Analysis of COVID-19 Pandemic

2022MCS120009 (Santhosh Kumar N)

2022-11-18

Objective

To review and critically appraise published and preprint reports of prediction models for diagnosing coronavirus disease 2019 (covid-19) in patients with suspected infection, for prognosis of patients with covid-19, and for detecting people in the general population at risk of being admitted to hospital for covid-19 pneumonia.

Preliminary analysis

You can clean, prepossess the data here.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
options(warn = -1)
```

```
### Read The input data
confirmedraw = read.csv("time_series_covid19_confirmed_global.csv")
deathsraw= read.csv("time_series_covid19_deaths_global.csv")
recoveredraw = read.csv("time_series_covid19_recovered_global.csv")

# Get the Dimention of the dataset
print(dim(confirmedraw))
```

```
## [1]  289 1017
```

```
print(dim(deathsraw))
```

```
## [1]  289 1001
```

```
print(dim(recoveredraw))
```

```
## [1]  274 1001
```

```
#Sample data

head(recoveredraw[1,1:10])
```

```
##   Province.State Country.Region      Lat     Long X1.22.2020 X1.23.2020
## 1                   Afghanistan 33.93911 67.70995          0          0
##   X1.24.2020 X1.25.2020 X1.26.2020 X1.27.2020
## 1          0          0          0          0
```

```
# Function for initial Preprocessing Activity to Create and Structure the raw data
formatTheData = function(data,colName) {

    # Rename the lengthy columns
    covid_data = data %>% rename('subregion' = 'Province.State','country' = 'Country.Region')

    # Explude not requied columns
    covid_data = covid_data %>%  pivot_longer(cols = -one_of('country','subregion','Lat','Long')
                ,names_to = 'date'
               ,values_to = colName
               )

    # Remove 'X' on the Dataframe header
    covid_data$date <- covid_data$date %>% sub("X", "", .)

    # COnver to dateformat
    covid_data$date = mdy(covid_data$date)

    # Sort the data in ascending order
    covid_data = covid_data %>% arrange(date)

    # Round off the values to 3 decimal on Lat and Long
    covid_data$Lat = round(covid_data$Lat,3)
    covid_data$Long = round(covid_data$Long,3)

    #covid_data = merge(countryToContinent,covid_data,  by.y = "Country", by.x = "Country.Region")

    print(unique(format(covid_data$date, "%Y")))
    print(dim(covid_data))

    # Have observed that some issue in canadat nation data where recovered and Death and Conf cases not
    covid_data  = covid_data[covid_data$country != 'canada',]
    return(data.frame(covid_data))

}
```

```
case.conf = formatTheData(confirmedraw,"conf")
```

```
## [1] "2020" "2021" "2022"
## [1] 292757      6
```

```
case.death = formatTheData(deathsraw,"death")
```

```
## [1] "2020" "2021" "2022"
## [1] 288133      6
```

```
case.recov = formatTheData(recoveredraw,"recov")
```

```
## [1] "2020" "2021" "2022"
## [1] 273178      6
```

```
head(case.conf)
```

```
##   subregion     country     Lat   Long       date conf
## 1           Afghanistan  33.939 67.710 2020-01-22    0
## 2               Albania  41.153 20.168 2020-01-22    0
## 3               Algeria  28.034  1.660 2020-01-22    0
## 4               Andorra  42.506  1.522 2020-01-22    0
## 5                Angola -11.203 17.874 2020-01-22    0
## 6            Antarctica -71.950 23.347 2020-01-22    0
```

```
## we have to remove recovered data for Canada due to mismatch issue (Canada recovered data is counted

case.conf = case.conf[case.conf$country!= 'Canada' & case.conf$country!= 'China',]
case.death = case.death[case.death$country!= 'Canada' & case.death$country!= 'China',]
case.recov = case.recov[case.recov$country!= 'Canada' & case.recov$country!= 'China',]
```

```
#head(case.conf)
head(case.conf %>% arrange(country))
```

```
##   subregion     country    Lat  Long       date conf
## 1           Afghanistan 33.939 67.71 2020-01-22    0
## 2           Afghanistan 33.939 67.71 2020-01-23    0
## 3           Afghanistan 33.939 67.71 2020-01-24    0
## 4           Afghanistan 33.939 67.71 2020-01-25    0
## 5           Afghanistan 33.939 67.71 2020-01-26    0
## 6           Afghanistan 33.939 67.71 2020-01-27    0
```

```
tail(case.conf)
```

```
##        subregion               country     Lat    Long       date     conf
## 292752                         Vietnam  14.058 108.277 2022-10-30 11502474
## 292753              West Bank and Gaza  31.952  35.233 2022-10-30   703014
## 292754            Winter Olympics 2022  39.904 116.407 2022-10-30      535
## 292755                           Yemen  15.553  48.516 2022-10-30    11939
## 292756                          Zambia -13.134  27.849 2022-10-30   333681
## 292757                        Zimbabwe -19.015  29.155 2022-10-30   257893
```

```
tail(case.death)
```

```
##        subregion               country     Lat    Long       date death
## 288128                         Vietnam  14.058 108.277 2022-10-14 43155
## 288129              West Bank and Gaza  31.952  35.233 2022-10-14  5707
## 288130            Winter Olympics 2022  39.904 116.407 2022-10-14     0
## 288131                           Yemen  15.553  48.516 2022-10-14  2158
## 288132                          Zambia -13.134  27.849 2022-10-14  4017
## 288133                        Zimbabwe -19.015  29.155 2022-10-14  5605
```

```
full_date = full_join(case.conf, case.death, by = c('subregion','country','Lat','Long', 'date'))

country = full_join(full_date, case.recov, by = c('subregion','country','Lat','Long', 'date'))
```

```
sum(is.na(country))
```

```
## [1] 7648
```

```
# Remove NA and Nan data from the processed data alsi negative values
country = country %>% filter(!is.na(Lat) & !is.na(Long) & !is.na(conf) & !is.na(death) & !is.na(recov))

country = country[country$recov >=0 & country$death >=0 & country$conf >=0,]

sum(is.na(country))
```

```
## [1] 0
```

```
dim(country)
```

```
## [1] 238275      8
```

```
length(unique(country$country))
```

```
## [1] 199
```

```
country$Active = country$conf -  country$death - country$recov

country = country[country$Active>=0 & country$recov>=0,]

dim(country)
```

Compute Active Cases

```
## [1] 237159      9
```

```
head(country[country$recov !=0,])
```

```
##      subregion  country    Lat    Long       date conf death recov Active
## 203            Thailand 15.870 100.993 2020-01-22    4     0     2      2
## 442            Thailand 15.870 100.993 2020-01-23    4     0     2      2
## 681            Thailand 15.870 100.993 2020-01-24    5     0     3      2
## 920            Thailand 15.870 100.993 2020-01-25    6     0     3      3
## 1063              Japan 36.205 138.253 2020-01-26    4     0     1      3
## 1159           Thailand 15.870 100.993 2020-01-26    8     0     6      2
```

```
getNewCases = function(nation,col,newcolname){
    df = country
    df = df[df$country == nation,]

    df = df %>% arrange(date)
    newCases = df[2:dim(df)[1],][,col] - df[1:dim(df)[1],][,col]
    newCases = append(newCases,0,0)

    newCases = newCases[-length(newCases)]
    df$newCases = as.integer(0)
    names(df)[names(df) == "newCases"] <- newcol

    df[,newcol] <- newCases

    return(df)
}
```

Get 'NewCases' 'NewDeath' and 'New Recovered' 'New Active' details

```
country_newCases <- country
country_newCases$newCases = as.integer(0)
country_newCases <- country_newCases[0,]

newcol = "newCases"
```

```r
for(c in unique(country$country))
    {
    country_newCases = rbind(country_newCases,getNewCases(c,'conf','newCases'))
}
```

Get New Cases

```r


country_newDeath <- country
country_newDeath$newDeath = as.integer(0)
country_newDeath <- country_newDeath[0,]

col = "death"
newcol = "newDeath"

for(c in unique(country$country))
    {
    country_newDeath = rbind(country_newDeath,getNewCases(c,col,newcol))
}
```

Get New Death Cases

```r


country_newRecovered <- country
country_newRecovered$newRecovered = as.integer(0)
country_newRecovered <- country_newRecovered[0,]

col = "recov"
newcol = "newRecovered"

for(c in unique(country$country))
    {
    country_newRecovered = rbind(country_newRecovered,getNewCases(c,col,newcol))
}
```

New Recovered Cases

```r


country_newActive <- country
country_newActive$newActive = as.integer(0)
country_newActive <- country_newActive[0,]

col = "Active"
newcol = "newActive"

for(c in unique(country$country))
    {
    country_newActive = rbind(country_newActive,getNewCases(c,col,newcol))
}
```

New Active Cases

```
fulldata1 = full_join(country_newCases, country_newDeath, by = c('subregion','country','Lat',
                                                            'Long', 'date','conf','death',
                                                            'recov','Active'))
```

```
fulldata2 = full_join(fulldata1, country_newActive, by = c('subregion','country','Lat',
                                                            'Long', 'date','conf','death',
                                                            'recov','Active'))
```

```
country = full_join(fulldata2, country_newRecovered, by = c('subregion','country','Lat',
                                                            'Long', 'date','conf','death',
                                                            'recov','Active'))
```

```
head(country)
```

Merge All cases

```
##   subregion     country    Lat  Long       date conf death recov Active
## 1           Afghanistan 33.939 67.71 2020-01-22    0     0     0      0
## 2           Afghanistan 33.939 67.71 2020-01-23    0     0     0      0
## 3           Afghanistan 33.939 67.71 2020-01-24    0     0     0      0
## 4           Afghanistan 33.939 67.71 2020-01-25    0     0     0      0
## 5           Afghanistan 33.939 67.71 2020-01-26    0     0     0      0
## 6           Afghanistan 33.939 67.71 2020-01-27    0     0     0      0
##   newCases newDeath newActive newRecovered
## 1        0        0         0            0
## 2        0        0         0            0
## 3        0        0         0            0
## 4        0        0         0            0
## 5        0        0         0            0
## 6        0        0         0            0
```

Detailed analysis

```
# Total cases reported so far

sum(country$newCases)
```

```
## [1] 606354330
```

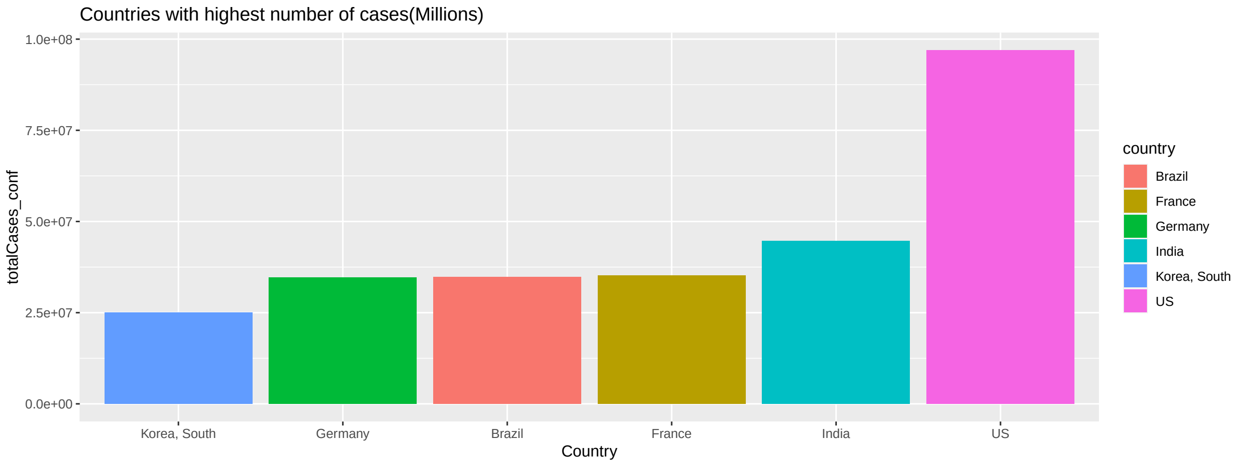
```
# Total cases reported in India
sum(country[country$country == "India",]$newCases)
```

```
## [1] 44626427
```

```
## Confirmed Cases Country wise

Countrywise_ConfCases = country %>% group_by(country) %>%
                                        summarise(totalCases_conf = sum(newCases)) %>%
                                        arrange(desc(totalCases_conf))

top_10_Country_Highest_ConfCases = head(Countrywise_ConfCases,10)

head(top_10_Country_Highest_ConfCases)
```

```
## # A tibble: 6 x 2
##   country     totalCases_conf
##   <chr>                 <dbl>
## 1 US                 96931266
## 2 India              44626427
## 3 France             35178403
## 4 Brazil             34746462
## 5 Germany            34608835
## 6 Korea, South       25098995
```

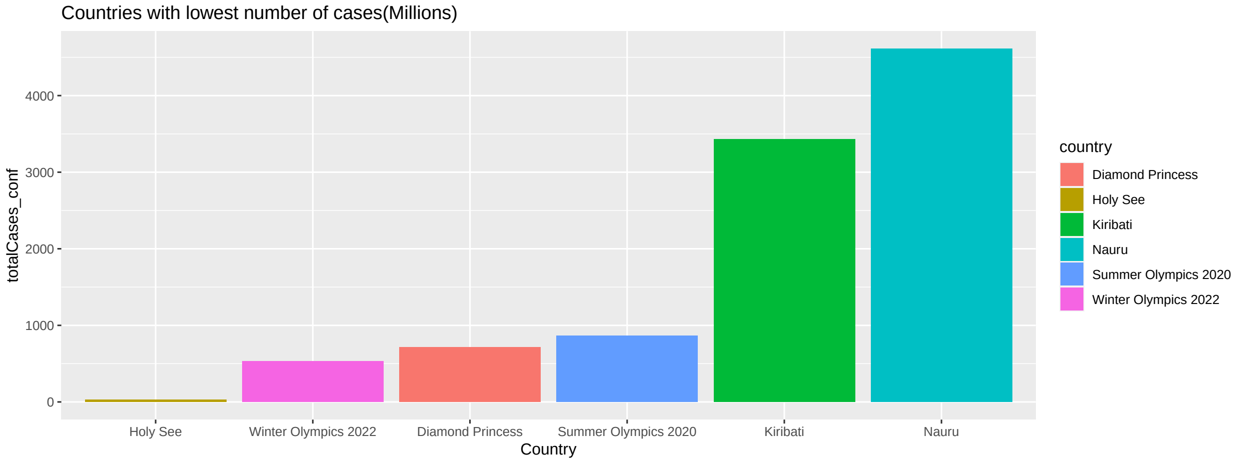
```
# Visualize the same using ggplot
ggplot(data = head(top_10_Country_Highest_ConfCases),aes(x = reorder(country,totalCases_conf),
                                                         y = totalCases_conf,fill = country)) + geom_ba
                    labs(title = "Countries with highest number of cases(Millions)") + xlab("Country")
```



```
# Lowest(Top 10) cases reported

top_10_Country_lowest_ConfCases = tail(Countrywise_ConfCases %>% arrange(desc(totalCases_conf)),10)

top_10_Country_lowest_ConfCases
```

```
## # A tibble: 10 x 2
##    country              totalCases_conf
##    <chr>                          <dbl>
##  1 Nauru                           4611
##  2 Kiribati                        3430
##  3 Summer Olympics 2020             865
##  4 Diamond Princess                 712
##  5 Winter Olympics 2022             535
##  6 Holy See                          29
##  7 Tuvalu                            20
##  8 Antarctica                        11
##  9 MS Zaandam                         9
## 10 Korea, North                       0
```

```
# Visualize the same using ggplot
ggplot(data = head(top_10_Country_lowest_ConfCases),aes(x = reorder(country,totalCases_conf),
                                                        y = totalCases_conf,fill = country)) + geom_ba
                    labs(title = "Countries with lowest number of cases(Millions)") + xlab("Country")
```

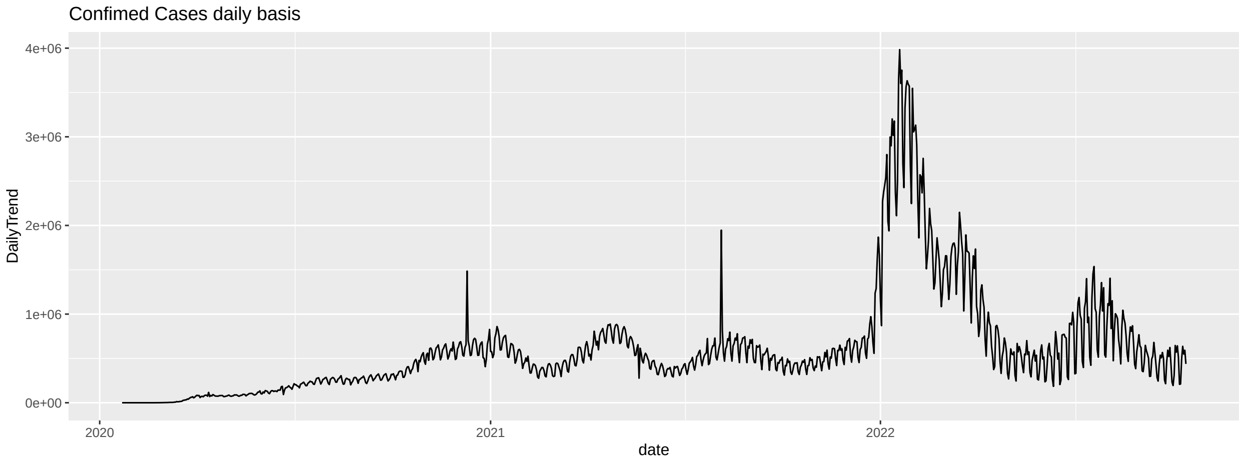


```
# Daily Trend on Confirmed Cases
conf_Cases_daily = data.frame(country %>% group_by(date) %>% summarise(DailyTrend = sum(newCases)))

head(conf_Cases_daily)
```

```
##         date DailyTrend
## 1 2020-01-22          0
## 2 2020-01-23          3
## 3 2020-01-24          9
## 4 2020-01-25          7
## 5 2020-01-26         10
## 6 2020-01-27          6
```

```
conf_Cases_daily$DailyTrend <- ifelse(conf_Cases_daily$DailyTrend < 0,0,conf_Cases_daily$DailyTrend)
```

```
ggplot(conf_Cases_daily,aes(x = date, y = DailyTrend)) +
        geom_line() +
        labs(title = "Confimed Cases daily basis")
```

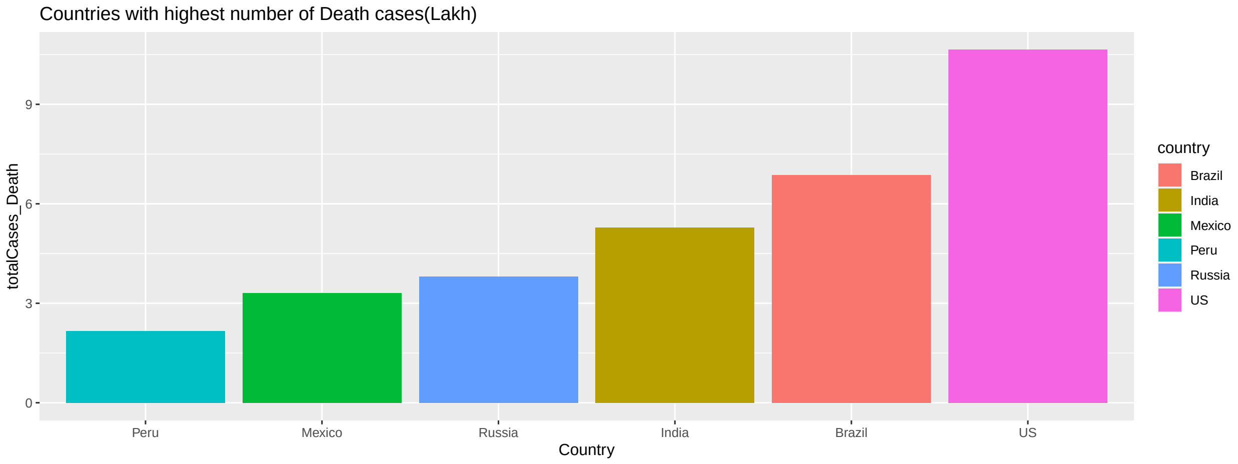


```
# Look ate Death Cases Countrywise

death_Cases_country = country %>% group_by(country) %>%
                                        summarise(totalCases_Death = sum(newDeath)) %>%
                                        arrange(desc(totalCases_Death))

top_10_Country_Highest_deathCases = head(death_Cases_country %>% arrange(desc(totalCases_Death)),10)

top_10_Country_Highest_deathCases$totalCases_Death =  top_10_Country_Highest_deathCases$totalCases_Deat
```

```
top_10_Country_Highest_deathCases
```

```
## # A tibble: 10 x 2
##    country        totalCases_Death
##    <chr>                     <dbl>
##  1 US                        10.7
##  2 Brazil                     6.87
##  3 India                      5.29
##  4 Russia                     3.81
##  5 Mexico                     3.30
##  6 Peru                       2.17
##  7 United Kingdom             2.08
##  8 Italy                      1.78
##  9 Indonesia                  1.58
## 10 France                     1.52
```

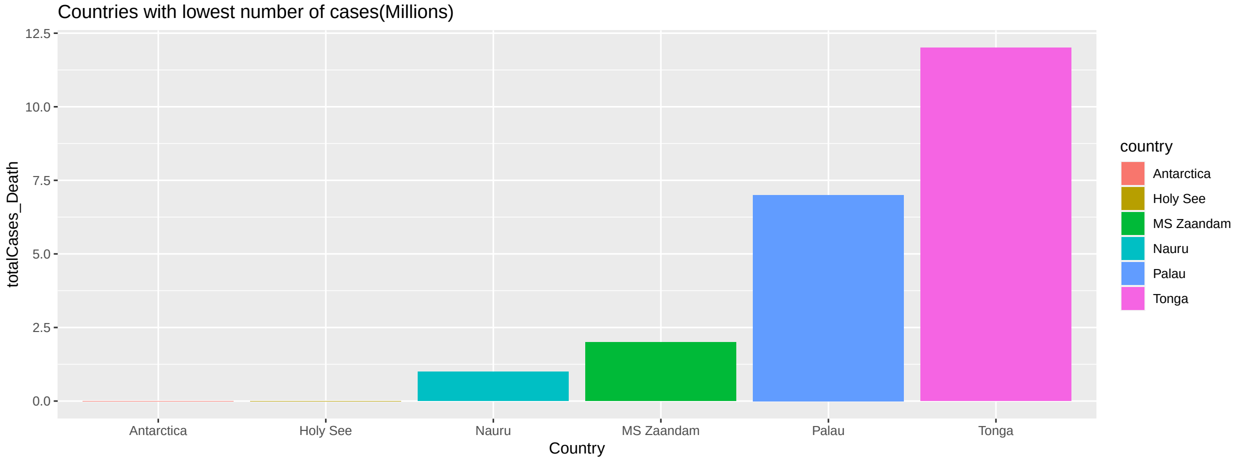
```
ggplot(data = head(top_10_Country_Highest_deathCases),aes(x = reorder(country,totalCases_Death),
                                                         y = totalCases_Death,fill = country)) + geom_ba
                    labs(title = "Countries with highest number of Death cases(Lakh)") + xlab("Country"
```



```
tail(death_Cases_country)
```

```
## # A tibble: 6 x 2
##   country              totalCases_Death
##   <chr>                           <dbl>
## 1 Antarctica                          0
## 2 Holy See                            0
## 3 Korea, North                        0
## 4 Summer Olympics 2020                0
## 5 Tuvalu                              0
## 6 Winter Olympics 2022                0
```

```
top_10_Country_lowest_DeathCases = tail(death_Cases_country %>% arrange(desc(totalCases_Death)),10)

ggplot(data = head(top_10_Country_lowest_DeathCases),aes(x = reorder(country,totalCases_Death),
                                                         y = totalCases_Death,fill = country)) + geom_ba
                    labs(title = "Countries with lowest number of cases(Millions)") + xlab("Country")
```

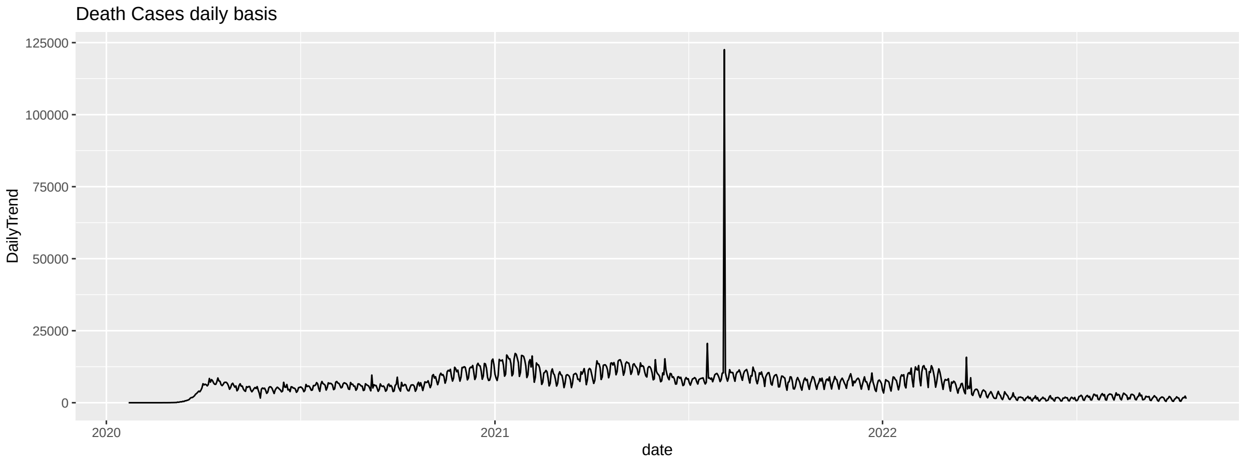


```
death_Cases_daily = data.frame(country %>% group_by(date) %>% summarise(DailyTrend = sum(newDeath)))

head(death_Cases_daily)
```

```
##          date DailyTrend
## 1 2020-01-22          0
## 2 2020-01-23          0
## 3 2020-01-24          0
## 4 2020-01-25          0
## 5 2020-01-26          0
## 6 2020-01-27          0
```

```
death_Cases_daily$DailyTrend <- ifelse(death_Cases_daily$DailyTrend < 0,0,death_Cases_daily$DailyTrend)
```

```
ggplot(death_Cases_daily,aes(x = date, y = DailyTrend)) +
        geom_line() +
        labs(title = "Death Cases daily basis")
```



#### ‘Recovered’ Cases Country wise

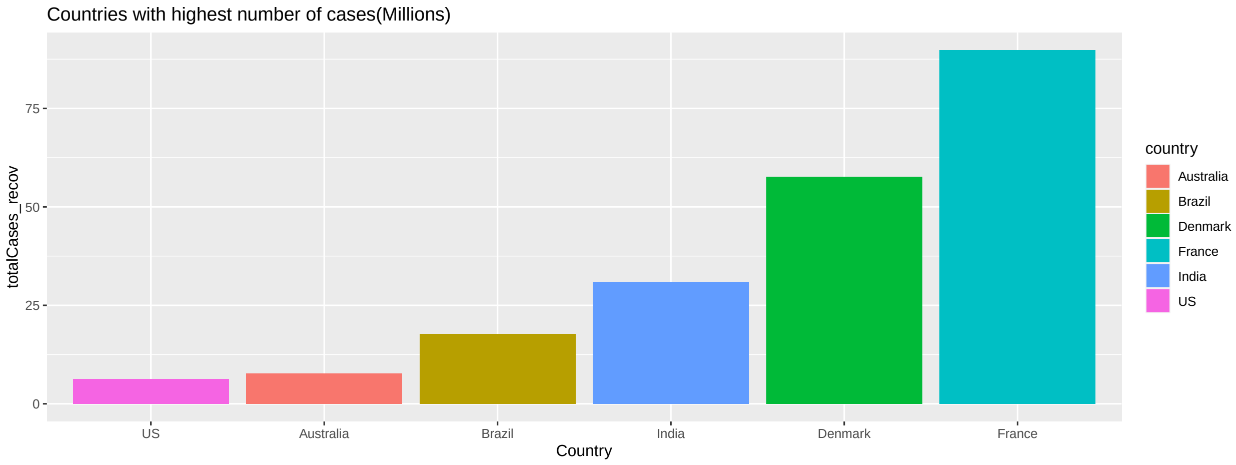
```
recov_Cases_country <- data.frame(country = NA,totalCases = NA)

df = country

getRecovCasesCounrywise = function(country)
{
    df = df[df$country == country,]

    df$recov_daily = df[2:dim(df)[1],]$recov - df[1:dim(df)[1],]$recov

    df$recov_daily <- ifelse(df$recov_daily < 0,0,df$recov_daily)

    return (data.frame(country = country,totalCases = sum(df$recov_daily)))
}

for(c in unique(country$country))
    {
    recov_Cases_country = rbind(recov_Cases_country,getRecovCasesCounrywise(c))
}

recov_Cases_country = recov_Cases_country %>% filter(!is.na(totalCases))

colnames(recov_Cases_country)[2] <- "totalCases_recov"
```

```
top_10_Country_Highest_recovCases = head(recov_Cases_country %>% arrange(desc(totalCases_recov)),10)

top_10_Country_Highest_recovCases$totalCases_recov =  top_10_Country_Highest_recovCases$totalCases_reco
```

```
top_10_Country_Highest_recovCases
```

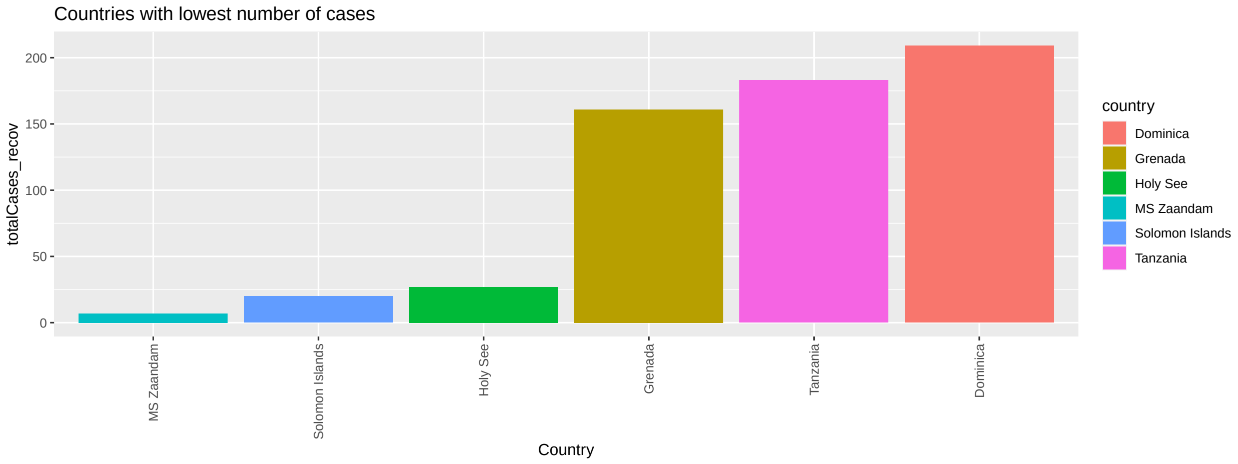
```
##      country totalCases_recov
## 1     France        89.757724
## 2    Denmark        57.572798
## 3      India        30.977006
## 4     Brazil        17.771228
## 5  Australia         7.734785
## 6         US         6.303715
## 7     Russia         5.609682
## 8     Turkey         5.478185
## 9   Colombia         4.681505
## 10 Argentina         4.615834
```

```
ggplot(data = head(top_10_Country_Highest_recovCases),aes(x = reorder(country,totalCases_recov),
                                                          y = totalCases_recov,fill = country)) +
                geom_bar(stat = "identity") +
                    labs(title = "Countries with highest number of cases(Millions)") + xlab("Country")
```



```
top_10_Country_lowest_recovCases = tail(recov_Cases_country %>% arrange(desc(totalCases_recov)),20)
```

```
ggplot(data = head(top_10_Country_lowest_recovCases),aes(x = reorder(country,totalCases_recov),
                                                         y = totalCases_recov,fill = country)) +
                geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
                    labs(title = "Countries with lowest number of cases") + xlab("Country")
```

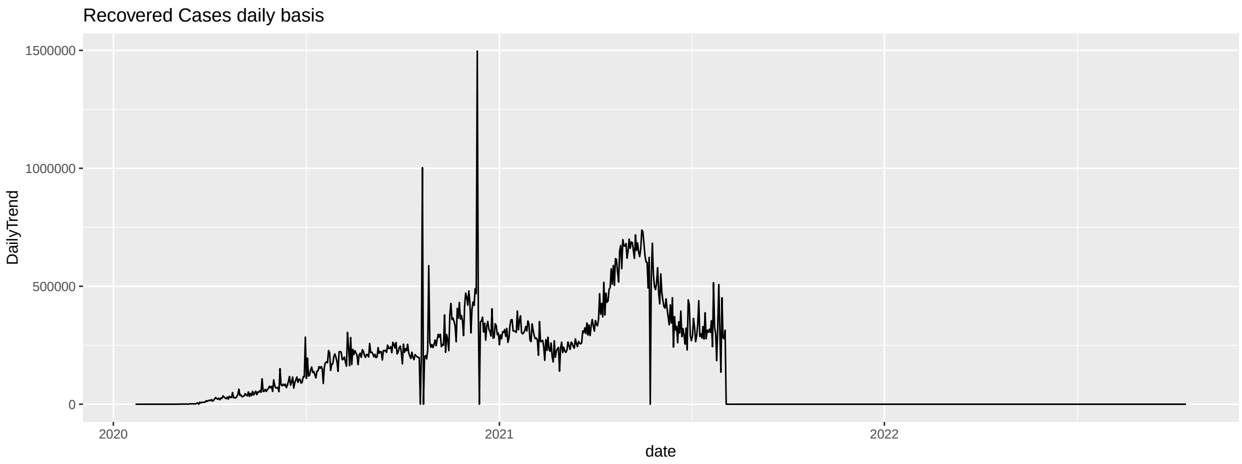

Countries with lowest number of cases

```
recov_Cases_daily = data.frame(country %>% group_by(date) %>% summarise(recovCases = sum(recov)))

head(recov_Cases_daily)
```

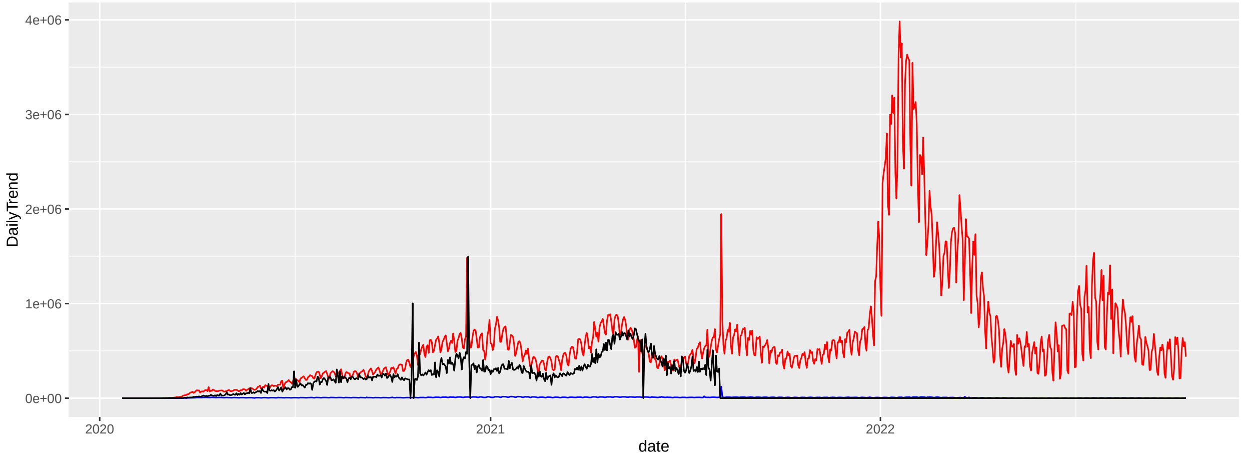
```
##         date recovCases
## 1 2020-01-22          2
## 2 2020-01-23          2
## 3 2020-01-24          3
## 4 2020-01-25          3
## 5 2020-01-26          7
## 6 2020-01-27          7
```

```
recov_Cases_daily$DailyTrend = recov_Cases_daily[2:dim(recov_Cases_daily)[1],'recovCases'] -
                                    recov_Cases_daily[1:dim(recov_Cases_daily)[1],'recovCases']

recov_Cases_daily$DailyTrend <- ifelse(recov_Cases_daily$DailyTrend < 0,0,recov_Cases_daily$DailyTrend)
```

```
ggplot(recov_Cases_daily,aes(x = date, y = DailyTrend)) +
        geom_line() +
        labs(title = "Recovered Cases daily basis")
```


Recovered Cases daily basis

```
#### All kind of cases in one plot daily basis
ggplot() +
  geom_line(data =conf_Cases_daily, aes(x=date,y=DailyTrend),color='red') +
  geom_line(data =death_Cases_daily, aes(x=date,y=DailyTrend),color='blue') +
  geom_line(data =recov_Cases_daily, aes(x=date,y=DailyTrend),color='black') +
     geom_point()
```

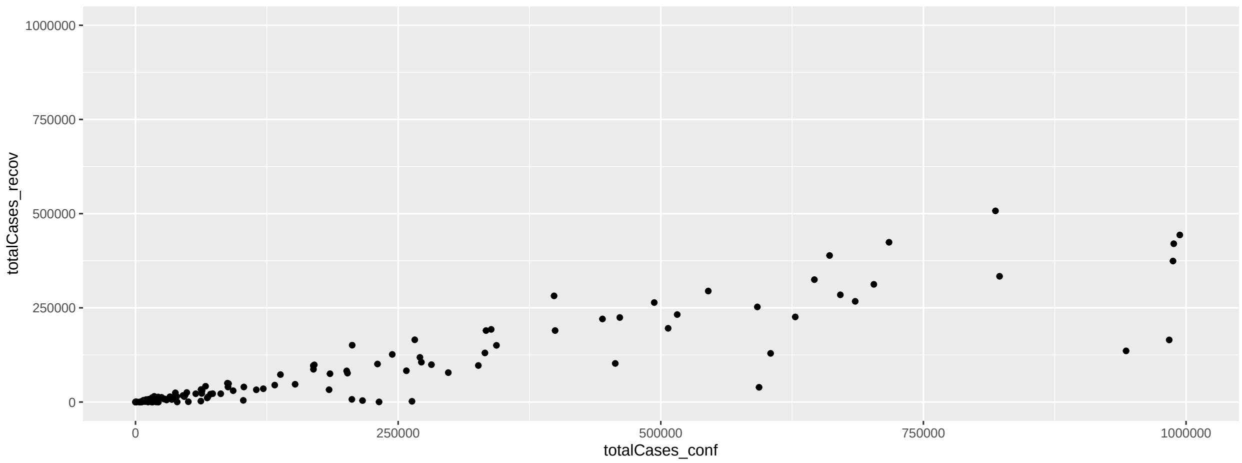


```
df_list <- list(Countrywise_ConfCases , death_Cases_country, recov_Cases_country)
df_list <- df_list %>% reduce(full_join, by='country')
```

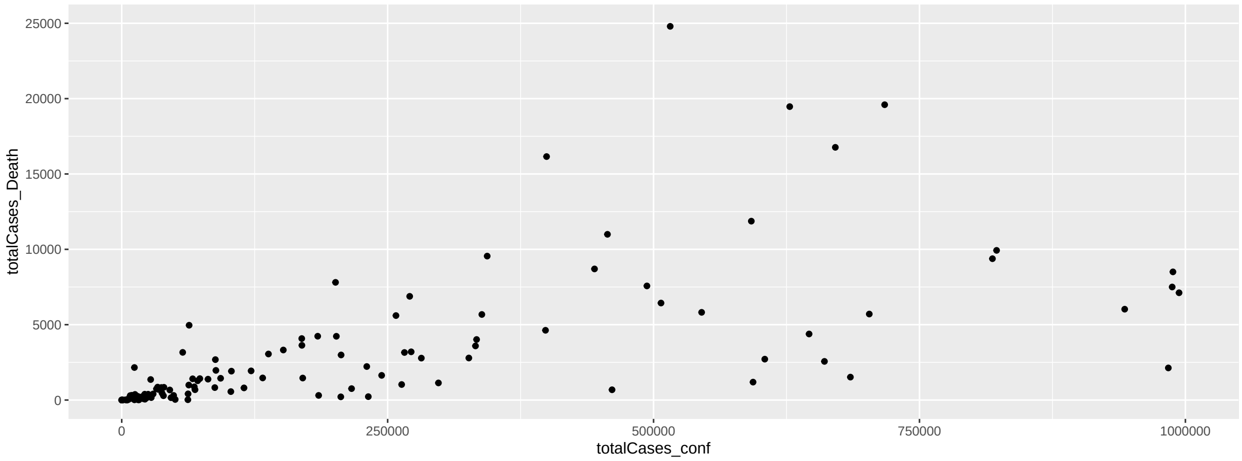
```
head(df_list)
```

```
## # A tibble: 6 x 4
##   country      totalCases_conf totalCases_Death totalCases_recov
##   <chr>                  <dbl>            <dbl>            <dbl>
## 1 US                  96931266          1065076          6303715
## 2 India               44626427           528874         30977006
## 3 France              35178403           152288         89757724
## 4 Brazil              34746462           687144         17771228
## 5 Germany             34608835           151420          3663580
## 6 Korea, South        25098995            28808           180736
```

```
#### Correlation between cases 'Confirmed' and 'Recovered'

ggplot(df_list,aes(x = totalCases_conf,y = totalCases_recov)) +
  geom_point() +
  ylim(0,1000000) +
  xlim(0,1000000)
```



```
#### Correlation between cases 'Confirmed' and 'Death'

ggplot(df_list,aes(x = totalCases_conf,y = totalCases_Death)) +
  geom_point() +
  ylim(0,25000) +
  xlim(0,1000000)
```



```
df_list$totalCases_conf = df_list$totalCases_conf
df_list$totalCases_death = df_list$totalCases_Death
df_list$totalCases_recov = df_list$totalCases_recov
```

```
head(df_list %>% arrange(desc(totalCases_conf)),10)
```

```
## # A tibble: 10 x 5
##    country        totalCases_conf totalCases_Death totalCases_recov totalCases~1
##    <chr>                    <dbl>            <dbl>            <dbl>        <dbl>
##  1 US                    96931266          1065076          6303715      1065076
##  2 India                 44626427           528874         30977006       528874
##  3 France                35178403           152288         89757724       152288
##  4 Brazil                34746462           687144         17771228       687144
##  5 Germany               34608835           151420          3663580       151420
##  6 Korea, South          25098995            28808           180736        28808
##  7 United Kingdom        23798793           207948          2260277       207948
##  8 Italy                 23030777           177883          4145492       177883
##  9 Japan                 21721502            45862           865938        45862
## 10 Russia                20975381           380854          5609682       380854
## # ... with abbreviated variable name 1: totalCases_death
```

```
#### Mortality Rate and Recovery Rate

df_list[df_list$country == 'India',]
```

```
## # A tibble: 1 x 5
##   country totalCases_conf totalCases_Death totalCases_recov totalCases_death
##   <chr>             <dbl>            <dbl>            <dbl>            <dbl>
## 1 India          44626427           528874         30977006           528874
```

```
df_list = mutate(df_list,recovery_rate = totalCases_recov / totalCases_conf)

df_list = mutate(df_list,mortality_rate = totalCases_death / totalCases_conf)
```

```
#### Order based on the highest 'death rate'

head(df_list %>% arrange(desc(mortality_rate)))
```

```
## # A tibble: 6 x 7
##   country    totalCases_conf totalCases_Death totalCas~1 total~2 recov~3 morta~4
##   <chr>                <dbl>            <dbl>      <dbl>   <dbl>   <dbl>   <dbl>
## 1 MS Zaandam               9                2          7       2   0.778  0.222
## 2 Yemen                11939             2158       4251    2158   0.356  0.181
## 3 Sudan                63375             4963      30647    4963   0.484  0.0783
## 4 Syria                57332             3163      22019    3163   0.384  0.0552
## 5 Peru               4150121           216844     789908  216844   0.190  0.0523
## 6 Somalia              27223             1361       7661    1361   0.281  0.0500
## # ... with abbreviated variable names 1: totalCases_recov, 2: totalCases_death,
## #   3: recovery_rate, 4: mortality_rate
```

```
#### Order based on the highest 'recovery_rate'

head(df_list %>% arrange(desc(recovery_rate)))
```

```
## # A tibble: 6 x 7
##   country          totalCases_conf totalCases_~1 total~2 total~3 recov~4 morta~5
##   <chr>                      <dbl>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Denmark                  3317617          7173  5.76e7    7173  17.4   2.16e-3
## 2 Australia                1164098           674  7.73e6     674   6.64  5.79e-4
## 3 France                  35178403        152288  8.98e7  152288   2.55  4.33e-3
## 4 Diamond Princess             712            13  7   e2      13   0.983 1.83e-2
## 5 Holy See                      29             0  2.7 e1       0   0.931 0
## 6 Tajikistan                 17786           125  1.50e4     125   0.845 7.03e-3
## # ... with abbreviated variable names 1: totalCases_Death, 2: totalCases_recov,
## #   3: totalCases_death, 4: recovery_rate, 5: mortality_rate
```

```
df_list <- list(Countrywise_ConfCases, death_Cases_country, recov_Cases_country)
df_list <- df_list %>% reduce(full_join, by='country')
```

```
#### World Reported Cases

sum(df_list$totalCases_conf) /1000000000 # in billion
```

```
## [1] 0.6063543
```

```
sum(df_list$totalCases_death) /1000000 # Million
```

```
## [1] 0
```

```
sum(df_list$totalCases_recov) /1000000 # Million
```

```
## [1] 296.3749
```

```
#### Average Cases worldwide Recoved on daily wise

# recov_Cases_daily

recov_Cases_daily %>% summarise("mean( in lakh)" = mean(DailyTrend)) /100000
```

```
##   mean( in lakh)
## 1       1.342568
```

```
#### Average Cases worldwide Confirmed on daily wise

conf_Cases_daily %>% summarise("mean( in lakh)" = mean(DailyTrend)) / 100000
```

```
##   mean( in lakh)
## 1       6.081789
```

```
#### Average Cases worldwide Recovered on daily wise

death_Cases_daily %>% summarise("mean( in thousands)" = mean(DailyTrend))/ 1000
```

```
##   mean( in thousands)
## 1             6.50374
```

```
#### Plot Recovery and Death Rate

tail(country)
```

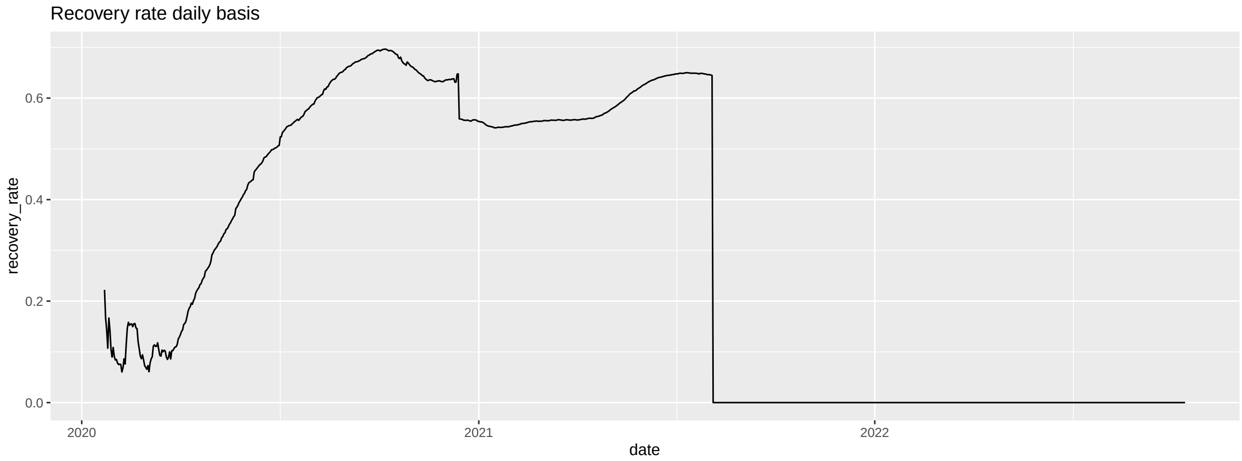
```
##        subregion  country     Lat   Long       date   conf death recov Active
## 237154           Zimbabwe -19.015 29.155 2022-10-09 257655  5604     0 252051
## 237155           Zimbabwe -19.015 29.155 2022-10-10 257749  5604     0 252145
## 237156           Zimbabwe -19.015 29.155 2022-10-11 257749  5604     0 252145
## 237157           Zimbabwe -19.015 29.155 2022-10-12 257798  5604      0 252194
## 237158           Zimbabwe -19.015 29.155 2022-10-13 257827  5605      0 252222
## 237159           Zimbabwe -19.015 29.155 2022-10-14 257827  5605      0 252222
##         newCases newDeath newActive newRecovered
## 237154        0        0         0            0
## 237155       94        0        94            0
## 237156        0        0         0            0
## 237157       49        0        49            0
## 237158       29        1        28            0
## 237159        0        0         0            0
```

```
daily = country %>% select('date','recov','conf','death') %>% group_by(date) %>% summarise(total_recov =
                                                                                           total_conf =
                                                                                           total_death =


head(daily)
```

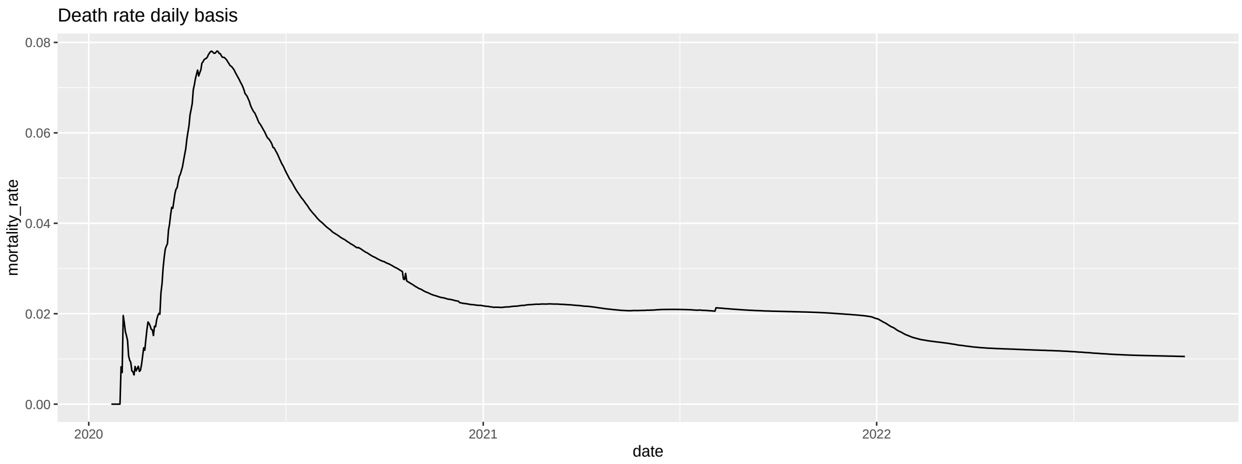
```
## # A tibble: 6 x 4
##   date       total_recov total_conf total_death
##   <date>           <int>      <int>       <int>
## 1 2020-01-22           2          9           0
## 2 2020-01-23           2         12           0
## 3 2020-01-24           3         21           0
## 4 2020-01-25           3         28           0
## 5 2020-01-26           7         42           0
## 6 2020-01-27           7         49           0
```

```
daily = mutate(daily,recovery_rate = total_recov / total_conf)

daily = mutate(daily,mortality_rate = total_death / total_conf)
```

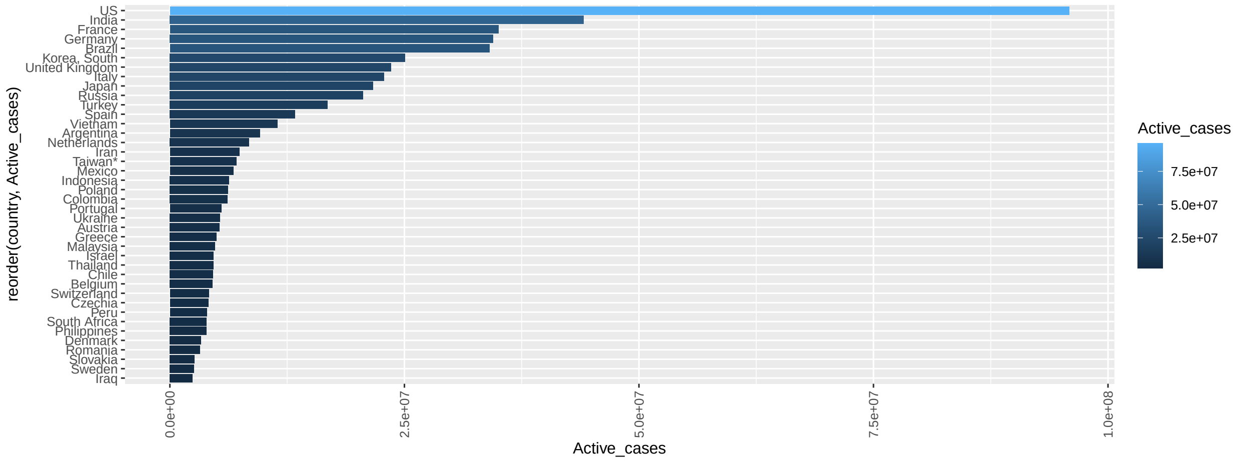
```
ggplot(daily,aes(x = date, y = recovery_rate)) +
        geom_line() +
        labs(title = "Recovery rate daily basis")
```



```
ggplot(daily,aes(x = date, y = mortality_rate)) +
        geom_line() +
        labs(title = "Death rate daily basis")
```



```
#### Analysize Active Cases coun ry wise

data = head(country %>% group_by(country) %>% summarise(Active_cases = sum(newActive)) %>% arrange(desc

ggplot(data,aes(x = reorder(country,Active_cases),y = Active_cases,fill = Active_cases)) +
geom_bar(stat = "identity") + coord_flip() +theme(axis.text.x = element_text(angle = 90, vjust = 0.5, h
```

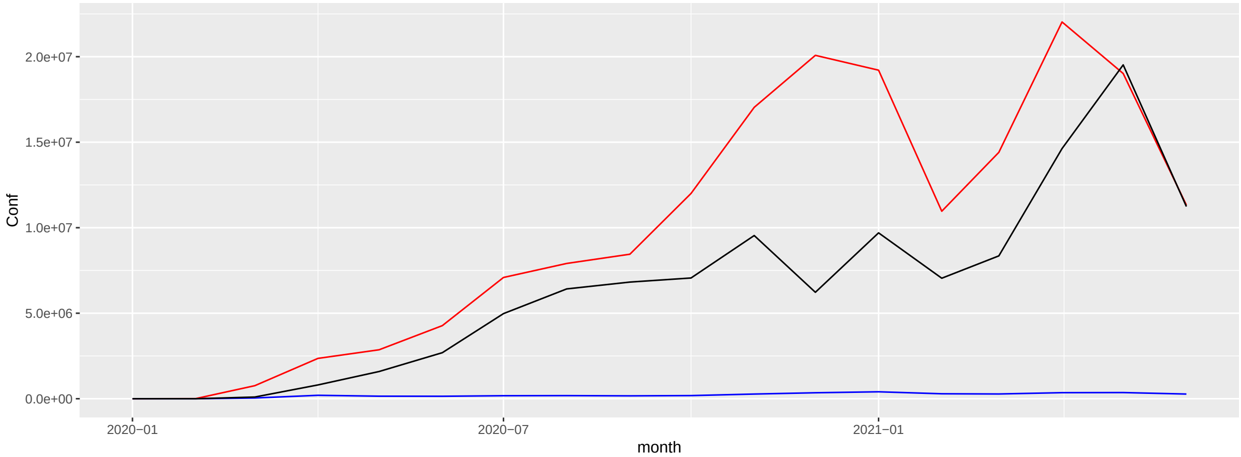


```
#### Group by Month

monthwise_cases= country %>% group_by(month = lubridate::floor_date(date, 'month')) %>%
    summarize(Conf = sum(newCases),death = sum(newDeath),recov = sum(newRecovered))

monthwise_cases = head(monthwise_cases,18)
```

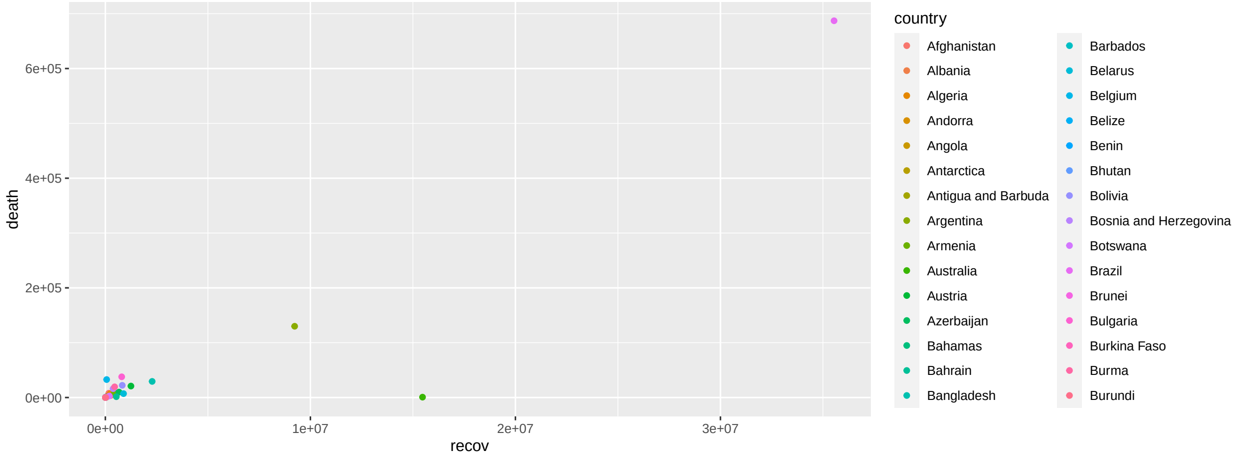
```
monthwise_cases
```

```
## # A tibble: 18 x 4
##    month           Conf  death    recov
##    <date>         <dbl>  <dbl>    <dbl>
##  1 2020-01-01       103      1        7
##  2 2020-02-01      6499    108      439
##  3 2020-03-01    762811  43604    99048
##  4 2020-04-01   2359106 200814   806871
##  5 2020-05-01   2862985 148917  1592156
##  6 2020-06-01   4273897 144969  2692352
##  7 2020-07-01   7089298 176092  4976782
##  8 2020-08-01   7909437 182243  6421098
##  9 2020-09-01   8449670 168109  6821165
## 10 2020-10-01  12002950 183340  7060183
## 11 2020-11-01  17038470 273980  9542802
## 12 2020-12-01  20078600 349601  6224141
## 13 2021-01-01  19210656 406370  9698931
## 14 2021-02-01  10963628 290286  7046702
## 15 2021-03-01  14407870 278149  8352511
## 16 2021-04-01  22029954 354387 14627537
## 17 2021-05-01  19015777 360833 19520798
## 18 2021-06-01  11323493 273871 11239401
```

```
ggplot() + geom_line(data =monthwise_cases, aes(x=month,y=Conf),color='red') +
           geom_line(data =monthwise_cases, aes(x=month,y=death),color='blue') +
           geom_line(data =monthwise_cases, aes(x=month,y=recov),color='black') + geom_point()
```



```
#### Analyze 'Recovered' and Death' rate

country$newRecovered = abs(country$newRecovered)

d = head(country %>% select('country','newRecovered','newDeath') %>%
                group_by(country)  %>%
                    summarise(recov = sum(newRecovered), death = sum(newDeath)),30)

ggplot(d,aes(x = recov, y = death,color = country)) + geom_point()
```



This section should describe the inferences you have made.

Outcomes

‘United states’ has highest covid cases reported so far with number of 100,456,053 ‘India’ is in second place where total reported cases are 44,672,304 and Looking at the lowest or least cases reported are ‘Holy See’. Looking at the trend where highest number cases reported daily wise where beginning of 2022 March it has reported peak. ‘United states’ has highest death cases so far with number of 1,104,743. ‘India’ is in second place where total death cases are 530,604 Looking at the lowest or least death cases are ‘Holy See’.‘Australia’ has highest recovered cases so far with number of 7,50,456. ‘Denmark’ is in second place where total death cases are 530,604.

Results and Discussions

The pandemic of Coronavirus Disease 2019 (COVID-19) is a timely reminder of the nature and impact of Public Health Emergencies of International Concern. As of 12 January 2022, there were over 314 million cases and over 5.5 million deaths notified since the start of the pandemic. The COVID-19 pandemic takes variable shapes and forms, in terms of cases and deaths, in different regions and countries of the world. The objective of this study is to analyse the variable expression of COVID-19 pandemic so that lessons can be learned towards an effective public health emergency response

We have found that regions and countries with high human development index have higher cases and deaths per million population due to COVID-19. This is due to international connectedness and mobility of their population related to trade and tourism, and their vulnerability related to older populations and higher rates of non-communicable diseases. We have also identified that the burden of the pandemic is also variable among high- and middle-income countries due to differences in the governance of the pandemic, fragmentation of health systems, and socio-economic inequities.