COVID-19 Tracking, Visualizing and Predicting the Life Expectancy of Virus

Nirbhay Mathur (✉ nirbhay.mathur@utp.edu.my)

Universiti Teknologi PETRONAS <https://orcid.org/0000-0002-0279-118X>

Vijanth S. Asirvadam

Universiti Teknologi PETRONAS

Naga Swetha Pasupulethi

Universiti Teknologi PETRONAS

Research Article

Keywords: COVID-19, Tracking App, Prediction, Bathtub Curve

Posted Date: June 19th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-35821/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

COVID-19 Tracking, Visualizing and Predicting the Life Expectancy of Virus

Nirbhay Mathur · Vijanth S.
Asirvadam · Naga Swetha Pasupulethi

Received: date / Accepted: date

Abstract An outbreak of the corona virus is hitting around the world. Since many research had stated the prediction of the corona virus based on the active stage. But somehow the results are not reliable or very close enough to get any results. This research main focus is on understanding the life expectancy behavior of the corona virus. This research also develops the COVID-19 tracking dashboard to visualize the trend of cases recorded, cases recovered, and the number of cases died. A Bathtub model is used to understand the survival behavior of the corona virus and based on the life expectancy, a prediction along with visualization will be carried out.

Keywords COVID-19 · Tracking App · Prediction · Bathtub Curve

1 Introduction

COVID-19 is a virus of the Coronavirus family and the source of a respiratory illness outbreak throughout the world that originated in Wuhan, China. Studies show that COVID-19 has clinical characteristics akin to the SARS-CoV. Coronaviruses belong to a large family of viruses which are usually found commonly in people and also in a certain different type of species of animals like camels, cats, and bats. Animal coronaviruses don't or very can rarely infect and spread among people. This is even with the new virus COVID -19 too.

The COVID -19 or SARS COV -2 virus is a beta coronavirus similar to MERS-CoV and SARS-CoV. These viruses have been found to have their origin from bats. The dominant symptoms include Fever and cough while gastrointestinal symptoms were uncommon. A few other commonly noticed symptoms include fatigue, muscle pain, diarrhea, sore throat, loss of odor, and sometimes even pain in and around the abdomen. The time from exposure to onset of symptoms is typically from two days to two weeks. While the majority of cases result in mild symptoms and the infected get cured, rarely some cases progress to severe pneumonia and multiple-organ failure or disorders. The mortality rate is around 2%, while seasonal flu typically has a mortality rate below 1% and causes about 400,000 deaths annually across the globe. In COVID-19 infected patients, the absence of fever is more frequent than in patients infected by similar viruses, i.e. MERS Corona Virus (2%) and SARS Corona Virus(1%) The initial patients infected by COVID-19, reportedly indicated an association with an animal market in Wuhan and has demonstrated an animal-to-person spread. Per contra, a burgeoning number of patients has not displayed any association with the animal markets revealing the fact of human-to-human spread of COVID-19. The virus is observed to primarily spread across people through small droplets that are produced when an infected person coughs, sneezes, or talks. While these droplets from an infected person that are produced while breathing out, fall onto the ground or on surfaces rather than remaining in the air due to their higher density. People can often get infected by touching a contaminated surface and then they can get infected by touching their eyes, nose, or mouth as it is said that the virus has a life span for up to 72 hours. The virus is most contagious during the first

three days after the onset of symptoms, but the key aspect of the virus is that it can spread even before symptoms appear and in later stages of the disease. This pandemic has been declared a global health emergency and is spreading at an alarming rate. Not only the confirmed cases are increasing but also the number of deaths across the world seems to increase exponentially. Now known as COVID-19, was first encountered in November 2019, and has now it is gone on to spread to close to 150 countries and as of early April 2020, around 2.4 Million cases have been confirmed affected by COVID-19. Out of which 1,65,229 infected have been reported dead and There's no vaccine present currently for this virus/disease. One of the major problems associated with this is complexity in the detection of the disease.

The proposed research paper uses R and R Shiny to predict and develop the application to track COVID -19 and Bathtub curve to predict and visualize and map the COVID – 19 trends and its life expectancy. R is a statistical computing and graphical environment. The software provides a wide variety of

statistical modeling, analysis, clustering, and graphical techniques. R is widely used as an integrated platform for data manipulation and calculation and graphical display, it includes effective data handling and storage facility too. hence, in this paper, the authors have used R and Shiny to develop a web-based dashboard for tracking coronavirus cases.

The paper also proposes the use of a bathtub curve to visualize and model the three key phases of coronavirus life. The bathtub curve is generally found to be used in reliability engineering to describe and predict the hazard function. Bathtub curve looks resemble a cross-sectional shape of the bathtub with steep sides and flat bottom it is generated by mapping the rate of early infant mortality failures. The bathtub curve shows how many might fail during any given phase and the first portion of the curve is known as the infant mortality phase which shows the quick fail, the second part of the curve is called normal lifetime segment representing a low failure rate and the third part is end-of-life increasing failure rate.

2 COVID-19 Tracker Web Application in R & Shiny

In the current era of development, computer technology has enhanced a lot in all sectors of research. An enhanced tools for fundamental concepts, visualization, predicting, statistical significance had a new trend [1–10]. A study has shown great achievement by utilizing this enhanced technology, to have good prediction and using statistical tools and analysis based on visualization [11]. From the past few years, web-based applications had taken special space in everyone's daily life. Infect, academic teaching had adapted we application platform to teaching students, hence there are many applets available for teaching [12]. Since, there are many advantages in using web based applet as they are dynamic in use, user friendly ui, easy to access through Internet, provide better understanding of topic with visual references.

Since a new trend is in the market, it is a situation where the instructor had to develop their own tools. Although developing this kind of app is not a very easy task. It requires some good knowledge of few computational languages i.e., Java, HTML, CSS, PHP and some other languages. To learn all these languages it is pretty time consuming to understand their hierarchy and create a web-based application. To overcome all these time-consuming languages and to understand their typical syntax, an alternative method has been developed by some researchers and developers.

R studio was developed in 2015, which is the recent technology tool which helps to develop and design much application which is web-based. This tool/software doesn't demand instructors to have good knowledge of other above-mentioned programming languages. A package was also developed by the R software team which is known as "Shiny". This package is specially designed for a web-based application. The major benefit of using this software is "open source".

2.1 Case Study: COVID-19 Tracker

This section will demonstrate the usage of "R software" and package "Shiny". There are over 4000 contributed packages available in R. These packages are as diverse as medical imaging, signal processing, statistical, simulation, parallel processing, visualization, machine learning [13].

The Shiny is a free contributed package, which is specially developed for creating interactive web based dashboards. These dashboard are dynamic in use and can handle big data to generate plots, graphs and even prediction. Installation and making your desktop ready from web based application is very easy. Some steps can be followed by either using search engine or by a book as cited [13].

Shiny application consist of two components:

- ui. R
- server .R

Combination of these two file make and shiny app which is able to get web based.

- What is ui.R !

User Interface (UI), is a source code setup where all written code will display output on the web app. This means the layout of the web page i.e, title, slider, plots, map, bins and many other widgets will be coded on the ui.R section.

- What is server.R !

This section does the all computational work i.e, all the algorithm processes, data handling, plotting options (legends, color, bins, dimension, type of plot), this section instructs the computer to build the app. So, these are two source file which works together to create R shiny web application. Before we start looking at code for ui.R and server.R it is important to look at other packages that are required to build COVID-19 tracker. As shown in Table 1, all

Table 1 Software and Packages library used in developing COVID-19 tracker

Software/R Packages Library	Version
R	3.1.2
RStudio	1.0.153
Shiny	1.4.0.2
shinydashboard	0.7.1
shinydashboardPlus	0.7.0
ECharts2Shiny	0.2.13
DT	0.12
leaflet	2.0.3
readxl	1.3.1

the required packages are mentioned along with their version used in developing COVID-19 tracking web-based applications in R shiny. These packages are important and play an individual role in running the app smoothly and attractively.

2.2 Creating Shiny App

The first step to start developing web application is to set path of both file i.e. ui.R and server.R in same folder where app is being deployed, as shown in Figure 1.

Fig. 1 First step is to set path for shiny app.

As shown in the below code, first it is required to call and define the library which is going to be used in developing of the web app.

```
#code developed by : Nirbhay Mathur
# http://shiny.rstudio.com
#
library(shiny)
library(shinydashboard)
library(shinydashboardPlus)
library(ECharts2Shiny)
library(DT)
library(leaflet)
library(readxl)
data<-read_excel("MyLatLong.xlsx")
```

Once, all the libraries are defined, the next step is to get data in the app. To embed the data in shiny can be done in many ways. Either data can be called from a local file(such as excel format, CSV format, text format) or can be retrieved from online sources which can be extracted from the server or from some data storage(i.e, Github, google sheets, etc). All these formats can be used as loaded in RStudio as data set. Since to call each type of data format there is a specif library, in this app we have used offline data which is assembled in an excel file. Hence *library readxl* is used here.

```
library(readxl)
data<-read_excel("MyLatLong.xlsx")
```

Data can be defined as shown in the above code. Where, "*MyLatLong.xlsx*" is an excel file, which is called from a local path. Now, once all the library's and data set is defined, now we define ui.R to make our web application visible online. Below code shows the code for ui.R:

```
ui <- #shinyUI(DT::dataTableOutput('Home'))
dashboardPage(
  dashboardHeader(title="COVID-2019"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Home", tabName="Home"),
      menuItem("Malaysia", tabName="my"),
      menuItem("Bath Tub Curve", tabName="BTC", badgeColor="green")
    )
  ),

  dashboardBody(
    tabItems(
      tabItem("Home",

        box(title="About Covid-2019",
          tags$p(),
          box(title="Symptoms of COVID 2019",
            span(img(src="fact-sheet-5.png", width=750))),
            valueBoxOutput("cases"),
            valueBoxOutput("recovered"),
            valueBoxOutput("death"),
            valueBoxOutput("update"),
            valueBoxOutput("develop")
          ),
          tabItem("my",
            fluidRow(
              valueBoxOutput("totalmy"),
              valueBoxOutput("totalrecovered"),
              valueBoxOutput("totaldeath")
            )
          ),

          box(title="MAP",leafletOutput("mymap")),
          #p(),
          #ActionButton("recalc", "New points" )),

        widgetUserBox(title="State of Malaysia",
          type=2, collapsible=TRUE,color="warning", width=3,
          tags$ul(
```

Fig. 2 Output of ui.R

```
tags$li("Kuala Lumpur , 166"),
tags$li("Selangor , 292"),
tags$li("Johor , 129 "),
tags$li("Penang , 50"),
tags$li("Kelantan Province , 61"),
tags$li("Perlis , 9"),
tags$li("Negeri Sembilan , 70"),
tags$li("Melaka , 22"),
tags$li("Sabah , 136"),
tags$li("Pahang , 37"),
tags$li("Terengganu , 27"),
tags$li("Sarawak , 68"),
tags$li("Kedah , 47"),
tags$li("Perak , 55")
)
)
)
)
)
)
```

Figure 2 shows the output of ui.R. But still this out will not be produced until unless server.R is also compiled along with ui.R. Now, lets see the server.R code as follows.


```

# Define server logic required to draw a histogram
server <- function(input, output, session) {
#Value Box
output$cases <- renderValueBox ({
valueBox (
"2,345,476", "Number of Confermed Cases",
icon = icon("flash", lib = "glyphicon"),
color = "navy"
)
})
output$recovered <- renderValueBox ({
valueBox (
"604,606", "Total Recovered ",
icon = icon("thumbs up", lib = "glyphicon"),
color = "red"
)
})
output$death <- renderValueBox({
valueBox (
"161,196", "Total Death ",
icon = icon("send", lib = "glyphicon"),
color = "yellow"
)
})
output$update <- renderValueBox({
valueBox (
"4/19/2020, 10:22 GMT", "Last update",
icon = icon("send", lib = "glyphicon"),
color = "yellow"
)
})
output$develop <- renderValueBox ({
valueBox (
"Developed by:", "This app is developed by Nirbhay Mathur",
icon = icon("send", lib = "glyphicon"),
color = "yellow"
)
})

output$totalmy <- renderValueBox({
valueBox (
"5,389", "Number of Confermed Casesin Malaysia",
icon = icon("flash", lib = "glyphicon"),
color = "navy"
)
})

```

```

output$totalrecovered <- renderValueBox({
  valueBox (
    "3,197", "Total Recovered in Malaysia",
    icon = icon("thumbs-up", lib = "glyphicon"),
    color = "red"
  )
})
output$totaldeath <- renderValueBox({
  valueBox (
    "89", "Total Death in Malaysia",
    icon = icon("send", lib = "glyphicon"),
    color = "yellow"
  )
})

#=====
# points <- eventReactive (input$recalc, {
#   cbind(lng=174.768, lat=-36.852)
#}, ignoreNULL = FALSE)

output$mymap <- renderLeaflet({
  leaflet() %>%
  setView (lng = 113, lat = 3.8, zoom = 5) %>%
  addTiles() %>%
  addCircles (data=data, lat=~Lat, lng=~Long, weight = 5,
  popup=~Name, radius= 10)

})

}

# Run the application
shinyApp (ui = ui, server = server)

```

As shown in Figure 3, the display is another tab of the same web app as shown in Figure 2. This is the main advantage to have a web-based application, as it provides the user to understand the things in a more easy way.

COVID-19 Tracker this web-based app is used to visualizing the number of cases recorded, recovered and death caused due to the coronavirus. This web-based application is divided into 2 major visualization tab. The first one is a home tab, where all the information is been displayed for COVID-19. As shown in Figure 2, a web page has an information box, that box is used to provide basic information, such as what is a coronavirus, how does it spread and other facts about coronavirus. Also, an image is been embedded to show symptoms

Fig. 3 Malaysia Dynamic Map integrated in web application.

of coronavirus. These web-based applications are not just limited to one case study. These web-based applications can be used for many applications based on the user area of utilization.

To track the latest update on coronavirus cases, data is been collected from many registered sources.

#Value Box

```
output$cases <- renderValueBox ({
  valueBox (
    "2,345,476", "Number of Confermed Cases",
    icon = icon("flash", lib = "glyphicon"),
    color = "navy"
  )
})
```

As shown in the above code, a value box is been used and defined in server.R. There are many widgets available to make the shiny app more responsive and dynamic. Since, in this app all the data is been updated manually, as the source of receiving data is not updated all time.

2.3 How to get online with Shiny app

The shiny app can run on both environments i.e, localhost and the webserver. When the app is being developed it is run on a local server and gets tested in the browser. Once the app is ready with all development and content, it can

Fig. 4 Bathtub curve and its 3 phases.

be uploaded to any server. Here Shiny also do provide free service for uploading web app on ShinyApps.io. These servers are hosted by RStudio since its free service security can be guaranteed. Also, RStudio is a free development environment so it gets linked with an existing account of Google or GitHub, whereas these accounts are free to make and get connected with a big community of developers and can get help.

Since many packages are required to develop a web app and to make it fancier, all these packages can be essays paired in the server and can work without any hassle. More information on ShinyApp.io can be found on this address (<https://www.shinyapps.io/>).

3 Introduction to Bathtub Curve

COVID-19 is a pandemic situation, where the entire world is suffering from it. This virus has taken many lives and almost affected much human life's in several ways. This virus even forced humans to stay underground, with practicing social distance. Some countries whose main source of income is tourism are suffering a lot, not only some countries but this situation had made the entire world to think about their GDP respectively. It will be true to say, if medical sciences develop a vaccination for COVID-19 it will take another one (1) year time lag to get available for everyone. Even many researchers are working hard to correlate all available parameters to link each other and predict some useful information for coronavirus. There is no concrete solution to follow, hence it is required to understand it from both aspects i.e, qualitative and quantitative.

Bathtub curve is a graphical representation of the lifetime or life expectancy of a product. Quantitative analysis for COVID-19 is required to understand its behavior. Hence, the author is proposing to have a look at the failure rate of coronavirus. The understanding is to map the failure rate or drop-down rate of coronavirus by looking at some parameters.

Fig. 5 Bathtub visualization of total cases and total recovered cases in Malaysia.

As shown in Figure 4, three stages can be linked with coronavirus. Phase I: Which is known as *Infant mortality period*, this is the early phase, in where either product (in our case: coronavirus) get failed due to fault in manufacturing, or we can relate it like coronavirus can get controlled in the same lab from where it gets accelerated. But due to failure of circumstances virus survived and moved to stage II. It can also be mapped from data that provide online. The hazard rate of the virus in December 2019 was not so high. Hence, the consequences were not known from this virus, whereas the safety measures were not practiced properly.

Phase II: This phase is named as *normal life* or *useful life*. In this stage when the product (i.e, coronavirus) get survived with random failure. That means the spread was continuous to happen with very few controlled cases which can be said as a random failure of coronavirus spread.

Phase III: Last stage named as *End of life* or *wear out phase*. IN this phase the hazard rate increase rapidly and the age progression start to decay. Here, the hazard rate is compared with several safety measures took to control the spread.

As shown in Figure 5, it shows the perfect bathtub curve at phase III. This can be illustrated as the life expectancy of coronavirus is towards the end of its life. But a very important factor which has to keep in mind is, this virus can reborn and will not be notified until the next 10 days (minimum).

So it is important to consider extending the number of days to see the trend. Figure 6 shows the small extended time with some predicted values. This plot shows the control behavior by stabilizing confirmed cases.

As shown in Figure 7, it visualizes more clear predicted value stating the stabilizing of cases. Since the medications are not available for this virus to pull down, the only major factor to stop its growth is mobility. To make this virus get clean from the country the only one possibility is to stop all kinds of mobility and gathering of people.

Fig. 6 Extended bathtub visualization with predicted cases.

Fig. 7 Mobility extended till July.

4 Conclusion

This research is conducted to understand the behavior of COVID-19 for Malaysia. A data was collected from the Ministry of health which was Internet release. In Malaysia coronavirus cases are clustered in a few clusters from where the

cases are suspected to have spread. This research gives a brief introduction to web-based application development in the R and Shiny environment.

The second main focus of this research is to visualize the life expectancy of virus. The major cluster is Tabligh (2,031), Bandar Baru Ibrahim Majid (183), Churuch (146), Sungai Lui (138), Case 26 (121), Selangor Mansion (107), Wedding (96), and other case are localized [14]. Since a pattern can be established by visualizing all the events. The pattern localized all spread based on mobility and having gathering. Since the bathtub curve was visualized to see the hazard rate and the life expectancy can be predicted. This research is conducted by assuming some assumptions to predict future data and able to visualize the mobility index. The number of cases (hazard rate) will be decreased and survival if the virus will decrease if mobility can be held for a few more days.

Declarations

Competing interests: The authors declare no competing interests.

References

1. Chance, Beth, and Allan Rossman. "Using simulation to teach and learn statistics." In *Proceedings of the Seventh International Conference on Teaching Statistics*, pp. 1-6. Voorburg, The Netherlands: International Statistical Institute, 2006.
2. Chance, Beth, Dani Ben-Zvi, Joan Garfield, and Elsa Medina. "The Role of Technology in Improving Student Learning of Statistics." (2007).
3. Garfield, Joan, and Dani Ben-Zvi. *Developing students' statistical reasoning: Connecting research and teaching practice*. Springer Science & Business Media, 2008.
4. Chance, B., J. Garfield, and B. delMas. "A model of classroom research in action: Developing simulation activities to improve students' statistical reasoning." 52nd Session of the International Statistical Institute, Helsinki, Finland (1999).
5. Garfield, Joan, and Dani Ben-Zvi. *Developing students' statistical reasoning: Connecting research and teaching practice*. Springer Science & Business Media, 2008.
6. Harraway, John A. "Learning statistics using motivational videos, real data and free software." *Technology Innovations in Statistics Education* 6, no. 1 (2012).
7. McDaniel, Scott N., and Lisa B. Green. "Using applets and video instruction to foster students' understanding of sampling variability." *Technology Innovations in Statistics Education* 6, no. 1 (2012).
8. Pratt, Dave, Neville Davies, and Doreen Connor. "The role of technology in teaching and learning statistics." In *Teaching statistics in school mathematics-challenges for teaching and teacher education*, pp. 97-107. Springer, Dordrecht, 2011.
9. Chance, B., J. Garfield, and B. delMas. "A model of classroom research in action: Developing simulation activities to improve students' statistical reasoning." 52nd Session of the International Statistical Institute, Helsinki, Finland (1999).
10. Potter, Gail, Jimmy Wong, Irvin Alcaraz, and Peter Chi. "Web application teaching tools for statistics using R and shiny." *Technology Innovations in Statistics Education* 9, no. 1 (2016).
11. Hagtvedt, Reidar, Gregory Todd Jones, and Kari Jones. "Pedagogical simulation of sampling distributions and the central limit theorem." *Teaching Statistics* 29, no. 3 (2007): 94-97.
12. Rossman, A., and B. Chance. "The Rossman." *Chance Applet Collection* (2004).
13. Beeley, Chris. *Web application development with R using Shiny*. Packt Publishing Ltd, 2016.
14. <https://newslab.malaysiakini.com/covid-19/en/clusters>.

Figures

Figure 1

First step is to set path for shiny app.

Figure 2

Output of ui.R

Figure 3

Malaysia Dynamic Map integrated in web application.

Figure 4

Bathtub curve and its 3 phases.

Figure 5

Bathtub visualization of total cases and total recovered cases in Malaysia.

Figure 6

Extended bathtub visualization with predicted cases.

Figure 7

Mobility extended till July.