangePerc : num  2.07 17 18.07 2.6 26.84 ...
 $ WHO_Region    : chr  "Eastern Mediterranean" "Europe" "Africa" "Europe" ...

```

We can see from the above figure that the data frame has 13 numeric values and 2 character values. In R there is a separate way of dealing with these data types if we want to extract the summary statistics. These functions are *PSYCH* and *HMISC*. While *HMISC* deals with the categorical data type *PSYCH* deals with the numerical data statistics. As both of the packages uses *DESCRIBE* as their function, we need to make sure that we have unloaded the one which we aren't using to avoid any errors.

```
detach("package:psych", unload = TRUE)
```

```
describe(country_wise$Country)
describe(country_wise$WHO_Region)
```

As described in the above figure as we are initially looking into the categorical data, we must unload PSYCH first and then use the describe function. For the numerical columns statistics, we will unload PSYCH and load HMISC like the figure attached below.

```
detach("package:Hmisc", unload = TRUE)
library(psych)

describe(country_wise$Confirmed)
describe(country_wise$Deaths)
describe(country_wise$Recovered)
describe(country_wise$Active)
describe(country_wise$New.cases)
describe(country_wise$New.deaths)
describe(country_wise$New.recovered)
describe(country_wise$DeathPer100)
describe(country_wise$RecoverPer100)
describe(country_wise$DeathPer100Recover)
describe(country_wise$ConfirmedPrevWeek)
describe(country_wise$WeekChange)
describe(country_wise$WeekChangePerc)
```

The above code for the *DESCRIBE* function results in different output for both types of variables. If we look at the numerical data type there is a lot to explore apart from the basic statistics like skewness, kurtosis, standard error, MAD (median absolute deviation) and much more.

```
> describe(country_wise$Confirmed)
vars  n    mean      sd median trimmed   mad min    max   range skew kurtosis    se
X1    1 187 88130.94 383318.7  5059 21699.69 7372.97  10 4290259 4290249 8.59   82.85 28031.04
> describe(country_wise$Deaths)
vars  n    mean      sd median trimmed   mad min    max   range skew kurtosis    se
X1    1 187 3497.52 14100    108  626.69 157.16   0 148011 148011 7.35   63.96 1031.09
> describe(country_wise$Recovered)
vars  n    mean      sd median trimmed   mad min    max   range skew kurtosis    se
X1    1 187 50631.48 190188.2  2815 11514.81 4134.97   0 1846641 1846641 6.87   53.48 13907.94
> describe(country_wise$Active)
vars  n    mean      sd median trimmed   mad min    max   range skew kurtosis    se
X1    1 187 34001.94 213326.2  1600  6104.5 2349.92   0 2816444 2816444 11.99  152.03 15599.95
> .
```

While the categorical output shows us the total observation, number of distinct values in that observation, the frequency and proportion of that data. The output looks like:

```
> describe(country_wise$WHO_Region)
country_wise$WHO_Region
  n missing distinct
187      0         6
```

	lowest : Africa	Americas	Eastern Mediterranean	Europe	South-East Asia
highest: Americas		Eastern Mediterranean	Europe	South-East Asia	Western Pacific

Value	Africa	Americas	Eastern Mediterranean	Europe
Frequency	48	35	22	56
Proportion	0.257	0.187	0.118	0.299

Value	South-East Asia	Western Pacific
Frequency	10	16
Proportion	0.053	0.086

Similarly, the overview was generated for each of our data frames.

Steps

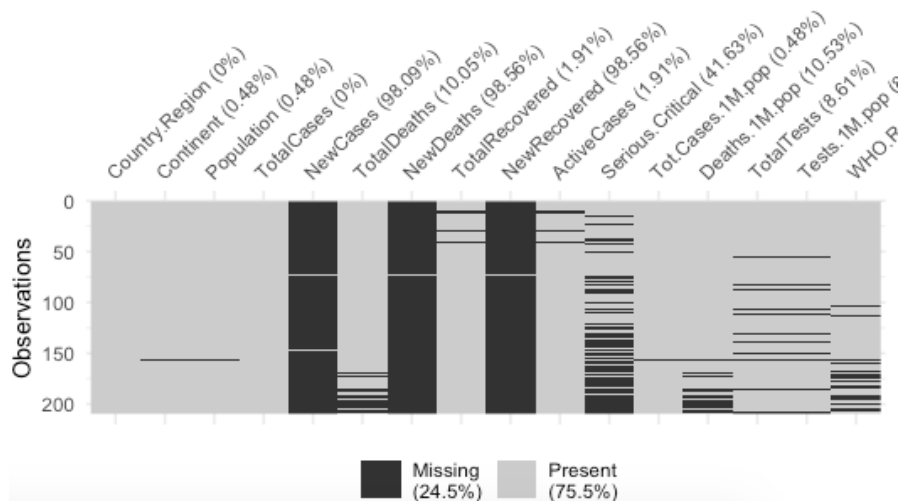
- Computing the missing values: As we are having 4 data frames, we need to calculate the NA/unknown values across all the columns.

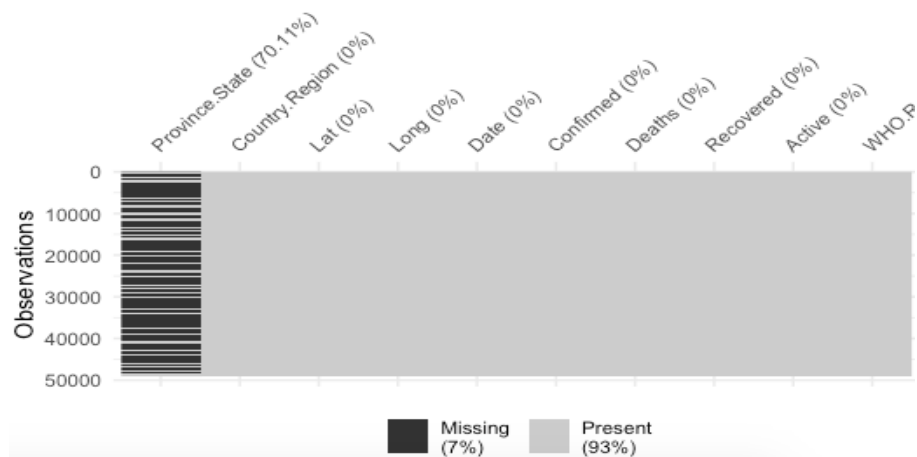
```
> apply(is.na(world),2,sum)
```

Country.Region	Continent	Population	TotalCases	NewCases	TotalDeaths
0	1	1	0	205	21
NewDeaths	TotalRecovered	NewRecovered	ActiveCases	Serious.Critical	Tot.Cases.1M.pop
206	4	206	4	87	1
Deaths.1M.pop	TotalTests	Tests.1M.pop	WHO.Region		
22	18	18	25		

In the above figure we are calculating the sum of all the NA's across the columns in the world data frame. Similarly, we will check it for other data frames. Looking at the output in RStudio we got to know that only two data frames, namely world & covid have missing values.

- To get a proper overview on how to deal with missing values, we need to plot the missing variables over a graph. This was done with *VIS_MISS* function. This function will calculate how much percentage of total value for each column is missing. The two data frames world and covid have these many missing values respectively as attached below.





- Replacing some of the missing values with '-' as no operations could be performed to fill in these values.

```
#Filling the NA with "-" as nothing can be done for these missing values
world$Continent[is.na(world$Continent)]="-"
world$WHO_Region[is.na(world$WHO_Region)]="-"
```

- Checking out for a population of a country, *SQLDF* was used to extract the country having this NA population, but as it turns out to be a cruise no proper details were available. So, replacing it with 0.

```
#Checking for the country with NA population
sqldf("select Country,Population from world where Population is NULL ")
#As it's data can't be extracted filling it with 0
world$Population[is.na(world$Population)]=0
```

- As we are having four different data frames to be analyzed, many of the data frames have same columns, so to make it more efficient, I would just delete duplicate entries for column.

```
#Removing a column "Province" as we already have Country as an identifier
covid = covid[-c(1)]

#Removing these columns as they were present in another dataframe
world = world[-c(5,7,9,11)]
full_grouped = full_grouped[-c(3,4,5,6,10)]
covid = covid[-c(9)]
```

- For two of our data frames namely covid and full_grouped we have two Date datatypes, but initially it was taking it as a char so converting the datatypes to date.

```
#Converting datatypes - char to Date
covid$Date = as.Date(covid$Date)
full_grouped$Date = as.Date(full_grouped$Date)
```

- For better readability we need to change some of the column's name of all the data frames.

```
#Changing name of the columns for better readability
names(world)[1]="Country"
names(world)[11]="SeriousCondition"
names(world)[12]="CasesPerMillion"
names(world)[13]="DeathsPerMillion"
names(world)[15]="TestsPerMillion"
names(world)[16]="WHO_Region"

names(country_wise)[1]="Country"
names(country_wise)[9]="DeathPer100"
names(country_wise)[10]="RecoverPER100"
names(country_wise)[11]="DeathPer100Recover"
names(country_wise)[12]="ConfirmedPrevWeek"
names(country_wise)[13]="WeekChange"
names(country_wise)[14]="WeekChangePerc"
names(country_wise)[15]="WHO_Region"

names(covid)[1]="State"
names(covid)[2]="Country"
names(covid)[10]="WHO_Region"

names(full_grouped)[2]="Country"
names(full_grouped)[10]="WHO_Region"
```

Problem & Audience

Public information is always a crucial factor for acquiring a more profound insight and revealing its basic insider facts, which prompts new AI models that can be utilized to make significant expectations about what's to come in future.

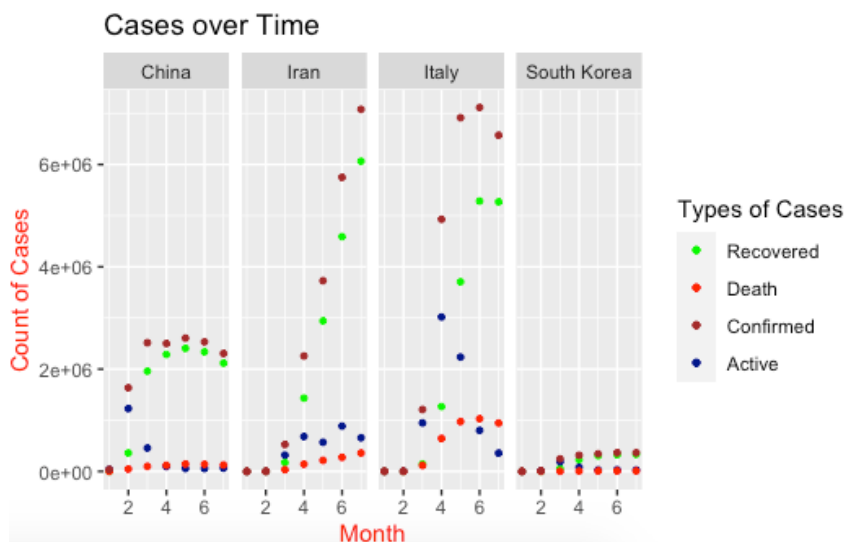
The biggest **problem** for this pandemic was public datasets were not made accessible in the start of the flare-up and even if they were made public, they didn't contain the essential details. Similarly, the kind of conclusion that was accomplished for patients, sorts of testing that every nation is as of now doing, just as the underlying manifestations are not accessible as information.

*The **motivation** behind creating this information accessible was to give **scientists, general wellbeing specialists, and the overall population** an ease to understand the current scenario of this pandemic.*

I would like to explain the importance of this COVID dataset with some real-life examples. Some of the countries like South Korea and China can be viewed as a leader in delivering information on its COVID-19 circumstance. According to some research done by me:

In mid-March, South Korea had the fourth most elevated graph apart from China, Italy and Iran. But in any case, their treatment of the emergency has been generally praised as a benchmark as far as both powerful reaction and its open based way to deal with pandemic

by utilizing front line innovation. The huge number of cases in South Korea can be described to the nation's boundless trying to include in excess of 200,000 individuals, and with an ability to test up to 20,000 every day. This was made conceivable by conveying numerous advancements like indicative applications, creative testing packs, and working from home arrangements.



Also, by testing numerous cases, and not just individuals, South Korea has distinguished more asymptomatic and positive instances of Covid than Italy, especially among youngsters. This information is significant on the grounds that it shows that more youthful individuals, who may not be tried for COVID-19 since they are asymptomatic, may be the ones that are spreading the infection. In any case, by applying this general wellbeing measure, asymptomatic individuals with the infection can segregate regardless of whether they don't feel debilitated and forestall spreading the infection.

In China, they have been utilizing enormous information to stop the spread of the infection. The main supplier of web, Tencent, added esteem benefits and has been revealing a QR code framework on the informal organization WeChat to follow potential COVID-19 transporters on open transportation. Travelers entering a transport, tram or taxi can present their data through Tencent's "ride enrollment code" and the framework will synchronize their genuine names with the vehicle's tag, loading up time and other data. At the point when a traveler is found to have been tainted, different travelers who may have been uncovered will get a message cautioning them.

These public resources are colossally useful for general wellbeing experts and data specialists reacting to the plague. They make information from various sources simple to utilize, which can empower snappy improvement of representations of the cases and their future effect.

Data Exploration

- Using *CROSS_TABLE* function to generate a proportion table. To get the impact of the COVID via the WHO region, we are using this column and function. As result we got to know that **Europe** has the maximum number of cases(proportion).
- As a part of exploration just to divide countries into **Zones** for a better overview, I have created four zones based on the number of confirmed cases namely **Red, Orange, Yellow** and **Green**. The condition through which I divided was:

Confirmed > 1Million (10Lakhs)	RED
Confirmed > 1 Lakh	ORANGE
Confirmed > 10 Thousand	YELLOW
Confirmed > 99	GREEN

```
> CrossTable(country_wise$WHO_Region)
```

```
Cell Contents
|-----|
|          N |
| N / Table Total |
|-----|
```

Total Observations in Table: 187

Africa	Americas	Eastern Mediterranean	Europe	South-East Asia
48	35	22	56	10
0.257	0.187	0.118	0.299	0.053

Western Pacific
16
0.086

After *FILTER* and *MUTATE* this data to the original data frame, using *CrossTable* to get a proper proportion of the new data. **USA** has the greatest number of Confirmed cases.

	Country	Confirmed	Zones
1	US	4290259	Red
2	Brazil	2442375	Red
3	India	1480073	Red
4	Russia	816680	Orange
5	South Africa	452529	Orange
6	Mexico	395489	Orange
7	Peru	389717	Orange
8	Chile	347923	Orange
9	United Kingdom	301708	Orange
10	Iran	293606	Orange

Cell Contents

```

|-----|
|              N |
|      N / Table Total |
|-----|

```

Total Observations in Table: 187

Green	Orange	Red	Yellow
107	21	3	56
0.572	0.112	0.016	0.299

- Further to explore the death trails dividing countries into **DZones** for a better overview, I have created five DZones based on the number of death cases namely **Red, Orange, Yellow and Green**. The condition through which I divided was:

Deaths > 50 Thousand	RED
Deaths > 10 Thousand	ORANGE
Deaths > 1 Thousand	YELLOW
Deaths > 500	GREEN
Deaths < 500	LOW

After *FILTER* and *MUTATE* this data to the original data frame, using *CrossTable* to get a proper proportion of the new data. **USA** has the greatest number of Death cases.

	Country	Deaths	DZones
1	US	148011	Red
2	Brazil	87618	Red
3	United Kingdom	45844	Orange
4	Mexico	44022	Orange
5	Italy	35112	Orange
6	India	33408	Orange
7	France	30212	Orange
8	Spain	28432	Orange
9	Peru	18418	Orange
10	Iran	15912	Orange

> `CrossTable(death_cases$DZones)`

Cell Contents

```

|-----|
|              N |
|      N / Table Total |
|-----|

```

Total Observations in Table: 187

Green	Low	Orange	Red	Yellow
132	11	9	2	33
0.706	0.059	0.048	0.011	0.176

- Further to investigate the hotspot areas dividing countries into **Hotspot** for a better overview, I have created five DZones based on the number of death cases namely **Red, Orange, Yellow and Green**. The condition through which I divided was:

Active > 1 Lakh	RED
Active > 10 Thousand	ORANGE
Active > 1 Thousand	YELLOW
Active < 1 Thousand	GREEN

After *FILTER* and *MUTATE* this data to the original data frame, using CrossTable to get a proper proportion of the new data. **USA** is the greatest Hotspot.

	Country	Active
1	US	2816444
2	Brazil	508116
3	India	495499
4	United Kingdom	254427
5	Russia	201097
6	South Africa	170537
7	Colombia	117163
8	France	108928
9	Canada	107514
10	Peru	98752

```
> CrossTable(active_cases$Hotspot)
```

```
Cell Contents
|-----|
|              N |
|      N / Table Total |
|-----|
```

Total Observations in Table: 187

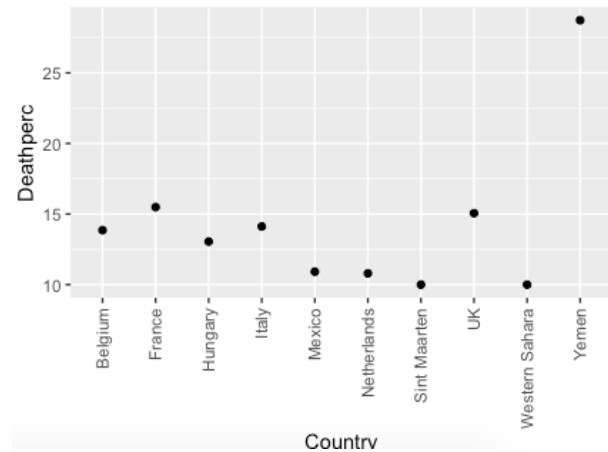
	Green	Orange	Red	Yellow
	83	35	9	60
	0.444	0.187	0.048	0.321

- *Calculating and Mutating* “**Death Percentage**” & “**Recovery Percentage**” column in the original dataset to get a proportional overview with respect to the total number of cases.

```
#Calculating Death% and Recovery% and mutate
world = world %>% mutate(Deathperc = round((TotalDeaths/TotalCases)*100,digits =2))
world = world %>% mutate(Recoverperc = round((TotalRecovered/TotalCases)*100,digits =2))
```

- Further to investigate the above calculated variables, I wanted to get the countries with **Top10** Death Percentage. To get a reasonable analysis, I even extracted their Population and Death per million along with the Death Percentage.

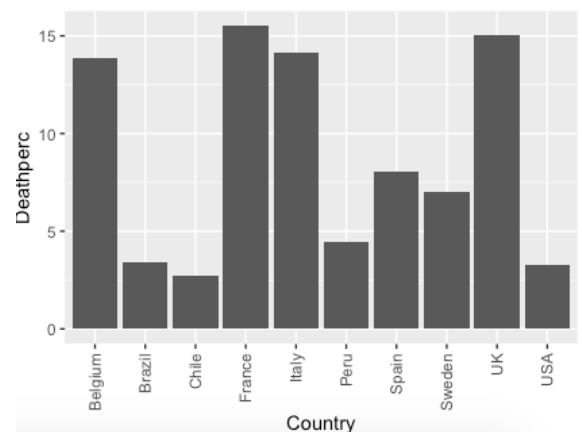
	Country	Deathperc	Population	DeathsPerMillion
1	Yemen	28.73	29886897	17
2	France	15.49	65288306	464
3	UK	15.06	67922029	683
4	Italy	14.12	60452568	582
5	Belgium	13.86	11594739	850
6	Hungary	13.05	9657785	62
7	Mexico	10.92	129066160	391
8	Netherlands	10.80	17138756	359
9	Sint Maarten	10.00	42924	373
10	Western Sahara	10.00	598682	2



To conclude the above statistics, we can see that the something is not correct about this subset being extracted. As in some of the cases we can see that Death per million is very less due to their overall population.

So, to get the proper statistics we need certain conditions where we can keep threshold for population and death per million.

	Country	Deathperc	Population	DeathsPerMillion
1	France	15.49	65288306	464
2	UK	15.06	67922029	683
3	Italy	14.12	60452568	582
4	Belgium	13.86	11594739	850
5	Spain	8.04	46756648	610
6	Sweden	7.03	10105596	571
7	Peru	4.48	33016319	619
8	Brazil	3.38	212710692	464
9	USA	3.24	331198130	492
10	Chile	2.70	19132514	517



Here in the above output I have set the condition as Population should be greater than 1M and DeathPerMillion should be greater than 400. Now there are the countries with a set population and a symmetry with no outliers. **France** has the highest death percentage as seen in the above bars graph.

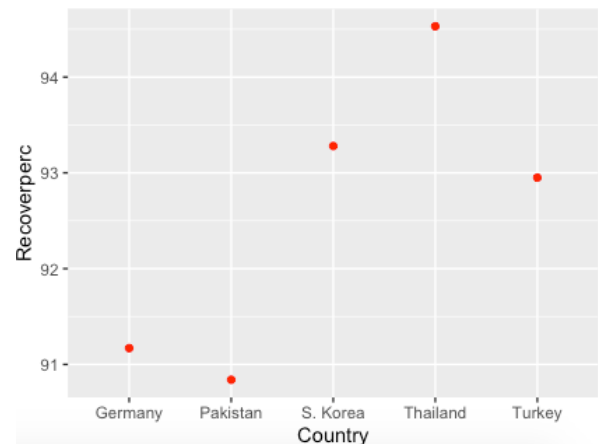
- In the same data frame, we also mutated Recover Percentage. But as in the above scenario if we are trying to extract countries with maximum recovery percentage, we end up getting a slight skewed output because of the population. It will consist of mostly small countries where cases were less, or the overall population was less ending up with a 100% recovery rate. This type of output will look like:

	Country	Recoverperc	Population
1	Macao	100.00	650193
2	New Caledonia	100.00	285769
3	Dominica	100.00	72004
4	Greenland	100.00	56780
5	Falkland Islands	100.00	3489
6	Vatican City	100.00	801
7	Cayman Islands	99.51	65798
8	Seychelles	98.41	98408
9	Brunei	97.87	437893
10	New Zealand	97.13	5002100

Again, setting some threshold for our subset, where Population is greater than 50M and Recover percentage is greater than 90%. While we were searching for top 10 countries, we end up only with 5 countries that satisfy these conditions.

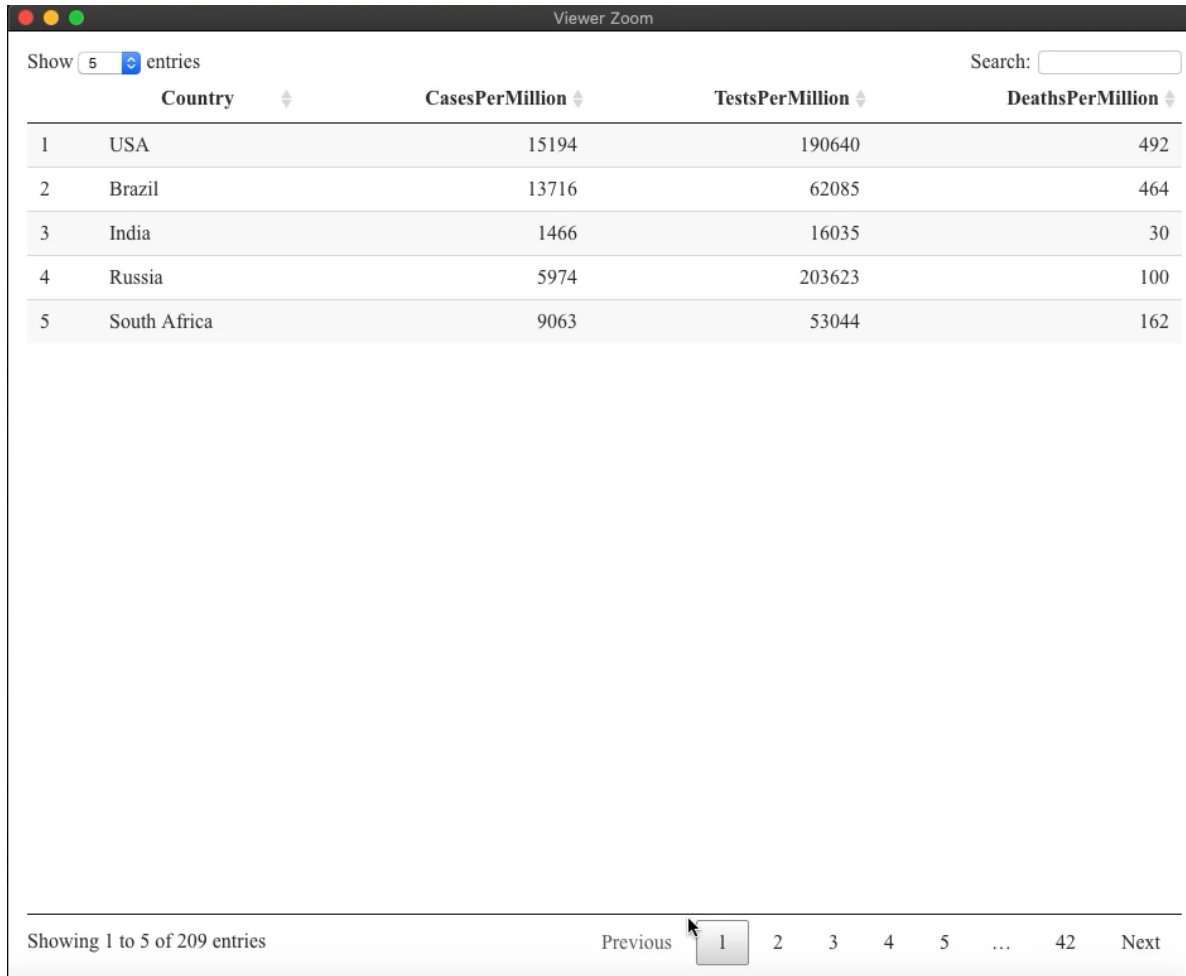
Thailand has the most elevated rate **94.53%**.

	Country	Recoverperc	Population
1	Thailand	94.53	69817894
2	S. Korea	93.28	51273732
3	Turkey	92.95	84428331
4	Germany	91.17	83811260
5	Pakistan	90.84	221295851



- To check the Cases per million(M), Tests per M, Deaths per M for top 10, last 10, or either searching data for a particular country, every time we need to come up with a query where we need to select, filter or search. But it can be modified with the *DATATABLE* function and all the above questions or queries can be analyzed within certain clicks even without modifying the query. (**Attaching video file for same**)

It's a video file in reference with the above exploration, right click and play(or just double click) this for further insights.



Viewer Zoom

Show 5 entries Search:

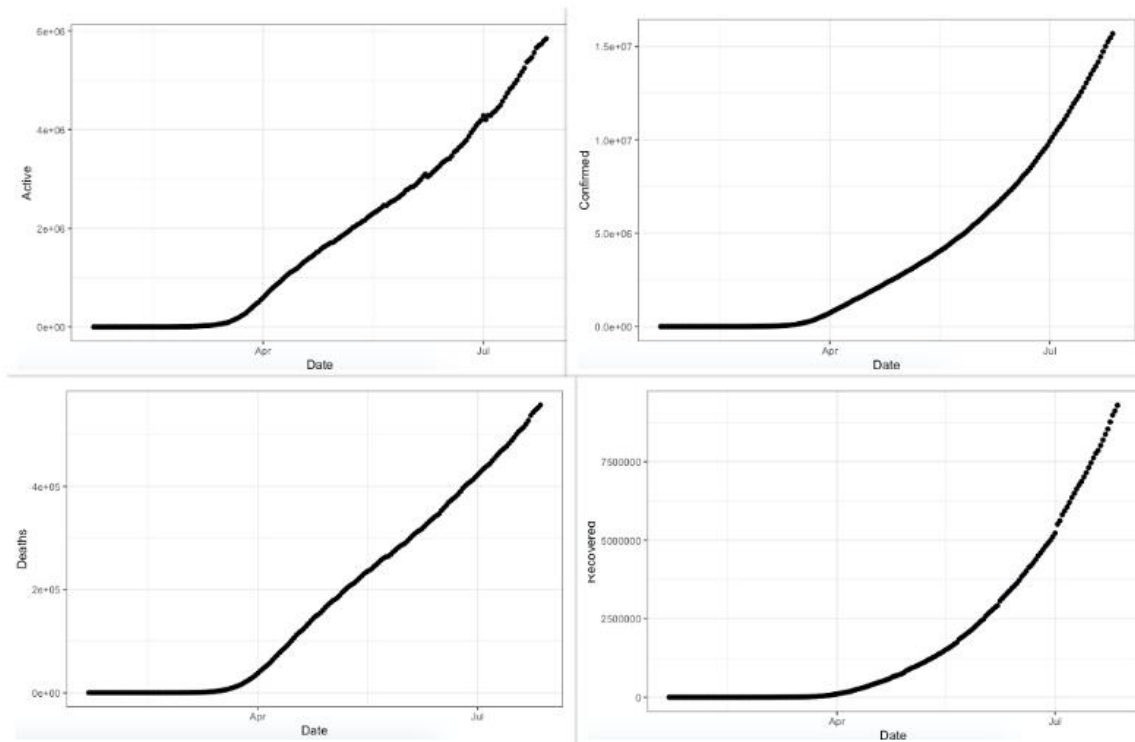
	Country	CasesPerMillion	TestsPerMillion	DeathsPerMillion
1	USA	15194	190640	492
2	Brazil	13716	62085	464
3	India	1466	16035	30
4	Russia	5974	203623	100
5	South Africa	9063	53044	162

Showing 1 to 5 of 209 entries Previous 1 2 3 4 5 ... 42 Next

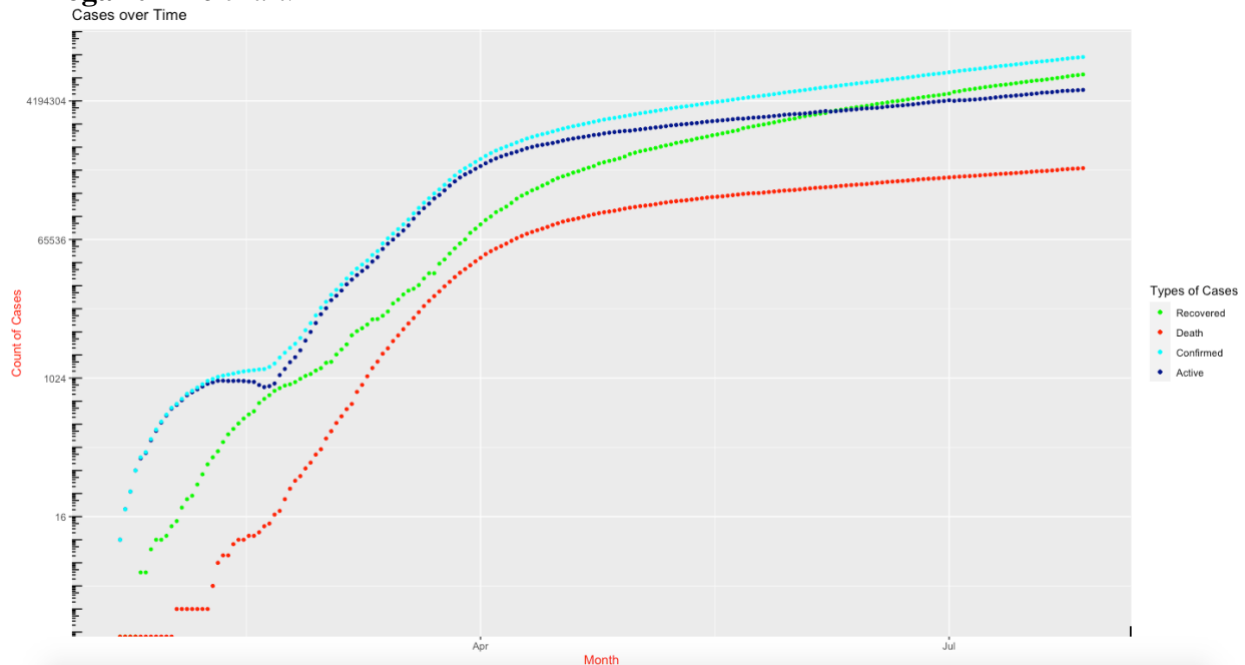
Insights & Previous Iterations

VISUALISATION 1:

I want to visualize the overall trend over the globe for all the Confirmed, Deaths, Recovered, Active cases. Initially (or you can say the first iteration), I have plotted a **Linear** graph for all the various cases during the given timeline based on our dataset. The figure attached below displays the same.



Problem based on initial iteration: Be that as it may, these pandemics don't spread in an even, linear design. On a linear scale, the pace keeps going up constantly – the line can turn out to be practically vertical and seem to go on for eternity. That can make the impression estimates like no factors like social distancing, lockdown is working in a positive way for the overall scenario. So, to overcome this, we will go with the **Logarithmic chart**.



Why this chart? On a logarithmic scale, numbers on the Y-pivot don't climb in equivalent augmentations however rather every stretch increments by a set factor depending upon us. Everything relies upon what is considered to be the best method of deciphering the information being referred to.

Functionalities added:

- *SUMMARISE* by sum of all the Confirmed, Deaths, Active & Recovered cases.

```
summarise(Confirmed=sum(Confirmed),Deaths=sum(Deaths),Active=sum(Active),Recovered=sum(Recovered))
```

- Adding Log to our base chart, the *SCALE_Y_CONTINUOUS* will allow us to set the Log scales for the Y-axis, here for our data we are taking LOG2 and meanwhile logticks are added for a better diagram.

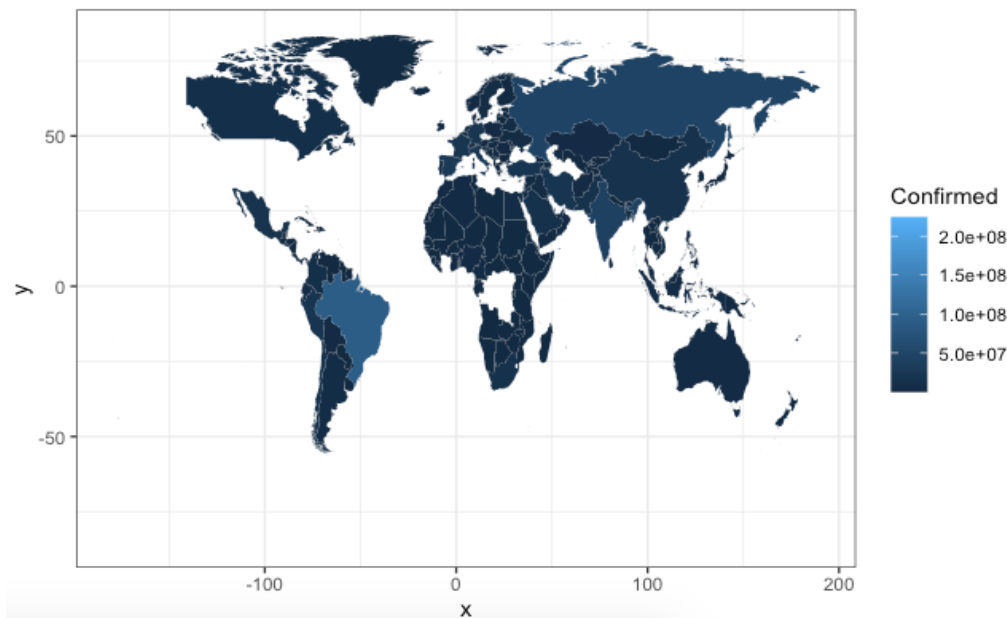
```
b+scale_y_continuous(trans = 'log2')+annotation_logticks()+
```

Conclusion: On a logarithmic chart of COVID-19, despite the fact that the general numbers are as yet expanding, you can see where the pace of virus begins to level off. By then, the logarithmic scale unveils it, when wellbeing measures like lockdown & social distancing are beginning to have the ideal impact. Also, the Death is quite lower than the Confirmed cases, this can be because of restricted testing and issues in the attribution of the reason for death. The recovery overall is great but the spike in all the types of cases is a big concern.

VISUALISATION 2:

Here I'm plotting the map based on the confirmed cases, based on the Latitude & Longitude of the region.

Problem based on initial iteration: In the initial stage when the map was plotted, the regions weren't clear, the legend scale was quite inaccurate (showing the exponential notation), the color palette was not good to map a position and status of that area and the most important thing, USA & DR of Congo was not even there on the map whereas in the query to plot a map, USA stands at the top position for most number of confirmed cases.



Functionalities added:

- *TRANS* in the scale functionality is used for reversing the scale of the legend. Initially in the above graph we can see the greatest number of confirmed cases(exponential) is represented by a lighter stroke of color while in standard format the color goes darker from low value to high value with respect to a variable. So, using `trans = "reverse"` will give the expected output.
- *LABELS* in the same functionality will describe the legend more appropriately. Initially everything will be in exponential form due to large numbers of cases but using this will give us an exact figure in numerical form.
- *N.BREAKS* will give us a detailed scale of the number of cases. When we plot the initial graph, the legend will scale down the colors in only 3-4 divisions but when we are dealing with a huge amount of dataset, so we need at least 9-10 divisions of color to get a detailed insight
- *VIRIDIS* package gives us the different color scales within package. We can choose our own scaling options using the *OPTION* parameter.

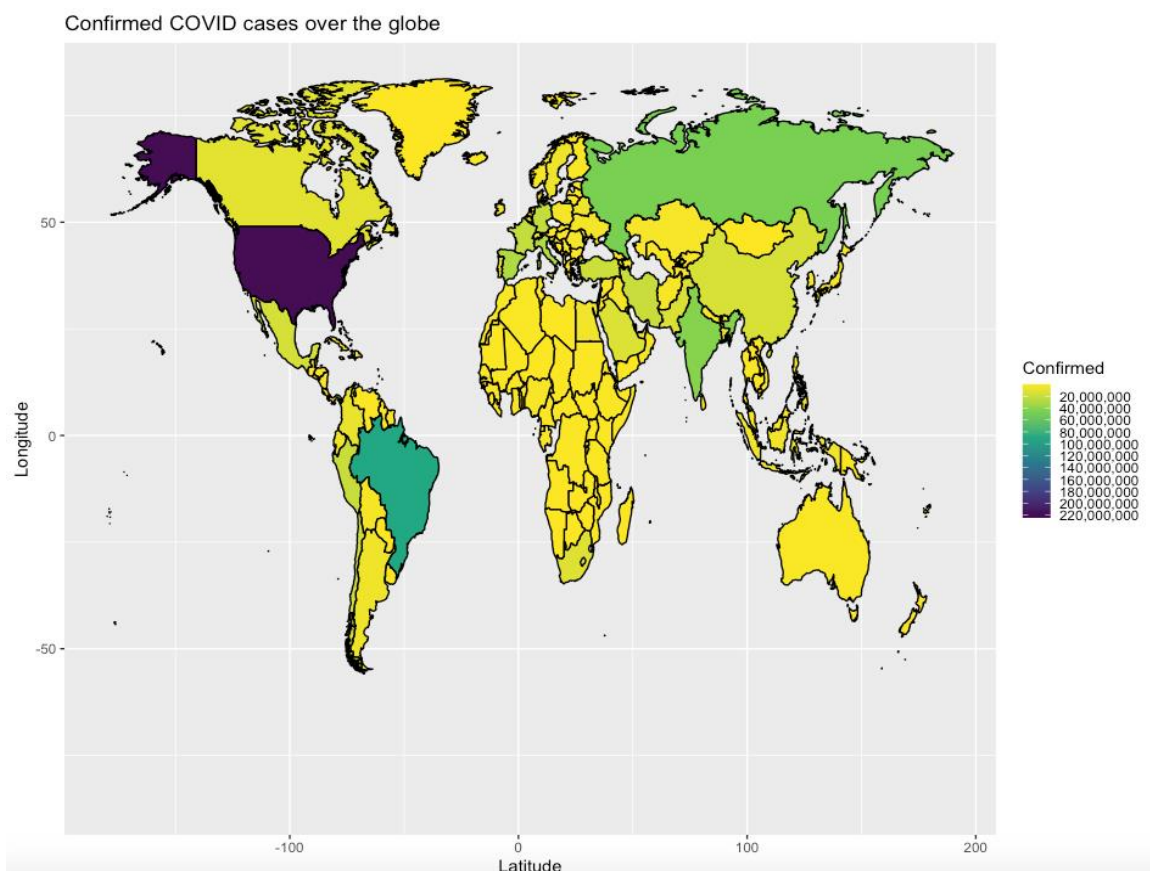
```
scale_fill_viridis_c(option = 'viridis',trans = 'reverse',labels = comma,n.breaks=11,guide = "colourbar")+
```

- While looking at the first iteration of the graph I found that some of the countries are missing, when I went back to the data frame, I found that USA was named in

```
a$Country= gsub("US", "USA",a$Country)
a = a %>% mutate(Country = case_when(str_detect(Country,"Congo")~"Democratic Republic of the Congo",TRUE ~ Country))
```

the data frame as US and Democratic Republic of the Congo was written as Congo (Brazzaville) so when we renamed it, they were also added to the graph.

Conclusion: An individual can be confirmed to be covid positive when he/she carries the virus and is tested positive (true positive) or if he/she doesn't carry the virus and is yet declared as positive for the virus (false positive). This visualization maps all the confirmed cases for the virus across the globe. As observed, USA confirmed around 2 billion positive cases followed by Brazil with around 90 million. Russia and India are at par with around 45 million cases. The graph highlights these countries with a denser color as the confirmed cases increase which can be observed in the legend.



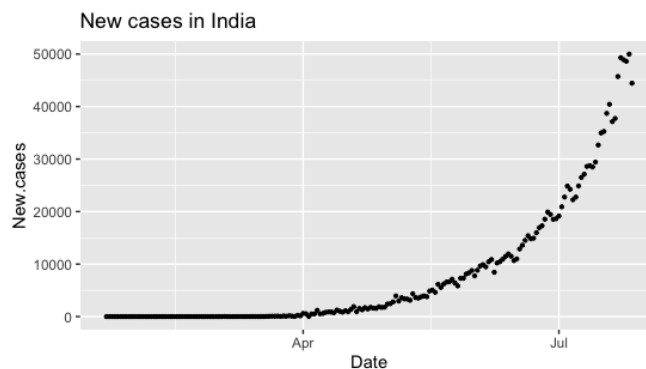
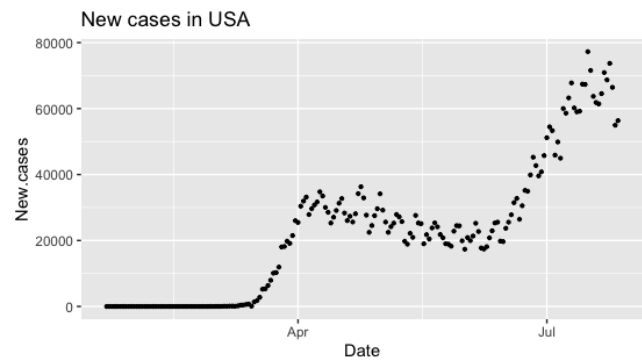
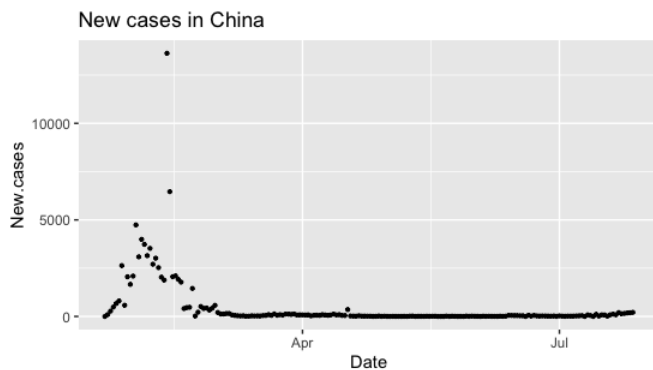
VISUALISATION 3:

This visualization is that of a scatter plot which depicts new covid cases against dates for a given country.

Problem based on initial iteration: As stated, the scatter plot maps data of new cases vs the date for China, USA and India. The visualization seen here seems ambiguous and unclear for the some months as seen in the graphs below. Additionally, we cannot scale the

data on a fixed y axis intervals. We cannot clearly interpret the pattern of this data. So, we need to design and plot an interactive graph such that the cases are easily analysed within the expected time frame.

Country	Months with unclear data
China	Till March
India	June onwards
USA	April onwards



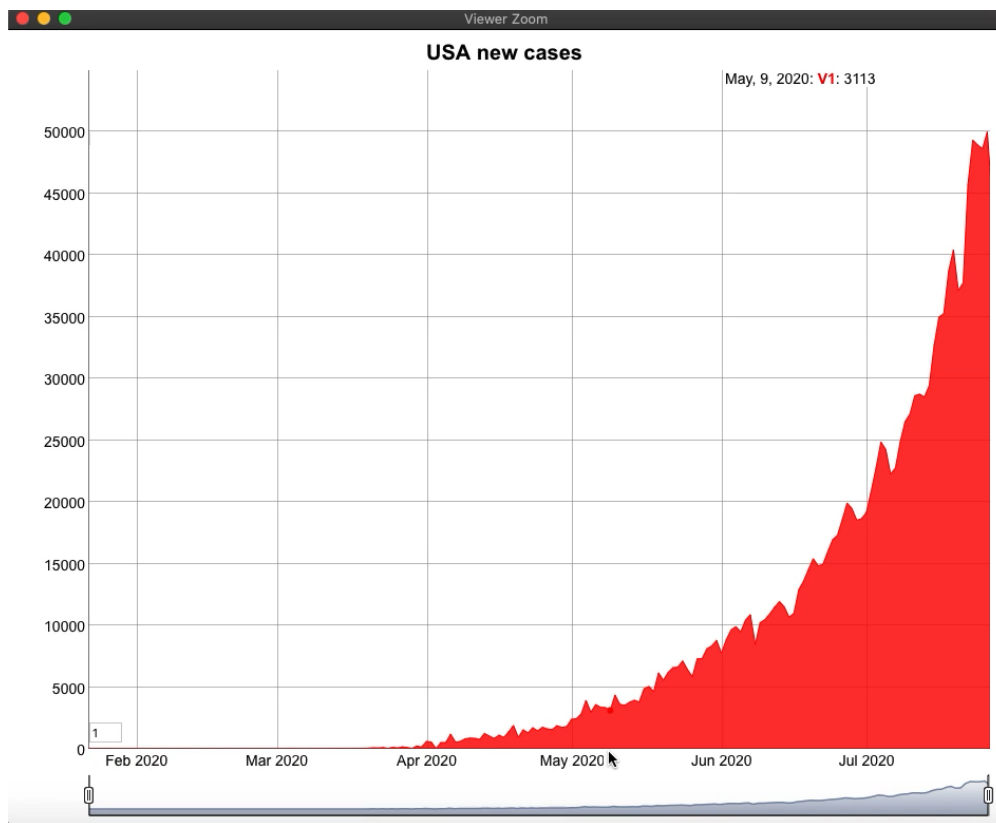
Functionalities added:

- *XTS (eXtensible Time Series)* is an extended zoo object and a combination of index and a matrix. To join the series, we need to call the XTS constructor and it should be in an ordered series so that it will create a matrix with its associated time for each observation.

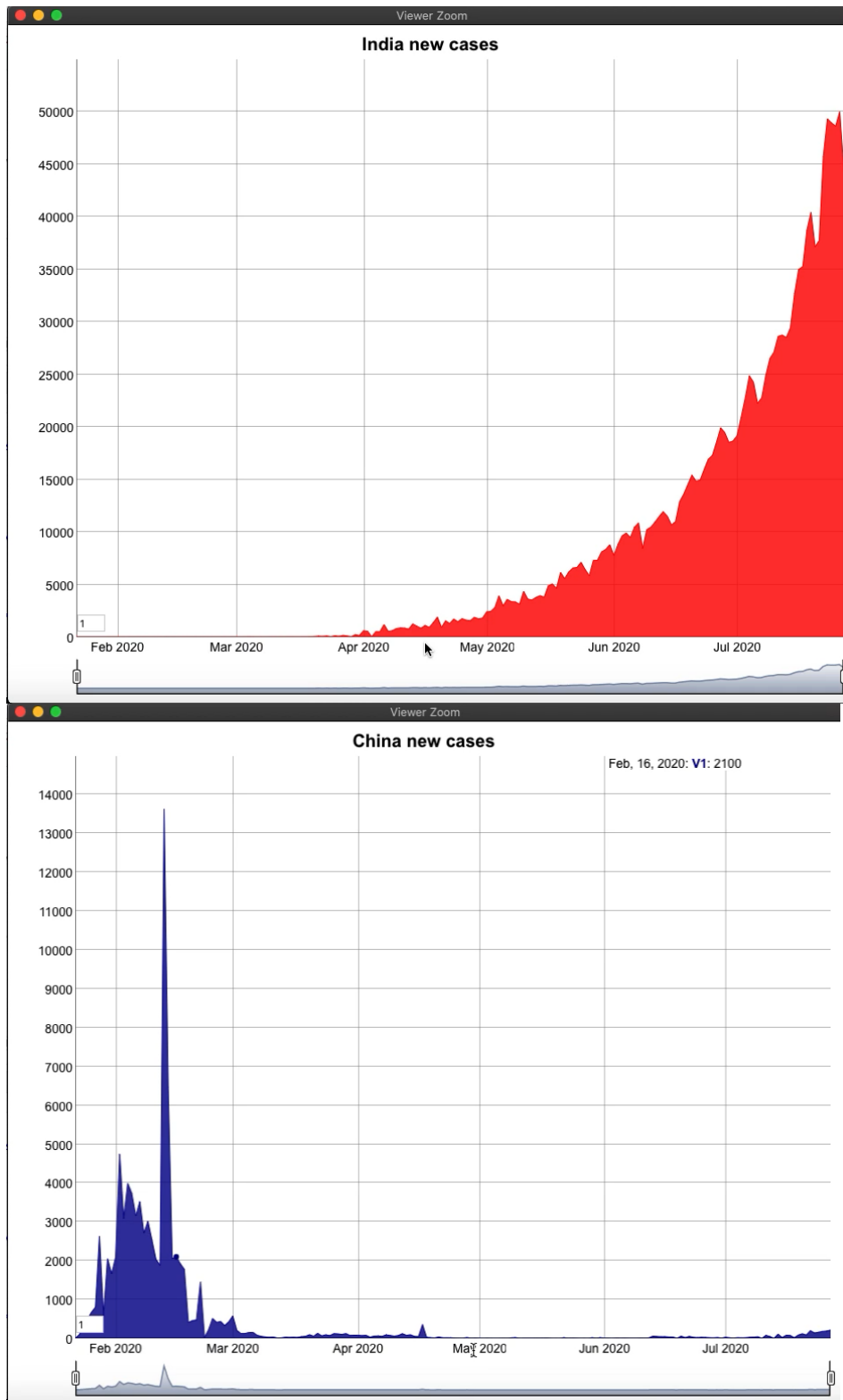
```
f= xts(x=b$New.cases, order.by = b$Date)
```

- *DYGRAPH* is used to plot the xts time series objects, it is the best way to plot timeseries as it provides:
 1. Configurable axis
 2. Zoom and highlighting
 3. Displaying the lower bars within the least difference of time frame
 4. Point annotations

It's a video file in reference with the above exploration, right click and play(or just double click) this for further insights.



It's a video file in reference with the above exploration, right click and play(or just double click) this for further insights.



Conclusion:

Early on in the COVID-19 outbreak, China implemented intense measures to keep people physically distant and other restrictions in an attempt to slow the spread of the coronavirus that causes the disease. According to a study published recently in *Science*, those measures succeeded. The Chinese measures were stricter, which gave the virus fewer opportunities to jump to a new person. The number of contacts among people fell by a factor of between seven and nine during the lockdown. Meanwhile in USA, where measures to keep people physically distant from each other were in place for a month or more, depending on the state, the outbreak has not stopped as seen in the graph. China had immediately halted all air travels, whereas in USA, the process was slower and delayed. On the other hand, virus is spreading at the highest pace in India, adding more 70,000 new infections each day. India's GDP performance has traditionally been extremely weak. So, the virus hit the country where it was already weak. This led to below the poverty line crowd to move towards their state for work opportunities, hence an increase in cases.