

Final Report of Internship Program 2020

On

“VISUALIZING COVID19”

MEDTOUREASY, NEW DELHI



27th June 2020



ACKNOWLEDGMENTS

The internship opportunity that I had with MedTourEasy was a great change for learning and understanding the intricacies of the subject of Data Visualizations in Data Analytics; and also, for personal as well as professional development. I am very obliged for having a chance to interact with so many professionals who guided me throughout the internship project and made it a great learning curve for me.

Firstly, I express my deepest gratitude and special thanks to the Training Head of MedTourEasy, Mr. Ankit Hasija who gave me an opportunity to carry out my internship at their esteemed organization. Also, I express my thanks to him for making me understand the details of the Data Analytics profile and training me in the same so that I can carry out the project properly and with maximum client satisfaction and also for sparing his valuable time in spite of his busy schedule.

I would also like to thank the team of MedTourEasy and my colleagues who made the working environment productive and very conducive.

TABLE OF CONTENTS

Acknowledgments i

Abstract iii

Sr. No.	Topic	Page No.
1	Introduction	
	1.1 About the Company	6
	1.2 About the Project	7
	1.3 Objectives and Deliverables	9
2	Methodology	
	2.1 Flow of the Project	11
	2.2 Use Case Diagram	12
	2.3 Language and Platform Used	13
3	Implementation	
	3.1 Gathering Requirements and Defining Problem Statement	16
	3.2 Data Collection and Importing	16
	3.3 Designing Databases	17
	3.4 Data Cleaning	19
	3.5 Data Filtering	20
	3.6 Prototyping - Power BI	22
	3.7 Development of Dashboards	24
4	Sample Screenshots and Observations	
	4.1 COVID-19 Worldwide Tracker	29
	4.2 COVID-19 Testing Tracker	
	4.3 COVID-19 India Tracker	
6	Conclusion	
7	Future Scope	
8	References	

ABSTRACT

In December 2019, COVID-19 was first identified in the Wuhan region of China. The World Health Organization declared COVID-19 as a Public Health Emergency of International Concern (PHEIC) on 30th January 2020. By March 11, it categorized COVID-19 outbreak as a pandemic. Within a period of two to three months it spread to 210 countries across the world.

According to WHO reports, COVID-19 is a severe acute respiratory syndrome which is transmitted through respiratory droplets and contact routes. It can be contracted by direct contact with the infected person or indirect contact through touch of infected surfaces and materials. Due to the infectious nature of this disease, it has spread to 210 countries and territories around the world and infected (confirmed) cases have crossed the 7 million mark worldwide.

In India, the disease was first detected on 30 January 2020 in Kerala in a student who returned from Wuhan. The total (cumulative) number of confirmed infected people has crossed 2.5 lakh mark as on 11th June 2020. To tackle this disease, governments all over the world have taken major steps to combat it and to mitigate its effects. Most of the countries have followed policies of social distancing, travel controls, staggered and complete lockdown and intensive testing.

But these steps have brought the entire world to a standstill with many restrictions in all countries. The inability for people to go out for economic activities and other revenue generating actions is having a major financial impact on the world. Hence it is imperative for organizations to study and track the data of this pandemic.

Therefore, this project aims at collecting and analyzing wide variety of large data sets, create intuitive and interactive dashboards for representing COVID-19 in order to gain meaningful insights.

1.1 About the Company

MedTourEasy, a global healthcare company, provides you the informational resources needed to evaluate your global options. It helps you find the right healthcare solution based on specific health needs, affordable care while meeting the quality standards that you expect to have in healthcare.

MedTourEasy improves access to healthcare for people everywhere. It is an easy to use platform and service that helps patients to get medical second opinions and to schedule affordable, high-quality medical treatment abroad.

1.2 About the Project

The 2019-nCov pandemic began in Wuhan, China in December 2019 and has caused extreme havoc in nearly the entire world. COVID-19, commonly known as Coronavirus, is a novel highly contagious virus belonging to the Coronaviridae family that was suspected of being transmitted from animals to humans. This virus causes mild to serious respiratory disease and death.

This pandemic has engulfed 210 countries/regions in merely six months infecting 1,949,210 people and taking the death toll to 123,348. Presently, the highest cases of 2019-nCoV infections have been reported in US, Spain, Italy and UK. However, the cases are now abruptly rising in Brazil, Russia and India whereas China, the place of origin of the disease, is now receiving a very few cases.

It is currently a global healthcare crisis worldwide with many countries' health care systems falling apart. The only choice in this situation is to avoid the incidence of infection and plan our healthcare system for the possible outcomes. Additionally, ban on movement and lockdowns has caused a great impact on the growth and financial aspects of economies.

In that reference, it is extremely crucial to create visualizations which help firms to analyze this situation and to prepare themselves for the future. Additionally, MedTourEasy, being one of the globally upcoming tele-medicine company in global healthcare, it is important for the firm to understand the current situation of COVID19 so as to gain more insights on the intensity of the pandemic, the response of all countries and the impact it will have on their market. Also, depending on the results of the analysis, this may be used for increasing their market presence and capacity planning.

Hence, this project aims at collecting and analyzing large data sets to create intuitive and interactive dashboards for representing COVID-19 in order to gain meaningful insights. The project is majorly divided into 3 subsections, as below,

- *Analysis of the problem:* This is done to assess the gravity of COVID-19 worldwide and on India as well. It contains statistics and data representing the problem – total cases, deaths, active and recovered. Also, it contains many comparative statistics with respect to parameters like country, state, age, gender etc.

- *Analysis of the steps taken by the country:* This is done to analyze how various countries have reacted to this situation. It contains a thorough analysis on different steps taken by different countries to mitigate the impact of the pandemic on their country.
- A correlation between the above two sections to assess the impact of strategies taken by countries on the spread of COVID-19

Each of the above sub-section has been represented in the form of dashboards which are created using R language on RStudio IDE and RMarkdown package. These dashboards use a wide array of functions and packages in R to create intuitive and drillable dashboards, which can then be used by the firm to analyze the situation and draw conclusions about the same.

1.3 Objectives and Deliverables

This project focuses on creating easily understandable, interactive and dynamic dashboards by gathering data of COVID-19 from various sources like John Hopkins Data Repository, ECDC, covid19india.org etc. and using the coding language R and packages like readr, dplyr, ggplot, ggplot2, flex dashboard and other RShiny Packages to visualize these statistics which will enable the firm to analyze the situation and draw conclusions regarding the pandemic. The prototype for all the dashboards will be created using Power BI (primarily to create dynamic visualizations like world map, heat maps, forecasting, slicers etc.)

The project consists of 3 dashboards detailed as follows (3 deliverables):

- a. Analysis of the problem of COVID19: This dashboard focuses on analyzing the data regarding the problem of COVID19. It highlights the following points and displays them through various types of visualizations:
 - Country wise comparison of total cases – Total / Active / Recovered / Deaths with a thorough analysis of cases increasing date wise.
 - Country wise comparison of variation of cases over time. This will depict how the cases have increased over each day.
 - Comparison of top 10 countries with respect to increase in cases.
 - Country wise comparison of variation of cases with respect to age group, gender, health, ethnicity.
 - Analysis of most vulnerable areas Worldwide.
 - Analysis of impact of COVID-19 on GDP of a country and other factors
- b. Analysis of steps taken by countries: This dashboard focuses on analyzing various steps taken by the governments of many countries. It highlights the following points (World level):
 - Country wise comparison of number of tests done.
 - Country wise comparison of number of tests and positives obtained over time.
 - Country wise comparison of number of total and new tests done per million and thousand.
 - Country wise comparison of lock down measures in terms of time of implementation; duration; gravity of implementation and its impact of COVID spread and deaths.

- Comparison of testing policies and travel controls adopted by different countries.
 - Comparison of total tests done with respect to GDP per capita of a country.
- c. Analysis of COVID-19 situation in India: This dashboard focuses primarily on India's situation in the pandemic, which highlights the following points:
- Overall and state wise comparison of Total Cases, Active, Recovered and Deaths with a thorough analysis of cases increasing date wise.
 - Analysis of most vulnerable areas in India.
 - State wise comparison of variation of cases over time. This will depict how the cases have increased over each day.
 - Comparison of top 5 states with respect to increase in cases.
 - Comparison of top 10 districts of the 5 majorly affected states.
 - Comparison of cases with respect to age group, gender, diabetes prevalence, hospitalizations etc.
 - Overall comparison of Total tests conducted.
 - Comparison of total tests per million and thousand, test positivity rate and tests per confirmed cases with a thorough analysis of tests conducted date wise.
 - State wise comparison of number of tests done.
 - State wise comparison of number of total and new tests done per million and thousand and test positivity rate.

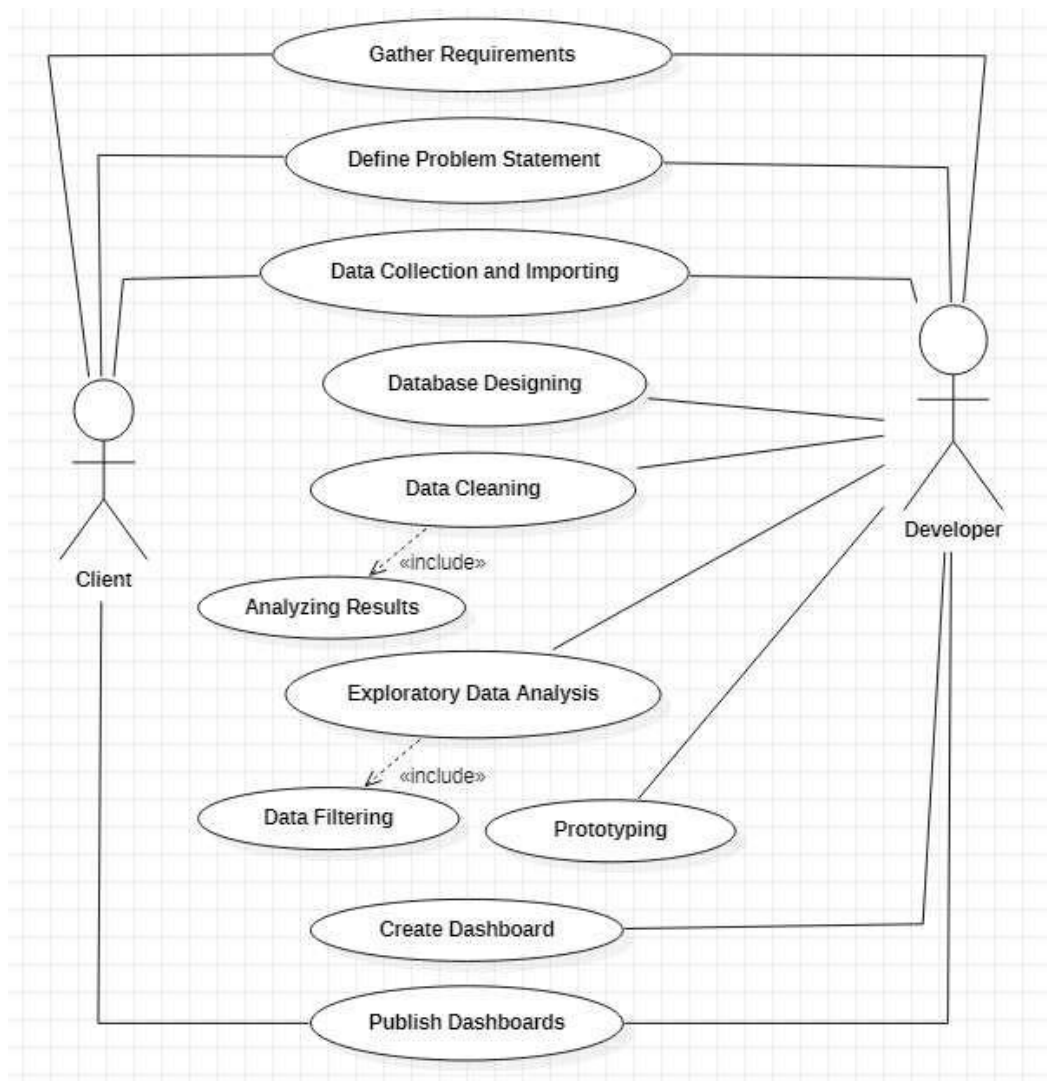
I. METHODOLOGY

2.1 Flow of the Project

The project followed the following steps to accomplish the desired objectives and deliverables. Each step has been explained in detail in the following section.



2.2 Use Case Diagram



Above figure shows the use case of the project. There are two main actors in the same: The Client and Developer. The developer will first gather requirements and define the problem statement then collecting the required data and importing it. Then the developer will design databases so as to identify various constraints and relations in the data. Next step is to clean the data to remove irregular values, blank values etc. Next, exploratory data analysis is conducted to filter the data according to the requirements of the project. Then a prototype of the dashboards is created using PowerBI to get a clear view of the visualizations to be developed. Finally, dashboard is developed and analyzed to publish the results to the client.

2.3 Language and Platform Used

2.3.1 Language: R

It is a programming language and software environment for statistical analysis, representation of graphics, and reports. R was developed in the University of Auckland, New Zealand by Ross Ihaka and Robert Gentleman, and is currently being developed by the R Technology Core Team. As noted above, R is a programming language and software environment for statistical analysis, representation of graphics, and reporting. The important features of R are:

- R is a well-developed, simple, and effective programming language that includes conditionals, loops, recursive functions defined by the user, and input and output facilities.
- R has efficient data processing and storage facilities.
- R includes a set of operators for arrays, lists, vectors, and matrix calculations.
- R offers a detailed, coherent and organized data analysis tool set.
- R provides graphical data analysis facilities and displays either directly on the computer or printing on papers.

2.3.2 IDE: RStudio

RStudio is an integrated development environment for R (IDE). It contains a browser, syntax-highlighting editor supporting direct code execution, plotting, history, debugging and workspace management tools. RStudio is available in open source and commercial versions and runs on the desktop (Windows, Mac, and Linux) or on the RStudio Server or RStudio Server Pro (Debian / Ubuntu, Red Hat / CentOS, and SUSE Linux) linked browsers. Major features are:

- RStudio runs on most desktops or on a server and accessed over the web.
- It integrates the tools you use with R into a single environment.
- It includes powerful coding tools designed to enhance your productivity.
- It enables rapid navigation to files and functions.
- It has integrated support for Git and Subversion.
- It supports authoring HTML, PDF, Word Documents, and slide shows.
- It supports interactive graphics with Shiny and ggvis.

2.3.3 Package: RMarkdown

R Markdown provides a data science authoring framework (.Rmd files). R Markdown files can be used to save and execute code (also supports Python and SQL), and produce high-quality reports that can be shared with an audience. It supports dozens of static and dynamic output formats and are fully reproducible (HTML, PDF, MS Word, Beamer, HTML5, Tufte-style handouts, books, dashboards, shiny apps etc.)

2.3.4 Template: Flexdashboard

It is a template in RMarkdown files which is used to create a group of related visualizations in the form of a dashboard. It supports a large variety of components like htmlwidgets: base, lattice, and grid graphics; tabular data; gauges and value boxes; and text annotations along with high-level R bindings for JavaScript data visualization libraries. Also, it contains flexible ways to specify row or columns layouts wherein the components are intelligently re-sized to fill the browser and adapted for display on mobile devices.

2.3.5 Dynamic element: RShiny

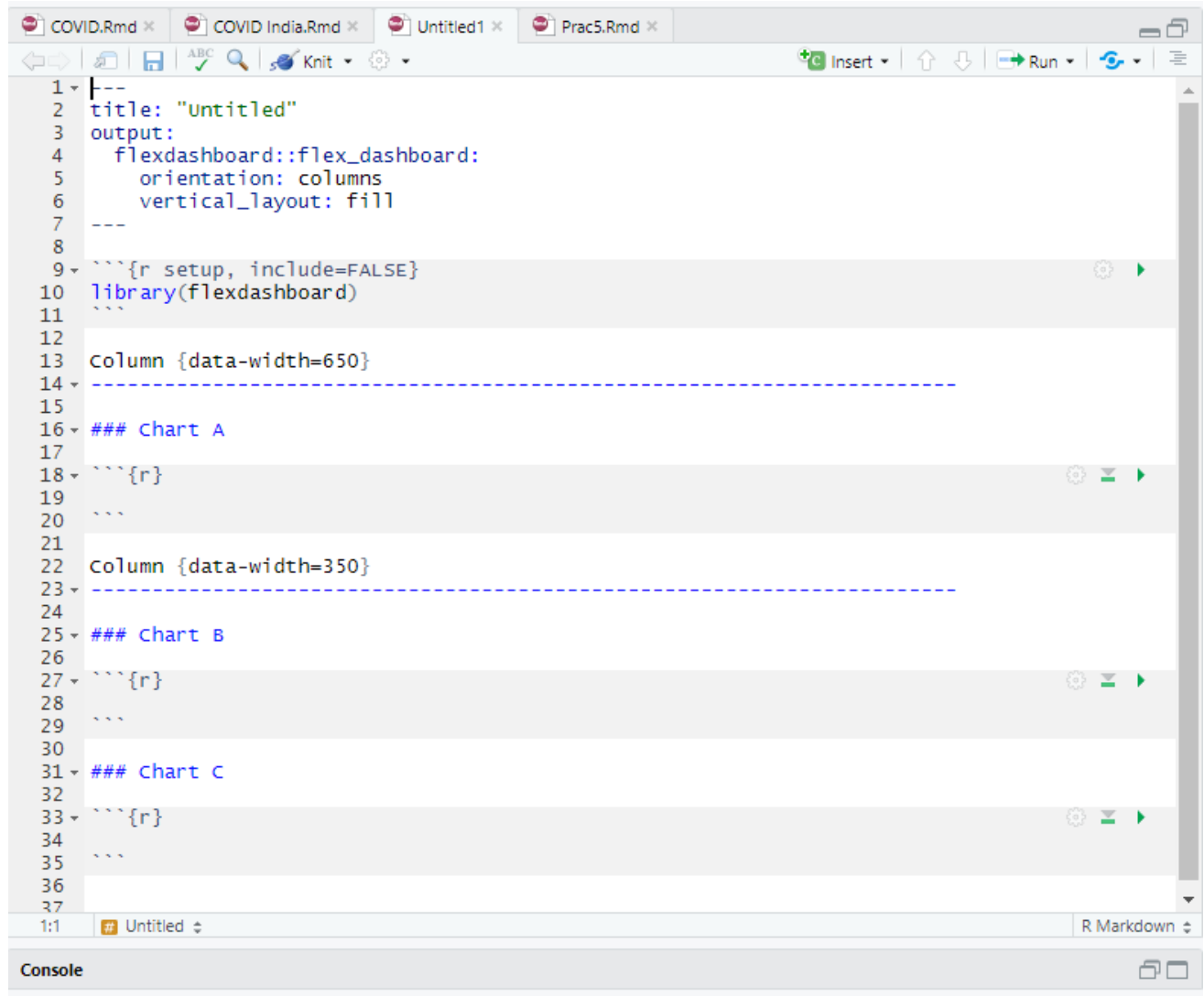
Shiny is an R-package that makes building interactive web apps straight from R, very easy. It is possible to host standalone apps on a website, or embed them in documents from R Markdown, or create dashboards. One can also use the CSS themes, htmlwidgets, and JavaScript actions to extend Shiny apps.

Installation:

```
install.packages("rmarkdown")  
install.packages("flexdashboard")  
  
runtime: shiny
```

Loading:

```
library(flexdashboard)  
  
library(shiny)
```



```

1  ---
2  title: "Untitled"
3  output:
4    flexdashboard::flex_dashboard:
5      orientation: columns
6      vertical_layout: fill
7  ---
8
9  ```{r setup, include=FALSE}
10 library(flexdashboard)
11 ```
12
13 Column {data-width=650}
14 -----
15
16 ### Chart A
17
18 ```{r}
19
20 ```
21
22 Column {data-width=350}
23 -----
24
25 ### Chart B
26
27 ```{r}
28
29 ```
30
31 ### Chart C
32
33 ```{r}
34
35 ```
36
37

```

1:1 | # Untitled | R Markdown

Console

This file contains the following parts:

- An (optional) YAML header surrounded by --- : The title, output format, orientation and layout of the file are defined.
- R code chunks surrounded by ```{r} : It is arranged in the form of rows and columns where the coding is to be done.
- Text mixed with simple text formatting.
- Knit button: To run the file and display output.

II. IMPLEMENTATION

3.1 Gathering Requirements and Defining Problem Statement

This is the first step wherein the requirements are collected from the clients to understand the deliverables and goals to be achieved after which a problem statement is defined which has to be adhered to while development of the project.

3.2 Data Collection and Importing

Data collection is a systematic approach for gathering and measuring information from a variety of sources in order to obtain a complete and accurate picture of an interest area. It helps an individual or organization to address specific questions, determine outcomes and forecast future probabilities and patterns.

The data COVID-19 has been collected through various GitHub repositories, mentioned as follows:

- Johns Hopkins University Center for Systems Science and Engineering
- European Centre for Disease Prevention and Control
- OurWorldData
- Covid19India.org

Data importing is referred to as uploading the required data into the coding environment from internal sources (computer) or external sources (online websites and data repositories). This data can then be manipulated, aggregated, filtered according to the requirements and needs of the project.

Packages Used:

Readr: The goal of readr is to provide a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. To accurately read a rectangular dataset with readr, one needs to combine two pieces: a function that parses the overall file, and a column specification.

Readxl: The readxl package is used to get data out of Excel and into R. Compared to many of the existing packages (e.g. gdata, xlsx, xlsReadWrite) readxl has no external dependencies, so it's easy to install and use on all operating systems. It is

designed to work with tabular data. readxl supports both the legacy .xls format and the modern xml-based .xlsx format.

Functions Used:

read.csv (): It is a wrapper function for read.table() that mandates a comma as separator and uses the input file's first line as header that specifies the table's column names. Thus, it is an ideal candidate to read CSV files. It has an additional parameter of url() which is used to pull live data directly from GitHub repository.

read_excel (): It calls excel_format() to determine if path is xls or xlsx, based on the file extension and the file itself, in that order.

Sample Code:

```
library(readxl)
library(readr)

x <- data.frame(read.csv(url(ECDC_ConfirmPath),stringsAsFactors = F))

vb <- data.frame(read_csv("D:/Internship/MedTourEasy - 6th April 2020/Project
- COVID/Input Data/JHU/05-28-2020.csv"))

world1 <- data.frame(readxl::read_excel("D:/Internship/MedTourEasy - 6th April
2020/Project - COVID/Input Data/ECDC/daily-cases-covid-
region.xlsx"))COVID/Input Data/JHU/05-28-2020.csv"))
```

3.3 Designing Databases

Once the data has been collected and imported into the R environment, it is important to design the structure of the database tables so as to identify the constraints in the data, keys, dependencies and relations between various tables.

Once the data is imported in the environment, it is converted into a data frame (data type in R) which makes it easy to maintain the data in form of tables. The various tables which have been created are mentioned as follows:

Attribute	Data type	Size	Extra
Country_ID	VARCHAR	5	Primary Key
Country_Name	CHAR	15	Not Null, Unique
Latitude	INT	3	Not Null
Longitude	INT	3	Not Null

Attribute	Data type	Size	Extra
Country_ID	VARCHAR	5	Foreign Key
Total Cases	INT	15	
Total Deaths	CHAR	10	
CFR	INT	15	
Median Age	VARCHAR	10	

Attribute	Data type	Size	Extra
Country_ID	VARCHAR	5	Foreign Key
GDP	INT	15	
Income Group	CHAR	10	
Total Cases	INT	15	
Total Deaths	INT	15	

Attribute	Data type	Size	Extra
Country_ID	VARCHAR	5	Foreign Key
Status	CHAR	10	Not Null
Dates	DATE	-	
Total Cases	INT	15	
New Cases	INT	15	
Total Cases / Mn	INT	15	
New Cases / Mn	INT	15	
Total Deaths	INT	15	
New Deaths	INT	15	
Total Deaths / Mn	INT	15	
New Deaths / Mn	INT	15	

**** Similar data frames are there for testing statistics and India statistics**

3.4 Data Cleaning

“Quality data beats fancy algorithms”

Data is the most imperative aspect of Analytics and Machine Learning. Everywhere in computing or business, data is required. But many a times, the data may be incomplete, inconsistent or may contain missing values when it comes to the real world. If the data is corrupted then the process may be impeded or inaccurate results may be provided. Hence, Data cleaning is considered a foundational element of the basic data science.

Data Cleaning means the process by which the incorrect, incomplete, inaccurate, irrelevant or missing part of the data is identified and then modified, replaced or deleted as needed.

With reference to the COVID-19 dataset, it may contain many null values or incorrect value simply because of inconsistency in reporting cases and testing statistics by countries and states. Hence various functions are used to clean this data.

Packages Used:

Tidyverse: It is a collection of essential data science R-packages. Under the tidyverse umbrella, the packages help perform and interact with the data. There are a whole host of things one can do with data, like sub setting, transforming, visualizing and so on.

Dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help solve the most common data manipulation challenges. It is simply the most useful package in R for data manipulation with the greatest advantage being the use the pipe function “%>%” to combine different functions in R. From filtering to grouping the data, this package does it all. It offers various functions like select, filter, group_by, summarize etc.

Functions Used:

Is.na(): In R, missing values are represented by the symbol **NA** (not available). Impossible values (e.g., dividing by zero) are represented by the symbol **NaN** (not a number). This function is used to check if a dataset contains NA values or not.

Na.rm(): When using a data frame function `na.rm` in `r` refers to the logical parameter that tells the function whether or not to remove NA values from the calculation. It literally means NA remove. It is a parameter used by several data frame functions.

Unique(): This function is used to filter out redundant data and keep only unique values from the data frame.

Na.omit(): This function returns the object with listwise deletion of missing values.

Mutate(): This function adds new variables that are functions of existing variables

As.date(): This function is used to convert between character representations and objects of class “Date” representing calendar dates.

Sample Code:

```
library(tidyverse)
library(dplyr)

cfr[is.na(cfr)] <- 0
cfr$Date <- anydate(cfr$Date)

xtot <- unique((x %>% filter(date >= as.Date('2020-02-15')))$location)

na.omit(s$Total.Confirmed)
```

3.5 Data Filtering

Data filtering is the method of choosing a smaller portion of the data set and using that subset to view, analyze and evaluate data. Generally, filtering is temporary – the entire data set is retained, but only part of it is used for calculation. It is also called subsetting or drill down data wherein data is extracted with respect to certain defined logical conditions. Filtering is used for the following tasks:

- Analyzing results for a particular period of time.
- Calculating results for particular groups of interest.
- Exclude erroneous or "bad" observations from an analysis.
- Train and validate statistical models.

With respect to COVID-19 dataset, the data needs to be filtered according to certain conditions like dates between December and July, top 10 COVID-19 hit countries,



gender, age groups etc.

Packages Used:

Tidyverse: It is a collection of essential data science R-packages. Under the tidyverse umbrella, the packages help perform and interact with the data. There are a whole host of things one can do with data, like sub setting, transforming, visualizing and so on.

Dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help solve the most common data manipulation challenges. It is simply the most useful package in R for data manipulation with the greatest advantage being the use the pipe function “%>%” to combine different functions in R. From filtering to grouping the data, this package does it all. It offers various functions like select, filter, group_by, summarize etc.

Functions Used:

Slice(): This function is used to extract rows by position.

Filter(): This function is used to extract rows that meet a certain logical criteria.

Logical Comparisons:

<: for less than

>: for greater than

<=: for less than or equal to

>=: for greater than or equal to

==: for equal to each other

!=: not equal to each other

%in%: group membership. For example, “value %in% c(2, 3)” means that value can takes 2 or 3.

Filter_all(), filter_at(): filter rows within a selection of variables. These functions replicate the logical criteria over all variables or a selection of variables.

Sample_n(): This function randomly select n rows

Top_n(): This function selects top n rows ordered by a variable

Sample Code:

```
library(tidyverse)
library(dplyr)

x1 <- x %>% filter(date >= as.Date('2020-02-15'))
x2 <- x %>% filter(date == as.Date('2020-06-07'))
x2 <- x2 %>% top_n(10, total_cases_per_million)
d <- x %>% filter(date == '2020-06-07')
```

3.6 Prototyping – Power BI

A prototype is an early version, model, or release of a product that is constructed to test a design or process. It is generally used by system analysts and users to assess a new design to enhance precision. Prototyping serves to specify a real, working system rather than a theoretical one. Creation of a prototype in some design workflow models is the step between formalizing and testing an idea.

Power BI is Microsoft's business analytics software. It aims to provide interactive visualizations and business intelligence capabilities with an interface that is easy enough to create your own reports and dashboards for end users. It provides cloud-based BI services, known as "Power BI Services," along with the "Power BI Desktop" desktop-based interface. It provides capabilities for data warehouse, including data planning, data discovery and interactive dashboards. It has the following features:

- Easy to connect, model, and visualize data, creating memorable reports personalized with KPIs and brand.
- Can generate fast, AI-powered answers to business questions
- Data is better secured across Power BI reports, dashboards, and data sets with persistent protection that keeps working even when shared outside the organization or exported to other formats such as Excel, PowerPoint, and PDF.
- Input: In the form of Excel, CSV, text, SQL and other formats
- Visualizations: Wide variety of graphs, infographics, KPIs, Filters, Slicers, etc.
- Output: Easily publishable reports and dashboards

The screenshots of the prototype are as follows:

COVID-19 WORLDWIDE TRACKER

location

Last Updated On: 05 May 2020

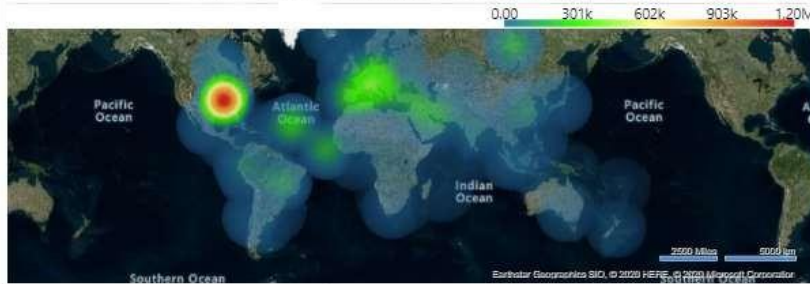
86309078
total_cases

Filter by Country

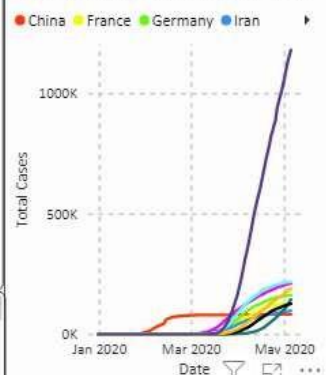
Search

- ☐ Select all
- ☐ Afghanistan
- ☐ Albania
- ☐ Algeria
- ☐ Andorra
- ☐ Angola
- ☐ Argentina

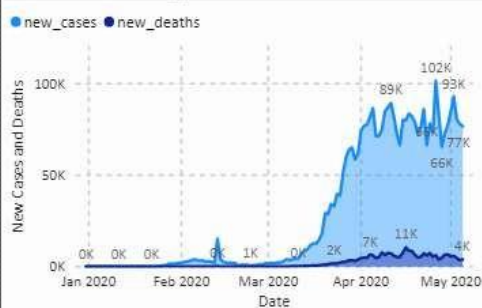
Country Wise Comparison of Total Confirmed Cases



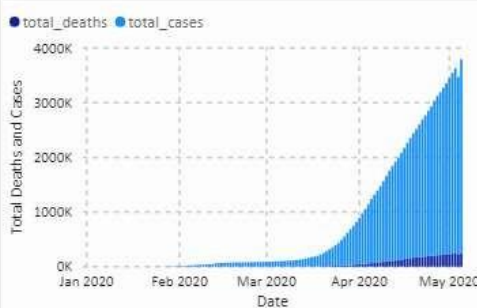
Top 10 Countries with COVID Cases



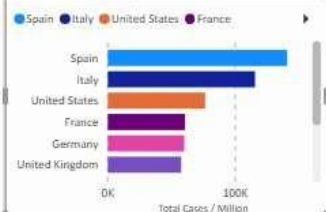
Date Wise Comparison of New Cases and Deaths



Date Wise Total Cases and Deaths Worldwide



Top Countries



United Kingdom 190584

Germany 163860

Russia 145268

France 131863

Turkey 127659

Brazil 107780

Iran 98647

COVID-19 WORLDWIDE TRACKER

location

Last Updated On: 05 May 2020

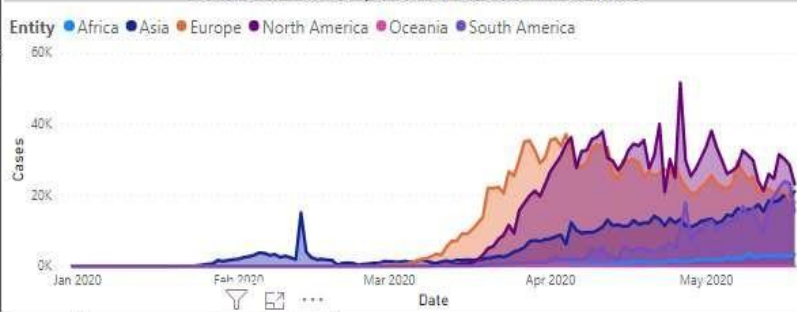
86309078
total_cases

Filter by Country

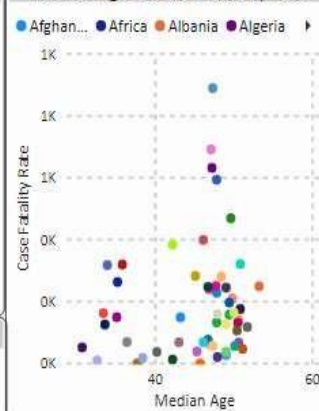
Search

- ☐ Select all
- ☐ Afghanistan
- ☐ Albania
- ☐ Algeria
- ☐ Andorra
- ☐ Angola
- ☐ Argentina

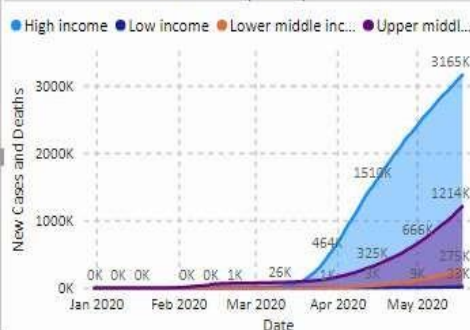
Continent Wise Comparison of Total Confirmed Cases



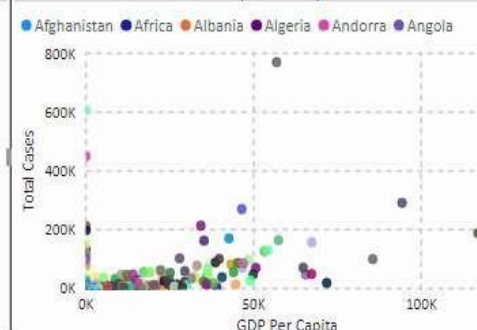
Median Age Vs. Case Fatality Rate



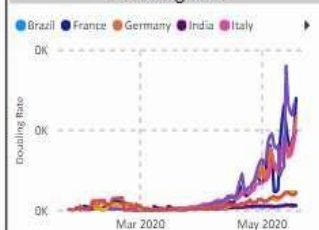
Income Group Comparison



GDP Per Capita Comparison



Doubling Rate



Zambia 137

Togo 126

Cambodia 122

Chad 117

Swaziland 116

Trinidad and Tobago 116

Bermuda 115

Aruba 10

3.7 Development of Dashboards

As stated earlier, the dashboards have been created using the flexdashboard package in R which provides a template in the form of rows or columns to display a group of related data through interactive visualizations from which conclusions can be drawn. The 3 dashboards to be developed are stated below:

- COVID-19 Worldwide Tracker
- COVID-19 Testing Tracker
- COVID-19 India Tracker

3.7.1 Defining Visuals

Data visualization is presenting data in a graphical or pictorial format. It allows decision-makers to see visually presented analytics, so that they can grasp difficult concepts or identify new patterns. In interactive visualizations, technology can be used to dig in charts and graphs for more detail, interactively modifying what data one can see and how it works.

Because of the way in which the human brain processes information, it is easier to visualize large amounts of complex data using charts or graphs than to poring over spreadsheets or reports. Data visualization is a quick, easy and universal way of conveying concepts. Data visualization can also:

- Identify areas that need attention or improvement.
- Clarify which factors influence customer behaviour.
- Help you understand which products to place where.
- Predict sales volumes.

In R, these visualizations are based on the Grammar of Graphics. It is a tool that enables one to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the ``scatterplot'') and gain insight into the deep structure that underlies statistical graphics. It contains the following layers:

1. Data: The data element is the data set itself. In this reference, the data is the COVID-19 datasets.
2. Aesthetics: The data has to be mapped onto the aesthetics element (variables

- mapped to x or y position and aesthetics attributes such as color, shape, or size)
3. Geometries: This element determines how the data is being displayed (bars, points, lines). It consist of `geom_line()`, `geom_scatter()`, `geom_bar()`, `geom_col()`, `geom_area()`, `geom_point()`, etc. Every single plot that is made will always consist of the above three layers.
 4. Facet: It is an optional layer. Facetting splits the data into subsets and displays the same graph for every subset.
 5. Statistics: It helps to transform the data (add mean, median, quartile)
 6. Coordinates: It helps to transforms axes (changes spacing of displayed data)

Packages Used:

Ggplot2: Ggplot2 is a declarative graphics development framework focused on The Grammar of Graphics. Once the user provides the data and tells ggplot2 how to map aesthetic variables and what graphic primitives to use, it takes care of the details. In most cases, one starts with `ggplot()`, supplies a dataset and aesthetic mapping (with `aes()`), the adds on layers (like `geom_point()` or `geom_histogram()`), scales, faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

Plotly: This is a complement to the ggplot package which includes javascript libraries to provide more interactive visuals.

Leaflet: Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. It makes it easy to integrate and control leaflet maps in R. Some of its features include:

- Interactive panning/zooming
- Compose maps using arbitrary combinations of Map tiles, Markers, Polygons, Lines, Popups, GeoJSON
- Create maps right from the R console or RStudio
- Embed maps in knitr/R Markdown documents and Shiny apps
- Easily render spatial objects from the `sp` or `sf` packages, or data frames with latitude/longitude columns

Sample Code:

```
leaflet(map) %>%
  addTiles() %>%
  setView(lng = 0, lat = 30, zoom = 1.5) %>%
  addPolygons(
    fillOpacity = 0.7,
    fillColor = ~ pal(Cases),
    color = "white",
    label = ~map_labels,
    highlight = highlightOptions(
      color = "black",
      bringToFront = TRUE
    )
  )

a <- ggplot(xf, aes(date, new_cases, fill = location)) + geom_area(alpha=0.6)
+ labs(x="Date", y = "New Cases")

a <- a + theme_minimal() + theme(legend.position="bottom",
                                legend.title = element_blank(),
                                legend.background = element_rect(fill =
"oldlace", size = 0.5, linetype = "dash", colour = "black"),
                                axis.text.x = element_text(face="bold",
color="black", size=10, angle = 30, vjust=.5),
                                axis.text.y = element_text(face="bold",
color="black", size=10),
                                axis.line = element_line(colour = "black",
size = 1.5, linetype = "dashed")) + scale_y_continuous(labels = scales::comma)

ggplotly(a)
```

3.7.2 Integration with RShiny

As discussed, Shiny is an R-package that makes building interactive web apps straight from R, very easy. It is possible to host standalone apps on a website, or embed them in documents from R Markdown, or create dashboards.

Using Shiny, the visualizations can be made more interactive and drillable. This means that the user, along with viewing the dashboards, can also give inputs to the dashboard which will then automatically update its visuals. By adding ‘runtime:shiny’ to the title block, the document can be embedded with shiny app.

In reference to the project, a combination of plotly+shiny is used for making the dashboard user interactive. Each shiny app consists of two major parts as follows:

1. The user interface, ui, describes how the web page displays inputs and output widgets. The fluidPage() function offers a nice and quick way to get a grid-

based responsive layout, and the UI is completely customizable and packages like shinydashboard make it easy to leverage more sophisticated layout frameworks.

2. The server function, server, defines a mapping between input values and output widgets. More precisely, the shiny server is a R function () between client input values and Web server outputs.

Shiny comes with a handful of other useful pre-packages input widgets. Although many shiny apps use them straight out of the box, CSS and/or SASS input widgets can be easily stylized, and even customized input widgets can be integrated.

Some of the shiny widgets include:

- selectInput()/selectizeInput() for dropdown menus.
- numericInput() for a single number.
- sliderInput() for a numeric range.
- textInput() for a character string.
- dateInput() for a single date and dateRangeInput() for a range of dates.
- checkboxInput()/checkboxGroupInput()/radioButtons() for choosing a list of options.

Sample Code:

```
ui <- fluidPage(
  selectizeInput(
    inputId = "worldregion",
    label = NULL,
    choices = c("Please choose a city" = "", worldregion),
    multiple = TRUE
  ),
  plotlyOutput(outputId = "p")
)

server <- function(input, output, session, ...) {
  output$p <- renderPlotly({
    req(input$worldregion)
    if (identical(input$worldregion, "")) return(NULL)
    a <- ggplot(data = filter(newerdata, Entity %in% input$worldregion)) +
      geom_area(aes(Date, Cases, fill = Entity), alpha=0.9) + labs(x="Date", y
= "New Cases") + theme_minimal()
    ggplotly(a, height=300, width = width) })
}

shinyApp(ui, server)
```

III. SAMPLE SCREENSHOTS AND OBSERVATIONS

4.1 COVID-19 Worldwide Tracker (Data as on 14th June 2020)

4.1.1 Heading and Quick Facts

This is the topmost bar of the dashboard which shows the heading along with two toggle buttons to represent the two pages of the dashboards – ‘World Situation’ and ‘Deep Dive’. 4 facts have been displayed as follows: Total Cases, Active Cases, Deaths and Recovered. All the data is directly obtained from the GitHub repository and is updated automatically.



4.1.2 World Hotspots

The first map of the dashboard is a heatmap of the world representing the total confirmed cases in the world represented by the leaflet widget. On the right, there is a legend which associates the case bins with the respective color. When a user hovers over a particular country, more information is displayed as shown below.

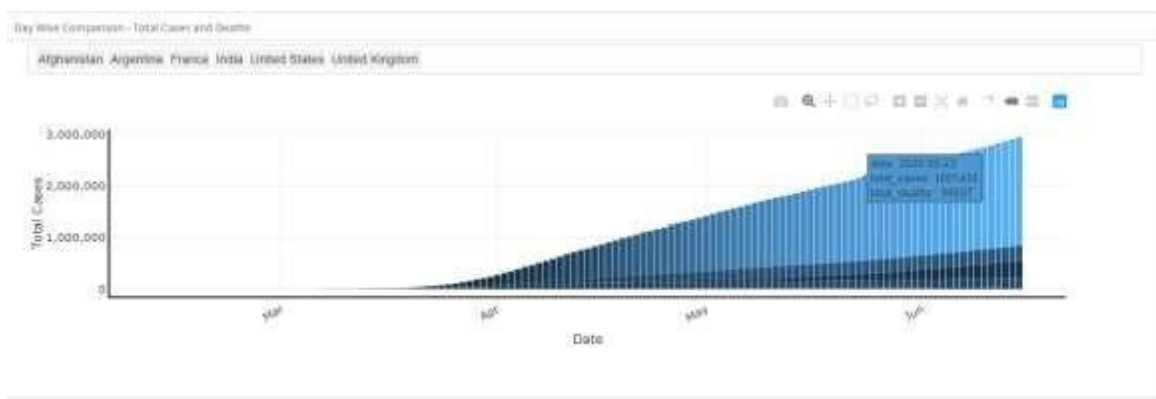
Observations: The heatmap shows that the countries with the darkest colors have been the worst affected. The top 5 worst hit countries include USA, Brazil, Russia, India and UK.



4.1.3 Day Wise Comparison - Total Cases and Deaths

The second visual is a bar chart representing the date wise increase in total cases and deaths in the world. R Shiny has been used to make the chart user interactive. The user can select any country and the chart will be updated according to it. Also, onHover event has been added.

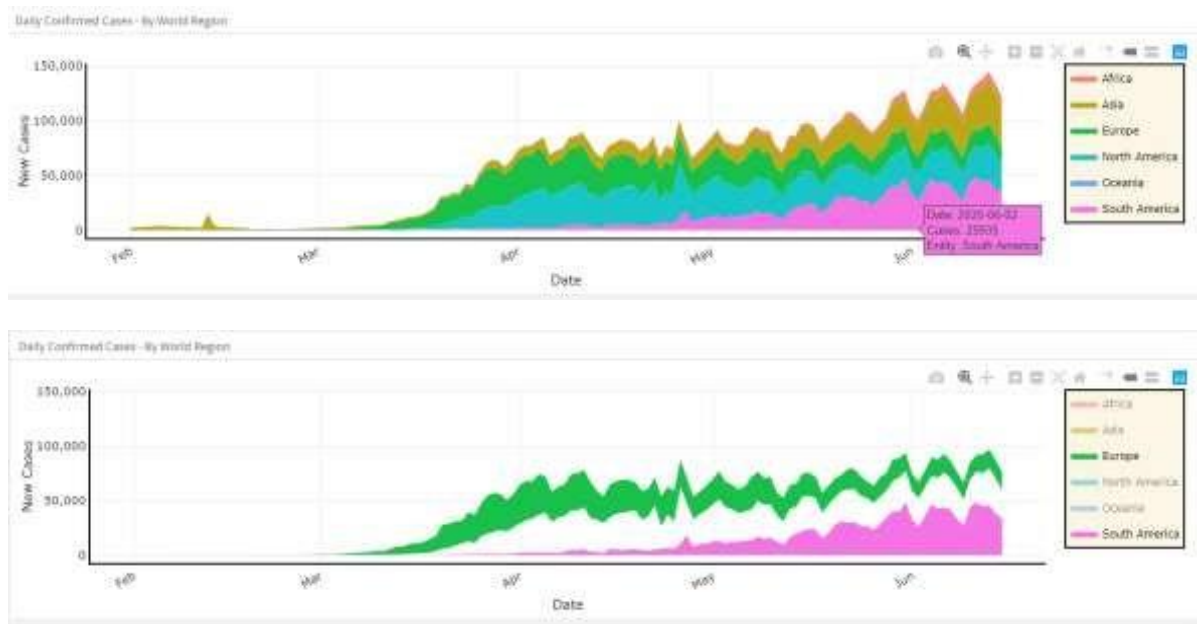
Observations: This visual indicates that there has been a linear increase in the total number of cases and deaths worldwide with the total number of cases crossing 7.9 million and deaths crossing 0.4 million as on 14th June 2020.



4.1.4 Daily Confirmed Cases - By World Region

This visual represents the new cases emerging by date with respect to 6 world regions. OnHover event has been added. Also, when a user double clicks on a world region, that particular region is isolated on the graph.

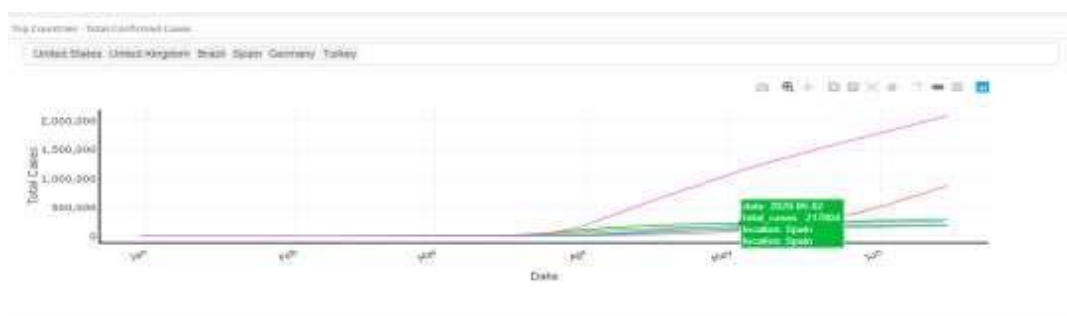
Observations: This visual represents the variation of new cases in the major world regions. In the initial months of February and March, the continents of Asia (China) and Europe (Spain, Italy, UK) were faced an increasing trend in the number of new cases everyday (indicated by larger area) whereas other continents were still in the early stages of the pandemic. Towards the month of April, South America (Brazil) and North America (USA) faced a peak in cases due to the countries reaching the community stage of transmission. With regards to the last 2 months, there has been a major peak in Asia again due to increasing number of cases in India and Russia, SA and NA continue to have greater number of cases and European countries have obtained a diminishing number of new cases per day.



4.1.5 Top Countries - Total Confirmed Cases

This visual represents a line chart of the date wise increase in total cases by top countries. R Shiny has been used to make the chart user interactive. The user can select any country and the chart will be updated according to it. Also, onHover event has been added.

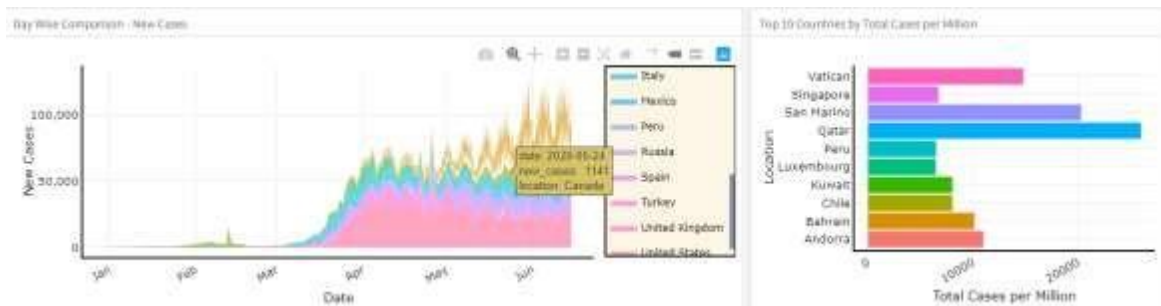
Observations: This visual represents the variation of total cases by country. As stated previously, China was the first country to face a peak in the total number of cases which was soon surpassed by European nations like Spain, Italy and UK. Soon after, due to less restrictions in movement, number of cases in USA increased exponentially. In the second half of the pandemic, countries like India, Russia and Brazil (which were able to delay the peak of total cases due to government measures like lockdown and social distancing) are now facing a great rise in the cases surpassing all European nations also.



4.1.6 Day Wise Comparison - New Cases and Top 10 Countries by Total Cases per Million

The first visual represents an area chart of the date wise increase in new cases by countries and the second one represents a bar chart of the top 10 countries by total cases per million. OnHover events are there on both and by selecting a country from the legend, the statistics for that can be isolated.

Observations: The first visual represents the same observations as stated before. The second visual represents the top 10 countries by total cases per million. It is observed that if measured by the total cases per million, the top 10 countries are entirely different to when measured by the total cases. This is because of the fact that the smaller nations have a relatively smaller population which makes the total cases per million higher indicating the fact that in countries like Qatar, Vatican, Chile etc., it is easier to contain the spread of the virus due to lesser population and restrictions of lockdown and social distancing can easily be implemented and followed. This is the reason why these countries are not in the top 10 worst affected list.

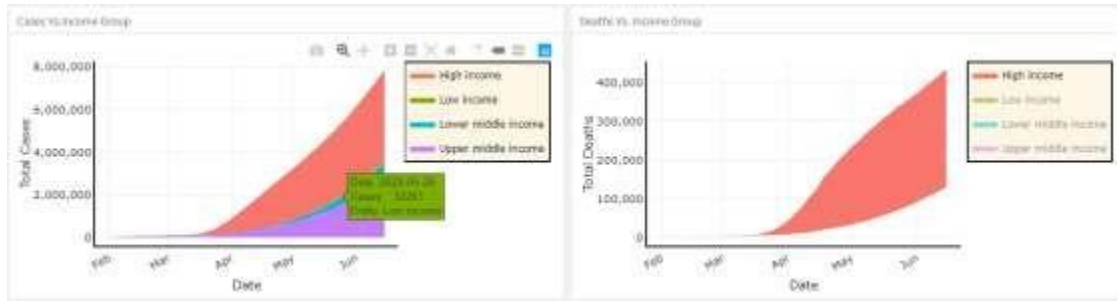


4.1.7 Cases and Deaths Vs. Income Group

These visuals represent a stacked area chart of date wise increase total cases and deaths by income group. OnHover event and legend selection is also included.

Observations: The following graphs indicate the variation of cases by the income group. As it is clearly highlighted, the high-income group has been the worst affected, followed by the upper middle income, lower middle income and finally the low-income group. Similar trend is highlighted in the variation of deaths as well. The primary reason for the same is that initially, the virus spread to major countries because of infected people travelling abroad and the major proportion of groups who can afford air travel is from the high and upper middle-income groups.

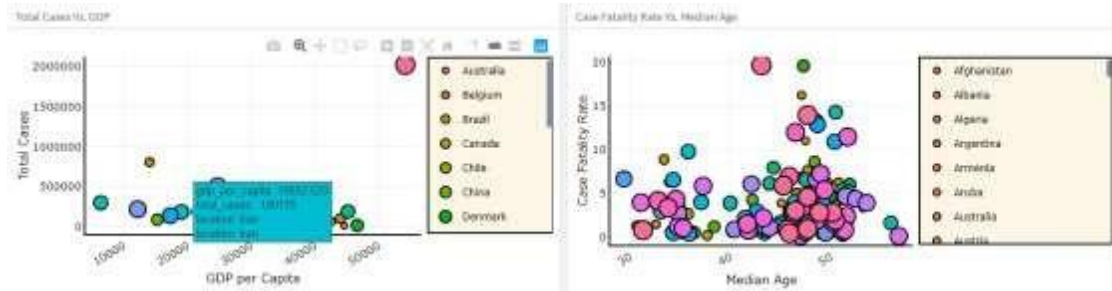
justifying the trends of the graph.



4.1.8 Total Cases Vs. GDP and Case Fatality Rate Vs. Median Age

The first visual represents a point chart of the GDP per Capita by Total cases and Country whereas the second one shows median age by case fatality rate and country. OnHover event and legend selection is also included.

Observations: These visuals indicate the countries with a greater GDP per Capita have greater capacities to invest in healthcare facilities of their country. Also, the countries which have a median age between 40 and 50 are facing greater number of COVID cases.

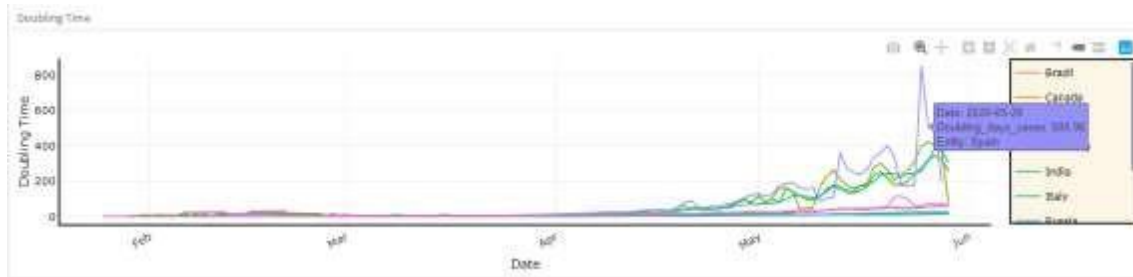


4.1.9 Doubling Time

This visual represents a line chart of date wise variation in doubling rate by country. OnHover event and legend selection is also included.

Observations: The data of this graph is as on 29th May 2020 after which the reporting for doubling time has been inconsistent. This graph indicates that the countries which faced a greater peak in total number of cases in the first stage of the pandemic have been able to tackle the spread of virus. Countries like UK, Germany, South Korea were able to do it ramping up their testing facilities whereas Spain, Italy and France by imposing restrictions like lockdown, due to which the doubling

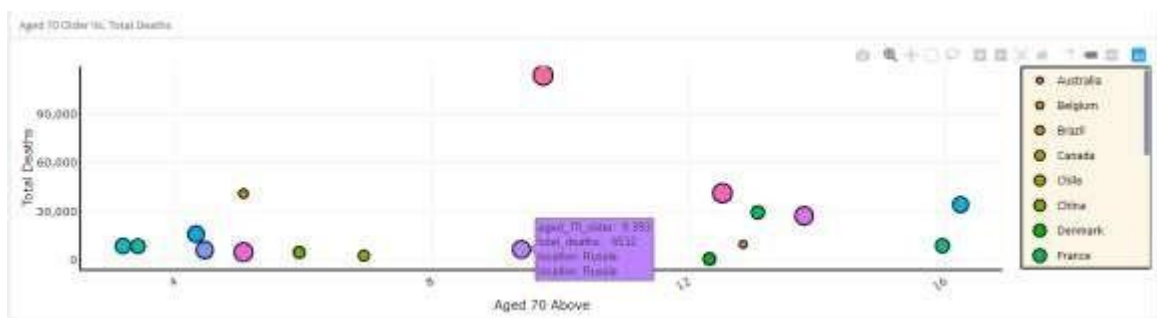
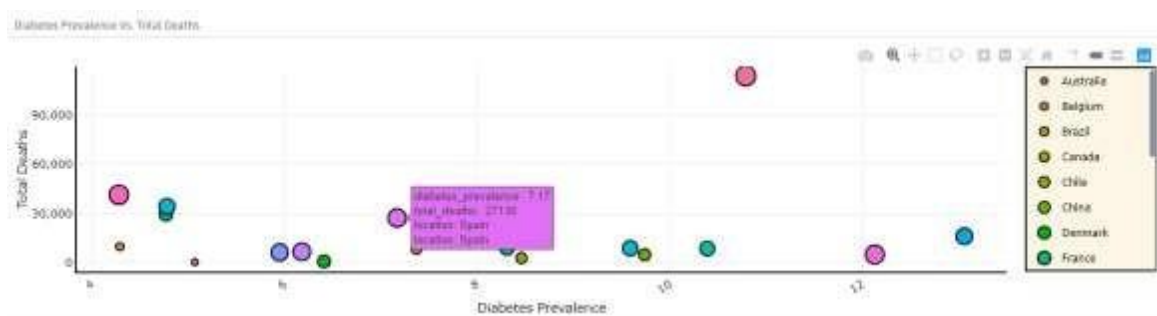
rate has increased to as high as 800 days. On the other hand, countries like Brazil, India, Russia, Mexico etc. haven't been able to increase the doubling rate due to an exponential increase in the cases which can be regarded to lesser amount of testing / lesser contact tracing / relaxation of restrictions.

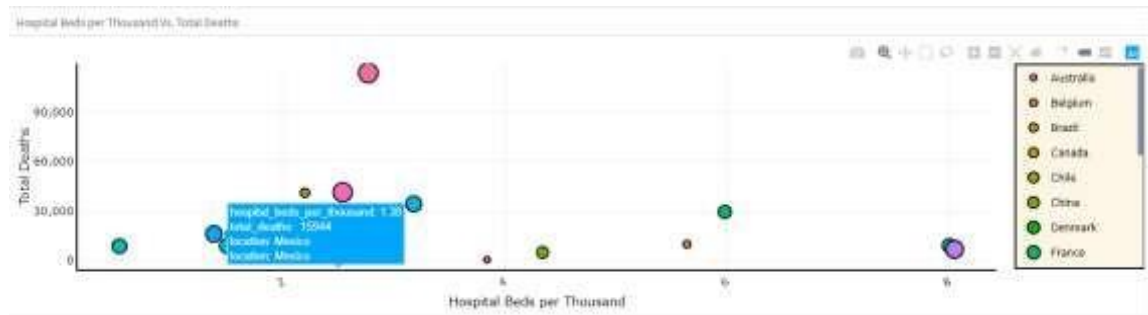


4.1.10 Diabetes Prevalence, Aged 70 Older and Hospital Beds per Thousand Vs. Total Deaths

The following visuals show the variation of total deaths by parameters like diabetes prevalence, aged 70 and hospital beds by country. OnHover event and legend selection is also included.

Observations: These visuals depict a correlation between various factors which affect the increase in total cases and deaths like diabetes prevalence, age and presence of hospital beds.





4.2 COVID-19 Testing Tracker (Data as on 14th June 2020)

4.2.1 Heading and Quick Facts

This is the topmost bar of the dashboard which shows the heading along with two toggle buttons to represent the two pages of the dashboards – ‘Worldwide Situation’ and ‘Deep Dive’. 2 facts have been displayed as follows: Total Tests and Total Tests per Thousand. All the data is directly obtained from the GitHub repository and is updated automatically.

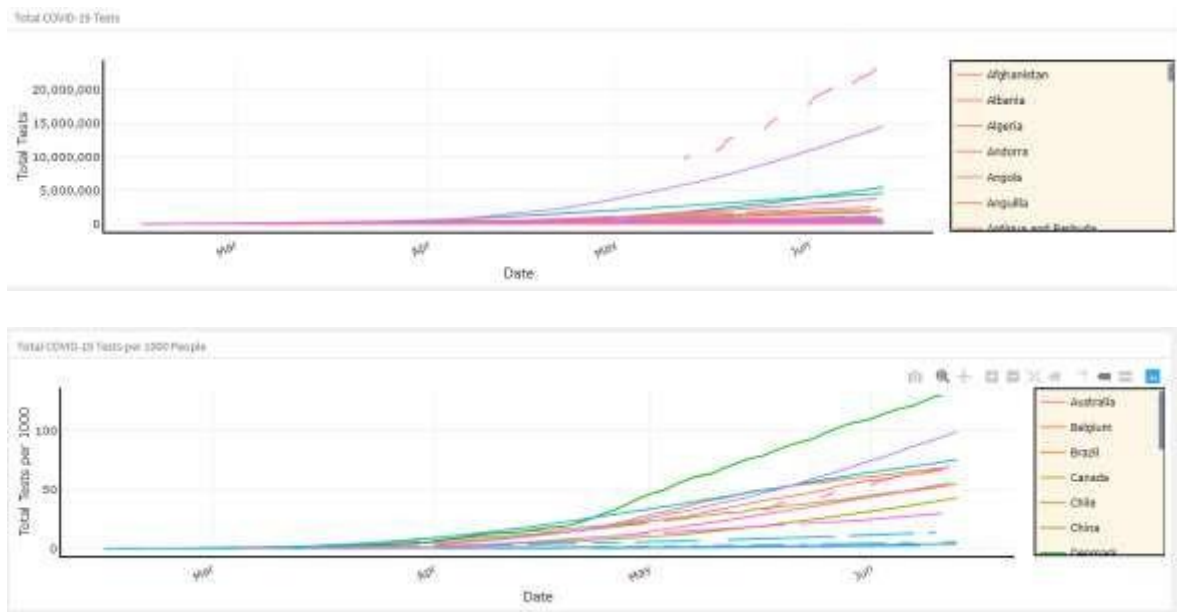


4.2.2 Total COVID-19 Tests and Tests per 1000 People

These visuals are line charts representing the date wise increase in total tests and total tests per 1000 people by country. OnHover and legend selection event has been added.

Observations: The first visual indicates how countries have ramped up their testing facilities to contain the spread of the virus. Although, US has not reported the testing conditions properly which is the reason for the broken line. In the first half of the pandemic, countries like South Korea, Iran, Singapore, New Zealand etc. increased their testing facilities rapidly which helped them to keep their cases in check, to the extent that New Zealand declared itself COVID free recently. On the other hand, insufficient testing in countries like Turkey, Italy, Spain, France etc. led to a relatively greater number of cases. In the second half, India and Russia have increased their testing and contact tracing facilities due to sudden increase in daily new cases.

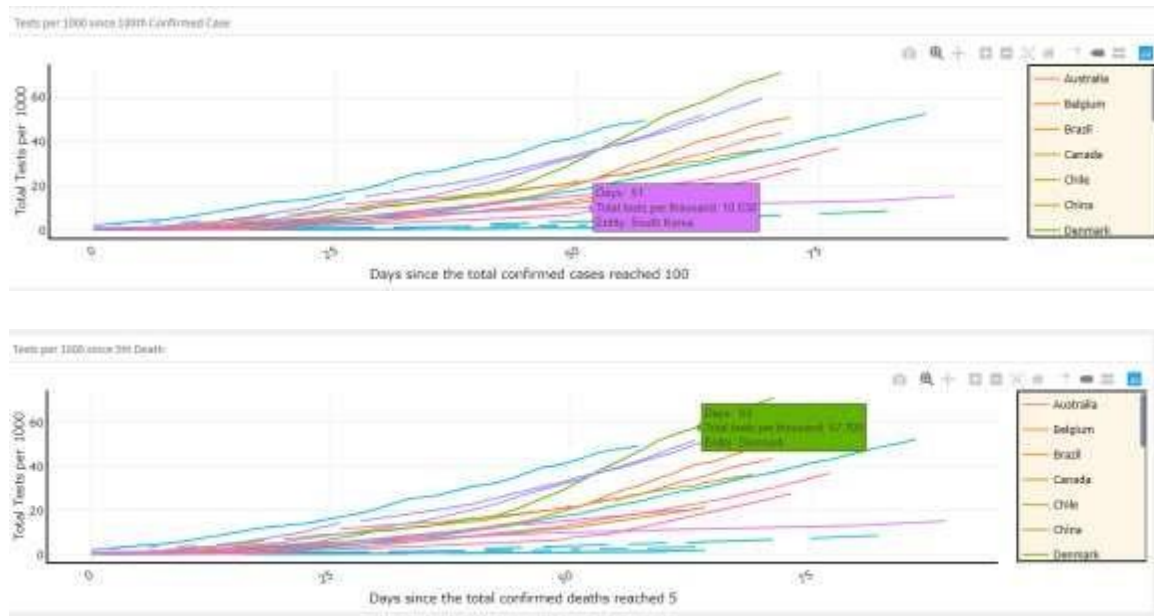
The second visual indicates that smaller countries like Denmark, Qatar, Australia etc. have a greater test per 1000 people figure when compared to India, USA, Russia even though the actual testing number of the latter countries is higher. This is regarded to the fact that the former countries have a lesser population and hence greater testing per 1000 people. These countries have been successfully able to contain the spread of the virus.



4.2.3 Tests per 1000 since 100th Confirmed Case and 5th Death

These visuals are line charts representing the total tests per 1000 people after the 100th case and 5th death were detected by country. OnHover and legend selection event has been added.

Observations: These graphs indicate how various countries have reacted to the pandemic situation after reporting their 100th case and 5th death. It indicates very contrasting situations throughout all the countries. As evident from the graph, countries like Denmark, Qatar, New Zealand, Belgium, Australia etc. ramped up their tests per 1000 exponentially just as soon as they reported the 100th case or the 5th death due to which they were able to keep the total number of cases in check. On the other hand, countries like India, Peru and Mexico did not ramp up the testing as required and hence the total number of cases are on the rise.



4.2.4 COVID-19 Testing Policies and International Travel Controls

These maps of the dashboard represent a heatmap of the world by COVID-19 testing policies followed and International travel controls imposed by various countries. It is represented by the leaflet widget. On the right, there is a legend which associates the bins with the respective color. When a user hovers over a particular country, more information is displayed as shown below.

Observations: Countries which followed a more open testing policy have been able to detect more cases and contain the spread. Also, major countries have imposed complete ban on international travel within countries.

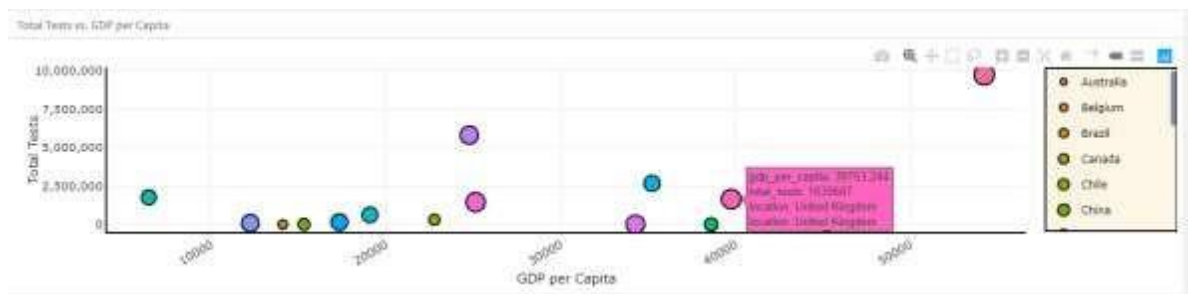




4.2.5 Total tests vs. GDP per Capita

This visual represents a point chart of the GDP per Capita by Total tests and Country. OnHover event and legend selection is also included.

Observations: This visual represents the variation of GDP per Capita by total tests and countries. It is very evident that countries having a greater GDP per Capita are able to invest more in their healthcare and testing facilities in contrast to those which have a lower GDP per Capita.

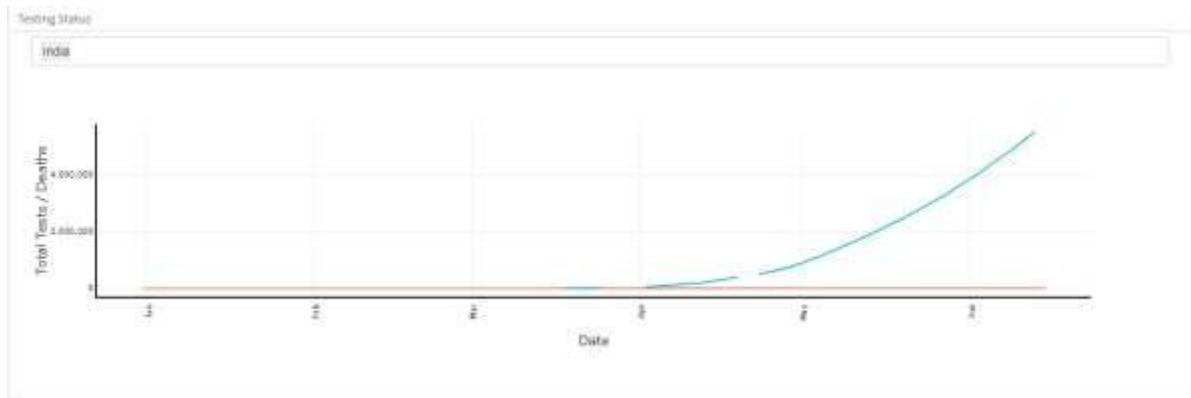


4.2.6 Testing Status

This visual represents a line chart of the date wise increase in total deaths and tests by each top country. R Shiny has been used to make the chart user interactive. The user can select any country and the chart will be updated according to it. Also, onHover event has been added.

Observations: This visual represents the date wise variation of total tests and deaths by country. It indicates that the countries which were able to increase the tests conducted by day, have been able to keep a check on the total deaths occurred due to the fact that with more testing, cases can be detected faster and the government

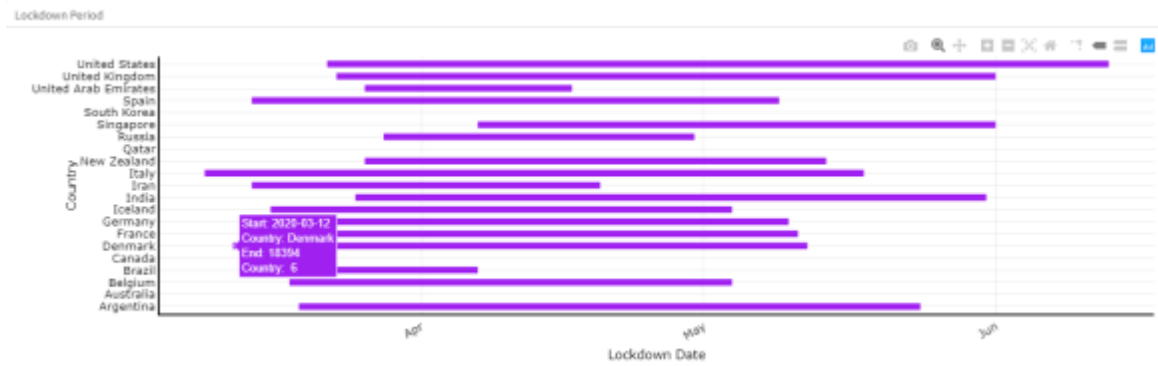
can keep a check on the spread of the virus.



4.2.7 Lockdown Period

This visual represents a gantt chart of the start and end dates of the majorly affected countries in the world. OnHover event has been added.

Observations: This graph represents a gantt chart of the dates of start and end of lockdown of major countries. In case of USA, the lockdown was introduced at a very later date which resulted in a magnanimous growth in number of cases. On the other hand, New Zealand, Denmark, Belgium and Australia started the lockdown just as soon as it reported the first case due to which the spread was contained. Also, India started the lockdown when its 500th case was reported which helped it to delay the peak cases for about one month. Additionally, South Korea and Singapore never went into a complete lockdown but due to their excellent testing facilities, they were able to keep the numbers in check.



4.3 COVID-19 Testing Tracker (Data as on 14th June 2020)

4.3.1 Heading and Quick Facts

This is the topmost bar of the dashboard which shows the heading along with two toggle buttons to represent the two pages of the dashboards – ‘Case Situation’ and ‘Testing Situation’. 4 facts have been displayed as follows: Total Cases, Deaths, Recovered and Tests. All the data is directly obtained from the GitHub repository and is updated automatically.

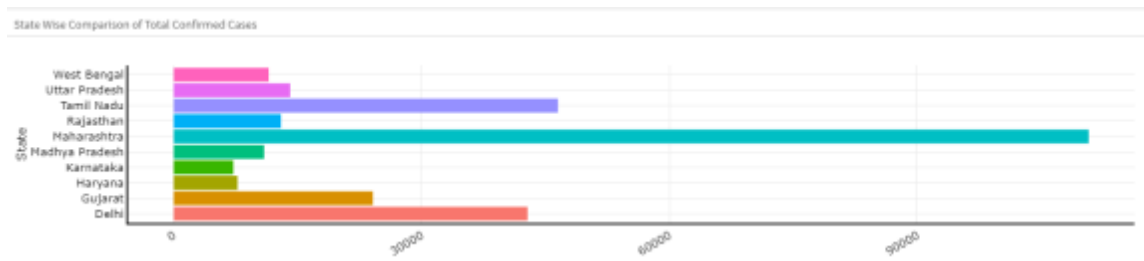


4.3.2 Confirmed Cases, Deaths and Recovered

These maps of the dashboard represent a heatmap of India by total cases, deaths and recovered. It is represented by the leaflet widget. On the right, there is a legend which associates the bins with the respective color. When a user hovers over a particular country, more information is displayed as shown below.

Observations: As on 14th June 2020, total number of cases reported in India has crossed the 3-lakh mark although the number of recoveries indicate a positive picture wherein the recoveries are almost half that of the cases, pushing the recovery rate to a 50% high mark. India has also managed to keep the number of deaths in check by increased hospitalizations. The top 10 affected states are: Maharashtra, Tamil Nadu, Delhi, Gujarat, UP, Rajasthan, West Bengal, Madhya Pradesh, Haryana and Karnataka.

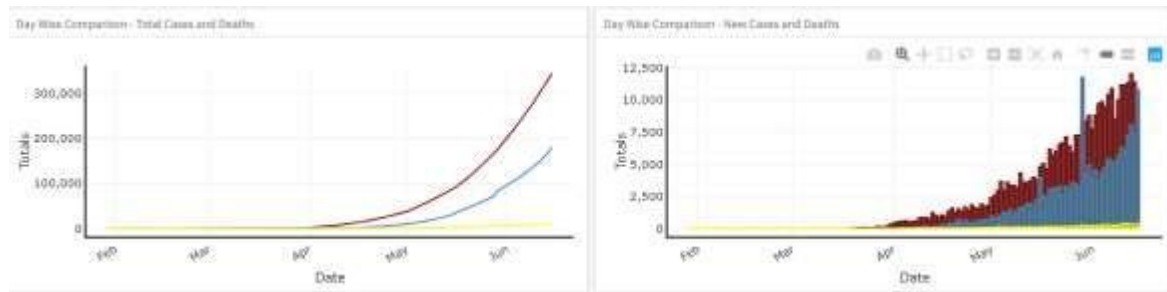




4.3.3 Day Wise Comparison – Total and New Cases and Deaths

These visuals are line and column charts respectively representing the date wise increase in total and new cases and deaths in the world. Also, onHover event has been added.

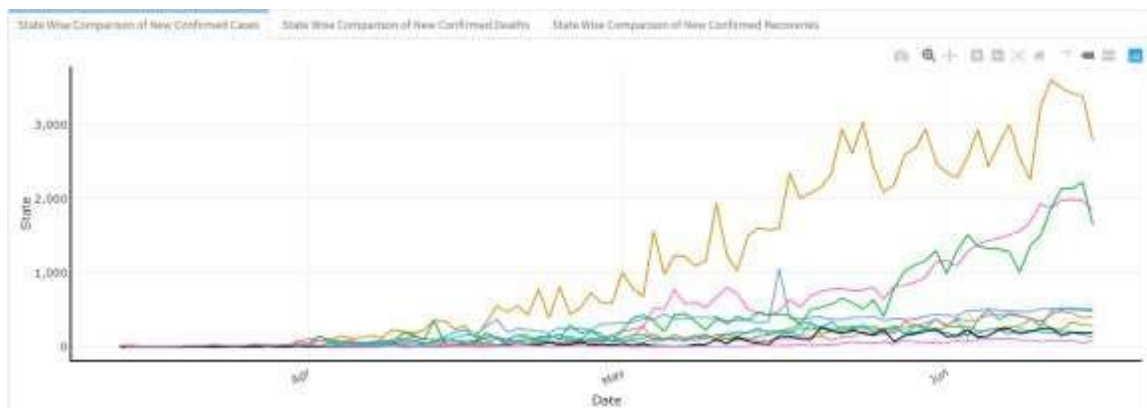
Observations: These visuals indicate overall increase in total cases with time. The total recovered cases have shown an increasing trend representing that India has been able to provide adequate healthcare facilities to the infected patients. This is also the reason why the total deaths have remained low when compared to other countries.

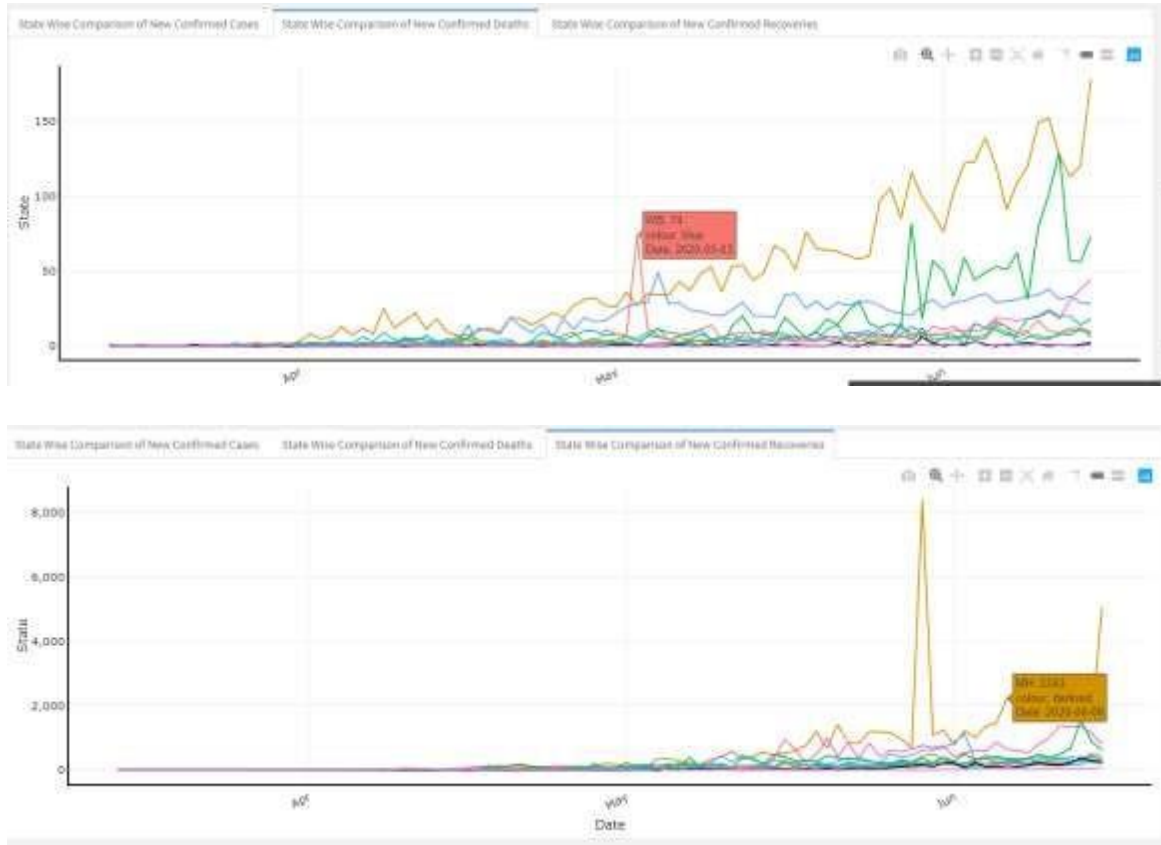


4.3.4 State Wise Comparison of New Confirmed Cases, Deaths and Recoveries

These visuals represent line charts of date wise variation of new cases, deaths and recoveries by states of India. OnHover event is included.

Observations: These graphs show a drill down, state wise analysis of variation of cases over time. The first case in India was reported in the state of Kerala on 30th January. In the months of February and March, the spread of the virus was fairly slower. India reported its 500th case on 25th March, when the country went into a complete lockdown. Then after, the cases started rising majorly in states like Maharashtra, Madhya Pradesh, Gujarat, Rajasthan and Delhi. As on 14th June, Maharashtra has emerged to become the major hotspot in India (crossing 1 lakh mark). On a positive note, the recoveries per state are also on a rise due to improved healthcare facilities provided to its citizens.

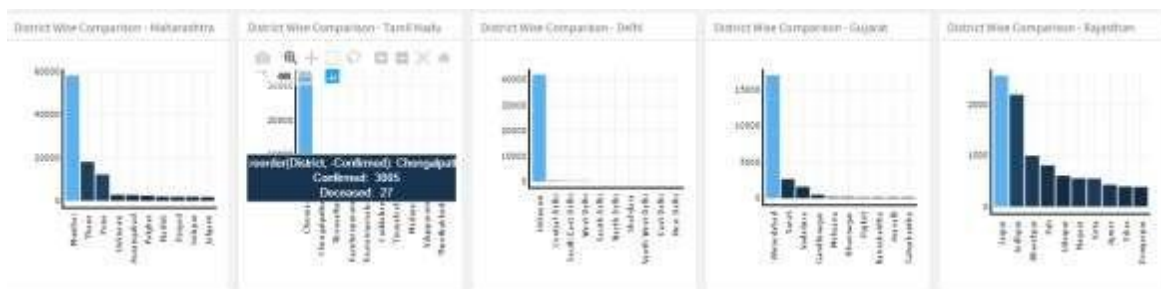




4.3.5 District Wise Comparison – Top 5 States

These visuals represent bar charts of district wise variation of total cases and deaths of top 5 infected states of India. OnHover event is included.

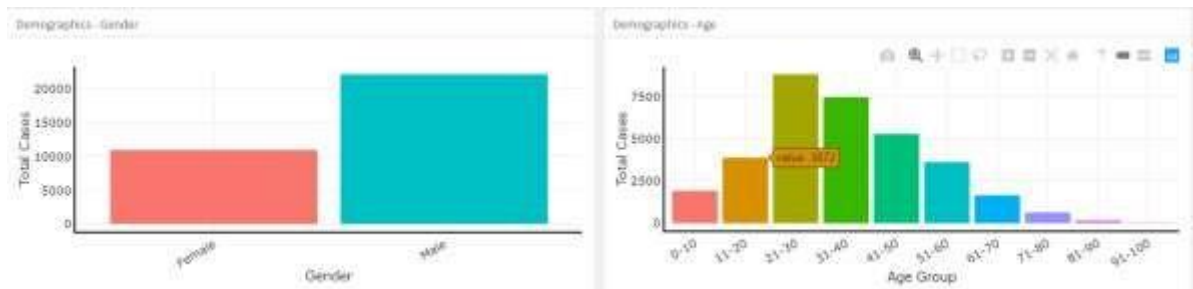
Observations: These graphs represent the top 10 worst affected districts of the 5 majorly affected states of India indicating the fact that major metropolitan cities like New Delhi, Mumbai, Chennai, Bangalore etc. have been reporting greater cases when compared to smaller cities and districts.



4.3.6 Demographics – Age and Gender

These visuals represent bar charts of gender and age variation of total cases of India. OnHover event is included.

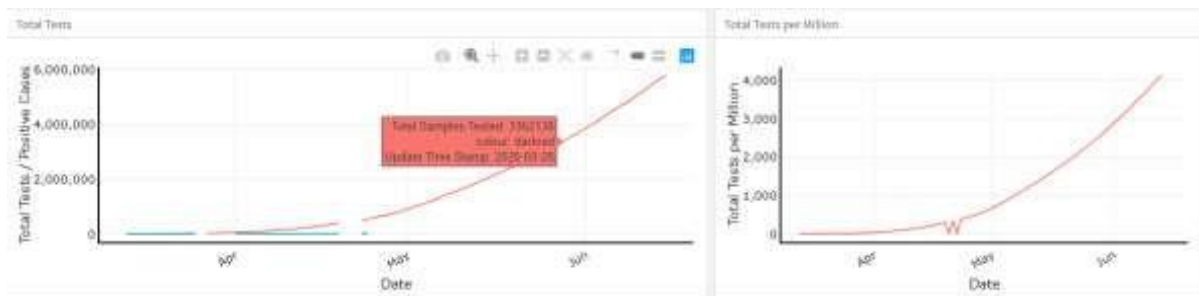
Observations: These graphs represent the overall demographics of the cases indicating that the ratio of infected males is greater and people between the age group of 21 and 30 are most infected.



4.3.7 Total COVID-19 Tests and Tests per Million

These visuals represent line charts of date wise variation of total tests and tests per million conducted in India. OnHover event is included.

Observations: The first visual indicates how India has ramped up its testing facilities to contain the spread of the virus. Although, the tests per million still remains low due to a greater population.

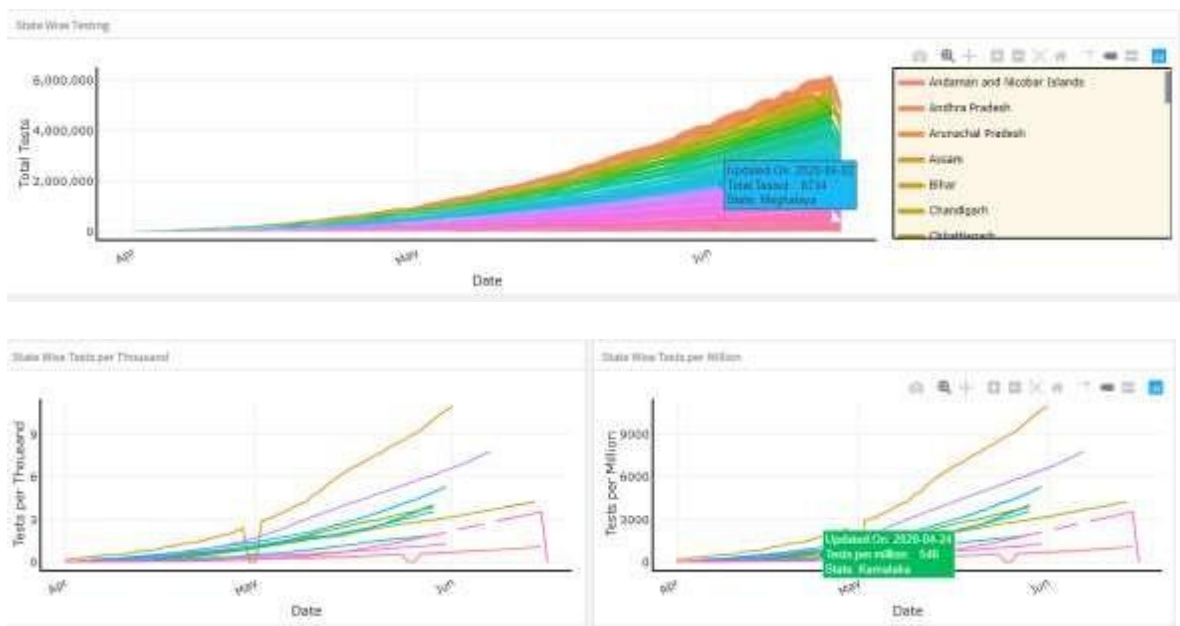


4.3.8 State Wise COVID-19 Tests, Tests per 1000 People, Tests per Million

These visuals represent stacked area charts of date wise variation of total tests, tests per 1000 and tests per million by states of India. OnHover event and legend selection is included.

Observations: These visuals indicate the state wise distribution of tests reported

every day. Kerala was successfully able to contain the spread, even though it reported the first case in India, can be attributed to the fact that it followed policies of extensive contact tracing and testing. A similar pattern was followed by Rajasthan and Karnataka which reported very high number of cases initially but were able to bring to down the numbers in later stages of the pandemic. In Maharashtra, there has been an exponential rise in cases without signs of slowing down which can be attributed to the fact that Mumbai emerged as a hotspot due to cramped living conditions and inconsistent testing and contact tracing. In contrast, states like Delhi, Gujarat and Tamil Nadu, even though with excellent test reporting, have been facing greater number of cases daily.



IV. CONCLUSION AND FUTURE SCOPE

The entire world is in midst of a serious pandemic which has affected more than 200 countries causing more than 7 million infected cases and 0.4 million deaths. This pandemic has taken its economic and financial toll on most of the major economies of the world.

This project aimed at analyzing the current situation of the pandemic by creating intuitive and user interactive dashboards and drawing conclusions on the impact it will have on the world. Currently, the project is in its last stage of development with the dashboards been developed and submitted for review and feedback.

With regards to the future work, the firm aims at regularly updating the dashboards with time and integrating it with their systems so as to continually draw conclusions and analyze the results. This will enable them to predict future business opportunities and provide a basis on which they can plan on increasing their market presence and capacity planning.

V. REFERENCES

Data Collection

The following websites have been referred to obtain the input data and statistics:

- a. <https://api.covid19india.org/>
- b. <https://www.aa.com.tr/en/latest-on-coronavirus-outbreak/worldwide-covid-19-testing-ratio-per-country-million/1800124#>
- c. <https://www.tableau.com/covid-19-coronavirus-data-resources>
- d. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data
- e. <https://github.com/owid/covid-19-data/tree/master/public/data>

Programming References

The following websites have been referred for R coding and Shiny tutorials:

- a. <https://datascienceplus.com/category/programming>
- b. <https://rstudio.com/resources/webinars/>
- c. <https://bookdown.org/yihui/rmarkdown/document-templates.html>
- d. <https://datascienceplus.com/map-visualization-of-covid19-across-world/>
- e. <https://bookdown.org/yihui/rmarkdown/dashboards.html>
- f. <https://rmarkdown.rstudio.com/lesson-12.html>
- g. <https://bookdown.org/yihui/rmarkdown/cheat-sheets.html>
- h. http://www.htmlwidgets.org/showcase_leaflet.html
- i. http://jeffgoldsmith.com/p8105_f2017/shiny.html
- j. https://rmarkdown.rstudio.com/flexdashboard/using.html#page_icons