Visualizing the COVID-19

Final Report

DSC 465 - Fall 2020

Rittam Debnath
rdebnath@depaul.edu

Gerado Palacios
gpalacios1019@protonmail.com

Viswa Chaitanya Seelam
viswachaitanya22@gmail.com

Azat Dovgeldiyev
adovgeld@depaul.edu

Instructor

Dr. Eli T. Brown

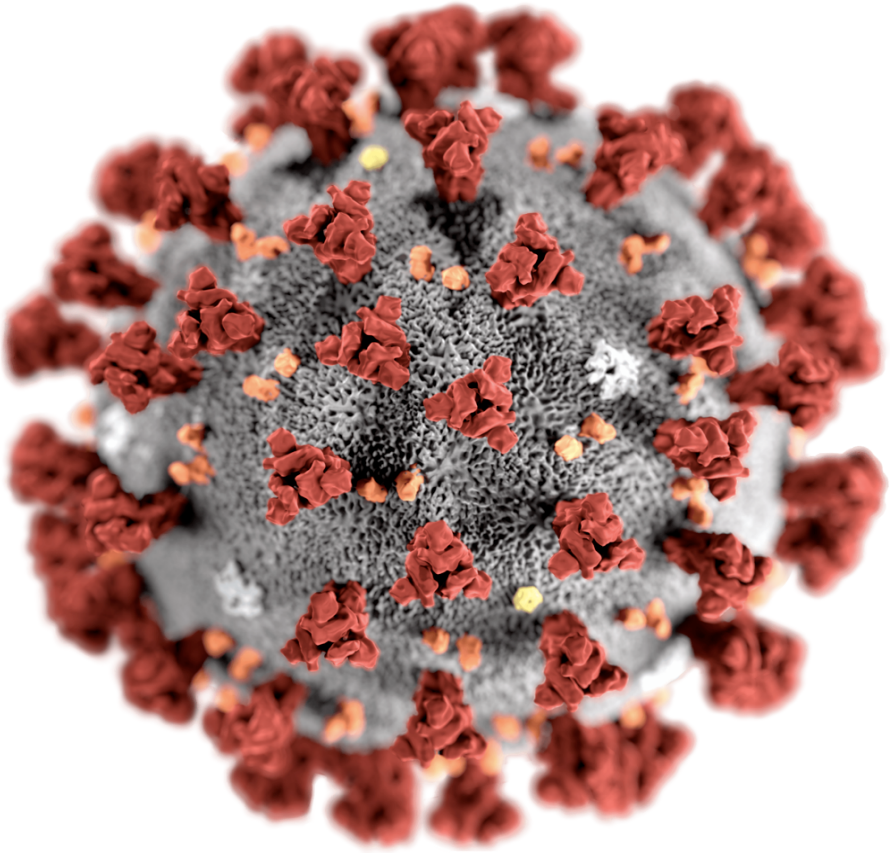


Table of Contents:

Introduction

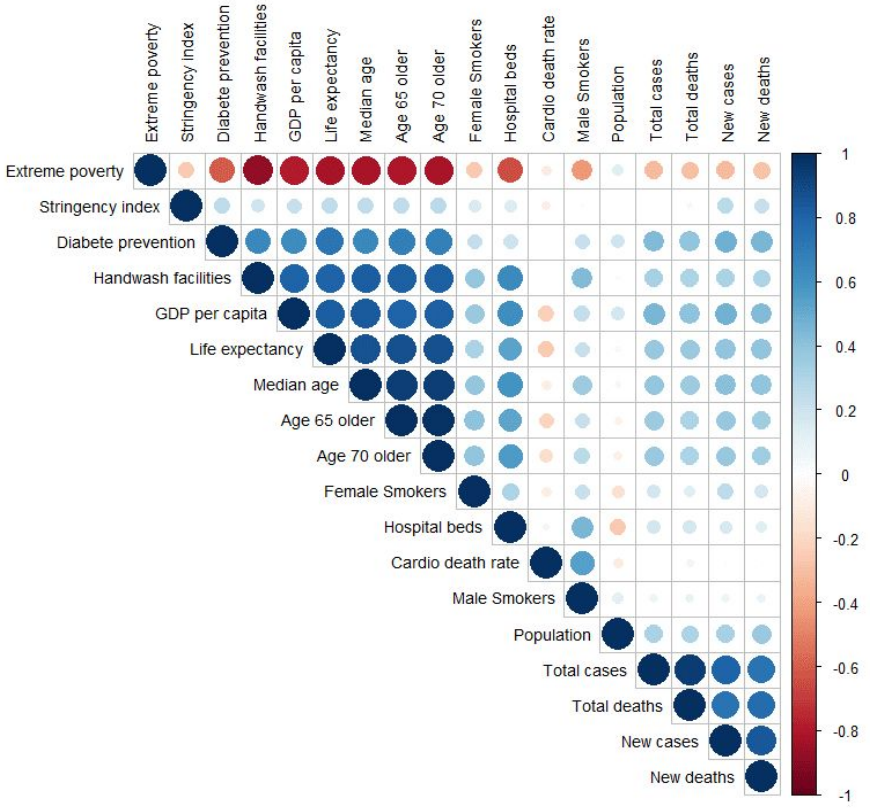
There's a highly contagious infection originating from Wuhan, China called COVID-19, otherwise called Covid. It is a group of infections that can cause a wide range of symptoms ranging from mild to fatal, for example, SARS or MERS. It began in December 2019. China's administration asserts that this infection was appended to bats initially; nonetheless, a few residents attempted to taste the kind of game meat, and afterward, the infection began to spread to individuals. As per the WHO, the COVID-19 assessments of the hatching time frame is 14 days. For most contaminated individuals, the side effect will show within five to six days. The WHO also expresses that a few contaminated individuals can be asymptomatic, and it implies they will not show any indications regardless of the transporter—finally, the WHO raises the caution and pronounces the emergency pandemic on March 11, 2020.

To understand this virus's effects worldwide, our team has gathered data on COVID-19, which was extracted from ourworldindata.org. This data was sourced from ECDC (European Center for Disease Prevention and Control) and WHO (World Health Organization). After removing incomplete entries, the dataset consists of 34,932 observations with 25 different attributes, three categoricals, and twenty-two numericals. Each daily observation represents a country's covid statistic (deaths, cases, etc.) along with various country indexes and descriptors between January 6, 2020, and October 6, 2020. Our initial step was to clean the data set with removing unnecessary attributes, blanks, duplicates, reformatting data types, renaming variable names, converting character types into numeric and so on. We changed the type of location from string to geographic, so we could generate longitude and latitude and plot maps. The cleaning and preprocessing were completed in RStudio.

Exploratory Analysis

We created a PCA plot and correlation plots in RStudio using both pearson and spearman methods between various attributes to check the relationships, such as the number of cases, new cases, numbers of deaths, population, poverty, male smokers, Diabetes prevalence, and several other categories listed in our data. With the use of the correlation visualizations, we have found a high correlation of population, stringency index with total deaths, and cases in a given country.

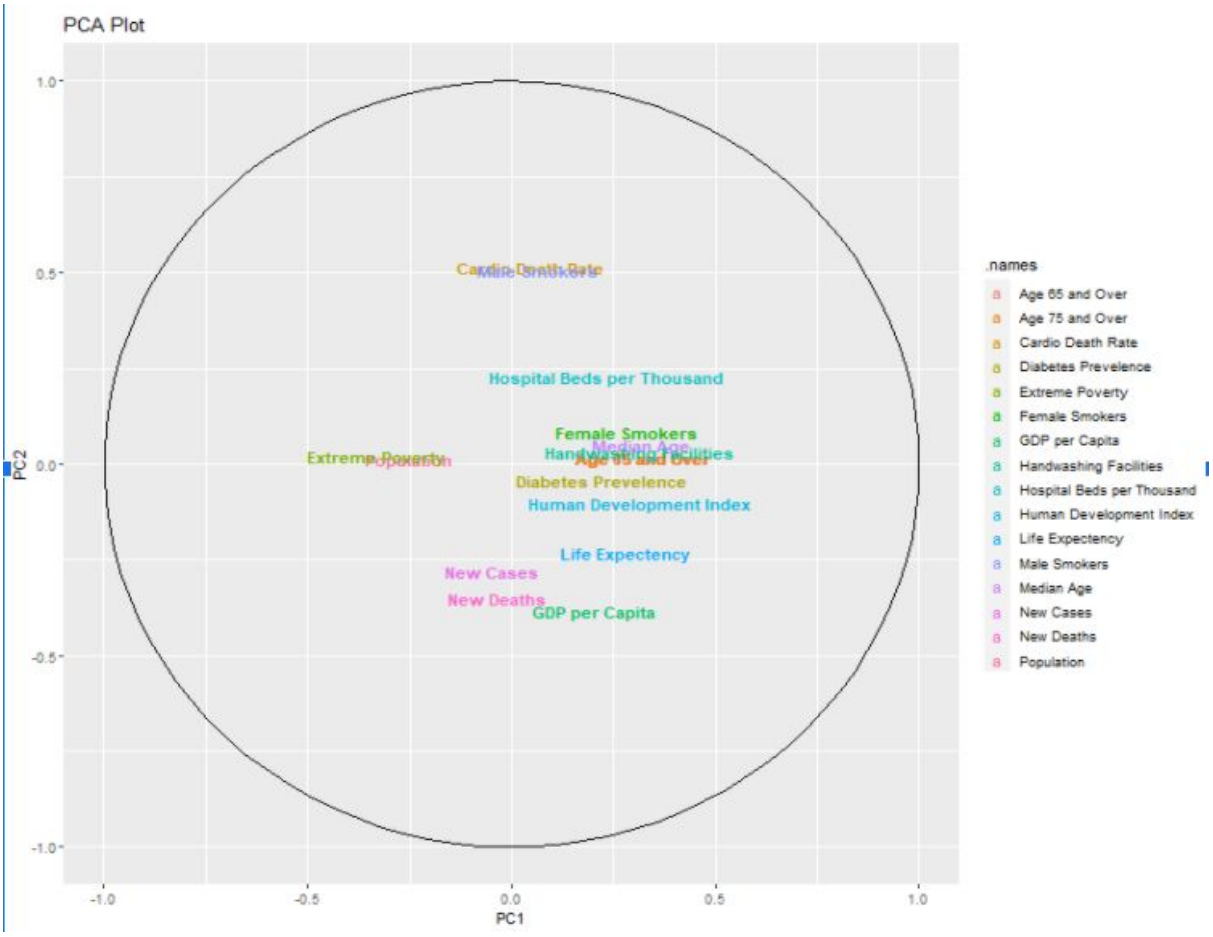
The correlation plots give us a better understanding on which direction we should focus on, there are some strong positive relationships as well as negative ones. We have created the following explanatory visualizations to show the correlations between the various attributes. As one can note from the correlation plot, there is significant multicollinearity within the dataset. This is to be expected with a complex time-series dataset. As an additional step for initial exploration, we also conducted an initial principal components analysis to look for any potential latent factors.



Visualizations

Generally, the initial PCA**(Visual #1)** The plot of the two major factors is the General quality of life, which lists attributes such as the human development index and extreme poverty, showing that continents that had a lower population and lower rates of extreme poverty did generally better than others. This also shows that the majority of higher quality variables are also highly correlated with increased number of hospital beds, senior population and life expectancy.

Visual #1 - ***Potential Latent Covid19 Factors***

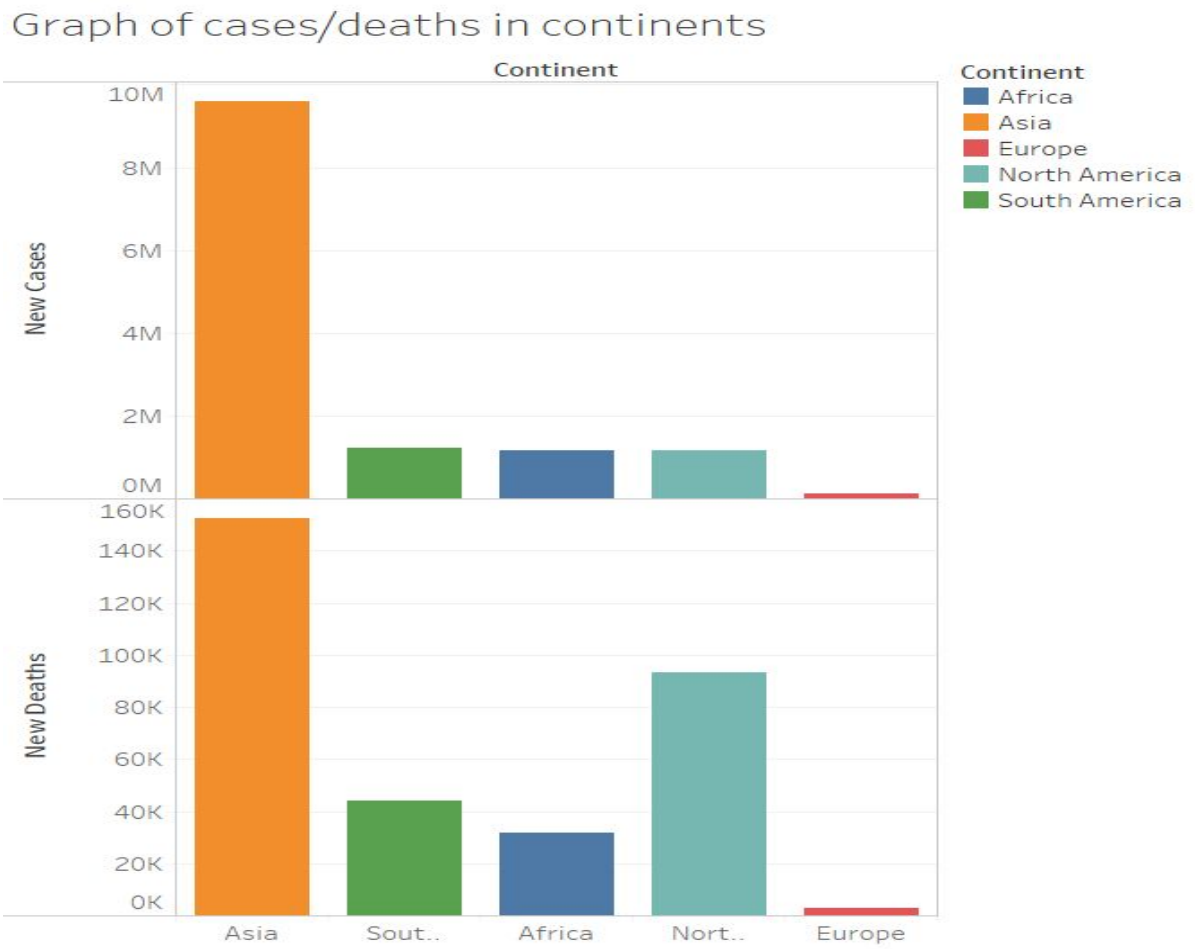


Principal Factor Analysis - Rotated Unscaled Covariance PCA analysis

General Quality of Life		***Prevalence of Senior Population***		***Hospital Quality and Smoker Prevalence***		***Cases and Deaths***	
Population	*-0.812*					*New Cases*	*0.984*
Extreme Poverty	*-0.930*					*New Deaths*	*0.942*
Median Age	*0.861*	*Age 65 and Over*	*0.689*	*Hospital Beds*			
Age 65 and Over	*0.692*	*Age 75 and Over*	*0.677*	*Per Thousands*	*0.598*		
Age 75 and Over	*0.701*	*Female Smokers*	*0.672*	*GDP per Capita*	*-0.569*		
Handwashing Facilities	*0.952*	*GDP per Capita*	*-0.569*	*Cardio Death Rate*	*0.978*		
Hospital Beds per Thousand	*0.704*	*Diabetes Prevalence*	*0.963*	*Male Smokers*	*0.957*		
Life Expectancy	*0.895*						
Human Development	*0.920*						
Female Smokers	*0.534*						

Following the exploration plots**(Visual #2)** are the bar graphs to look at the data in a more systematic fashion. Since the bar graphs are good for categorical 'x' values and cases where the 'y' value ratio scaled, we decided to visualize continents with a number of new cases and deaths. The bar graphs were created in Tableau with three different variables, new cases, new deaths and continents, colors representing each continent. Bars are in descending order by new cases. New cases and new deaths include all the records from the start date until October 30, 2020. Since India itself has new cases with more than 8 million and deaths more than 130 thousand, there is a huge gap between Asia and the rest of the world.

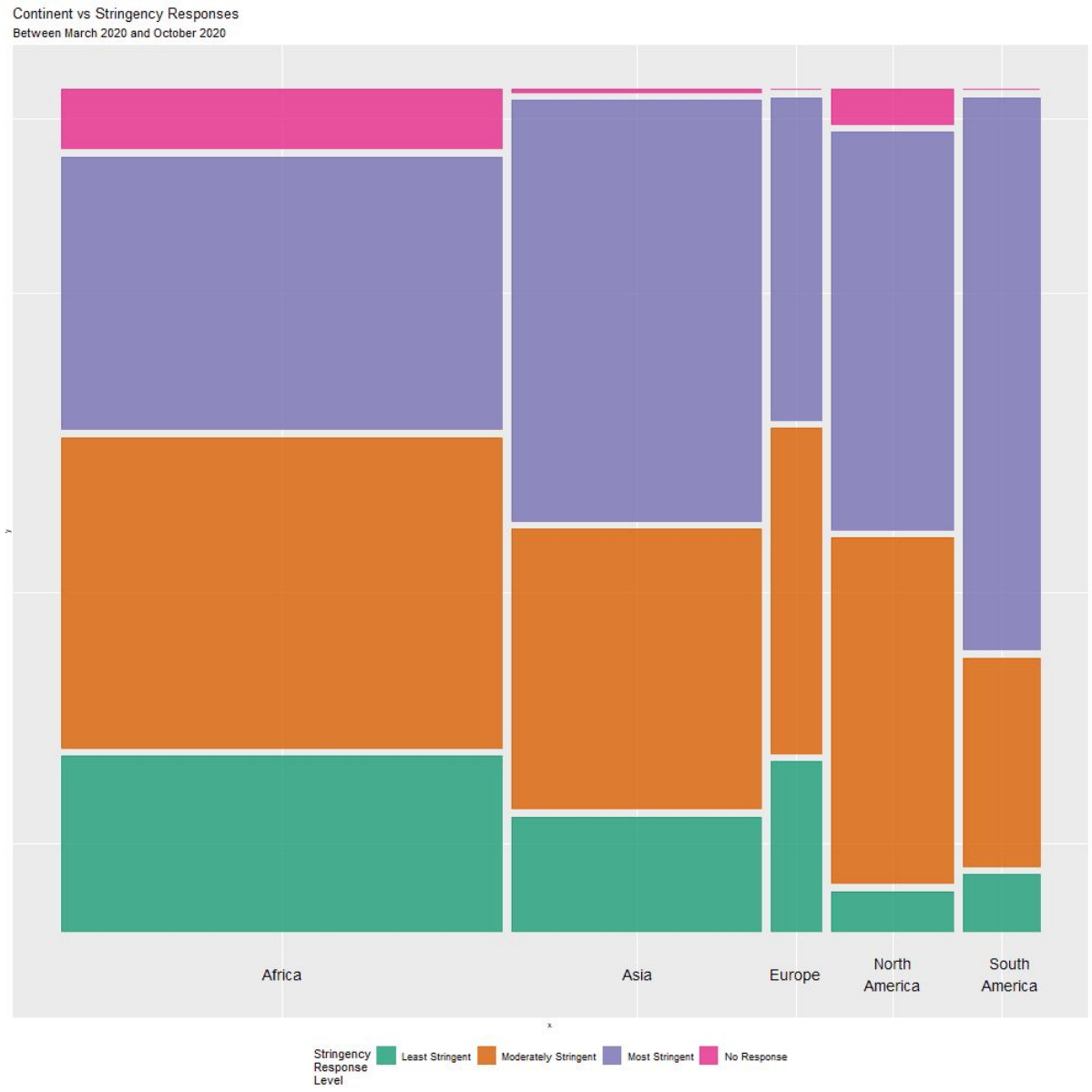
Visual #2 - *Quick look into World data with cases*



The third visualization, **(Visual #3)** mosaic plot visualizes the relationship between continents and stringency response levels between the months of March and October. A mosaic plot helps visualize two categorical variables. In this case we are organizing it by

frequency of observations in the dataset that adhere to continents and stringency levels. After creating a categorical variable from the stringency index, the visualization can quickly highlight how aggressive some continents responded to the pandemic between March and October. Mainly observing Asia and South America as the continents applying the most stringent responses from the other continents.

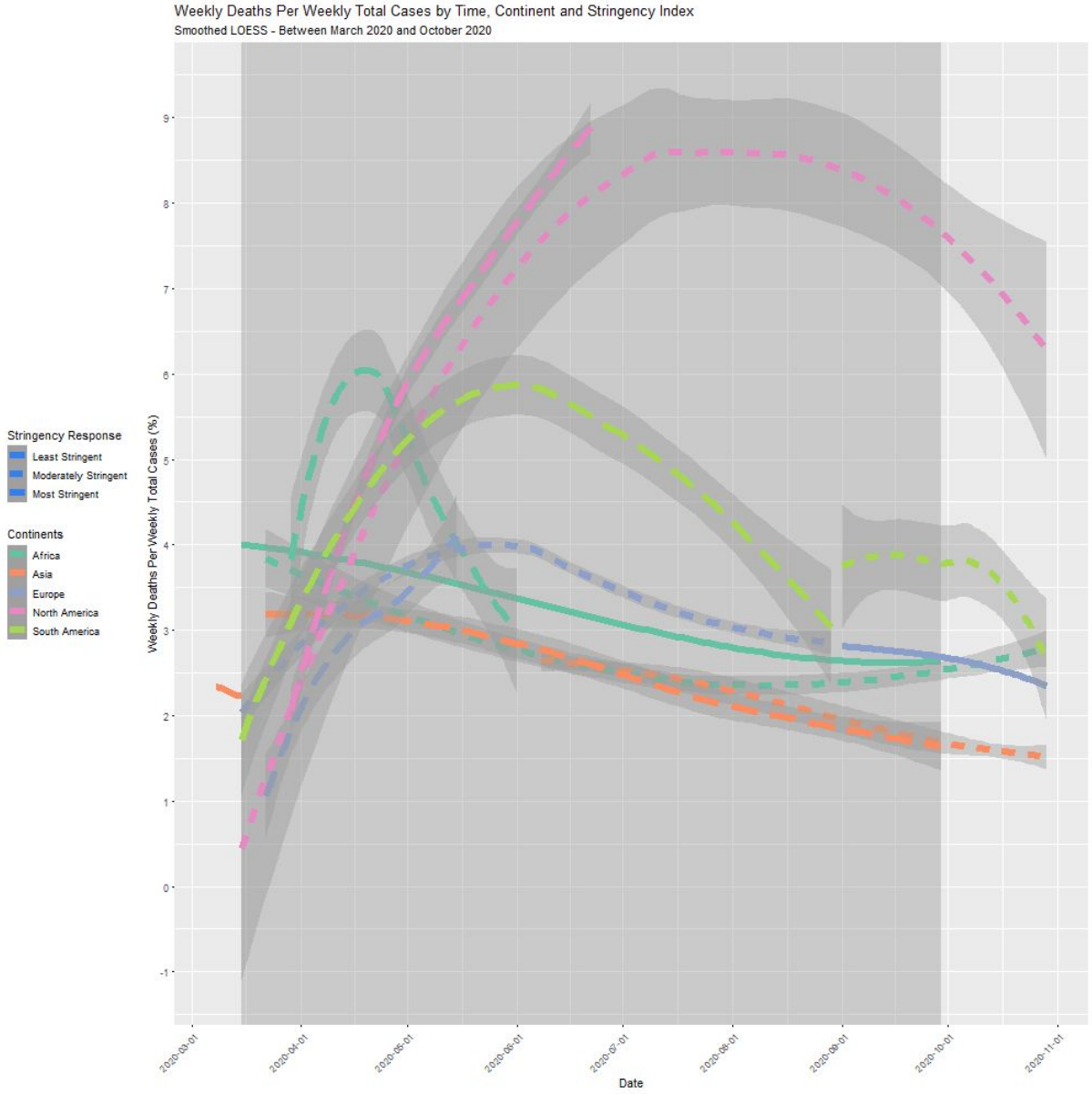
Visual #3 - *Continent vs Stringency Response Levels*



Contrary to the mosaic plot, in the fourth visualization(**Visual #4)** we use summarized data and applied locally weighted smoothing to depict how different countries were affected

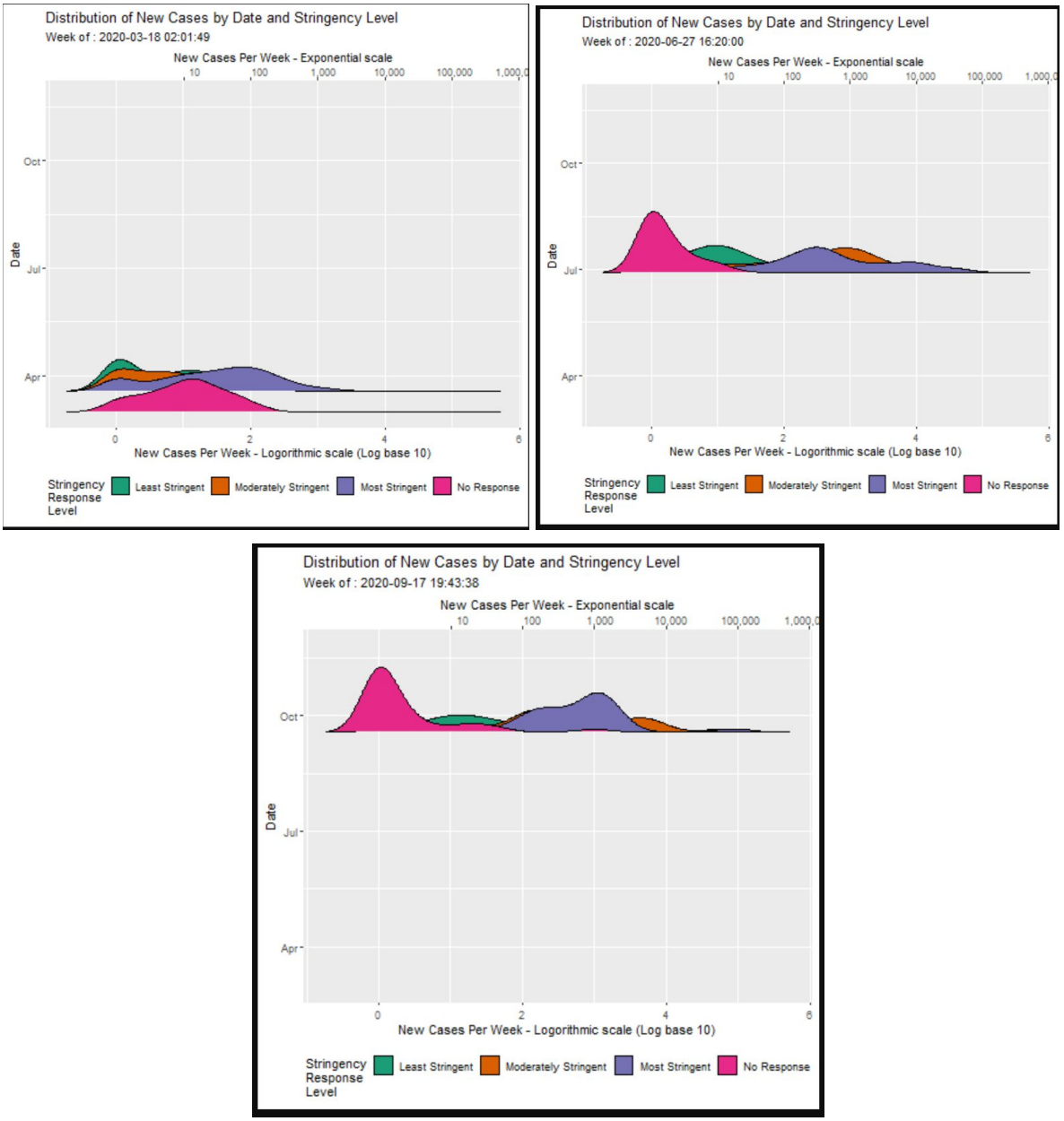
through the timeplot. This plot looks at the relationship between death rate and stringency response levels and continents between the same time frame.

Visual #4 -Average Weekly Deaths per Average Total Cases by Time, Continent and Stringency Index



Animated graph (**Visual #5)** was applied to show the distribution of new cases by stringency level in the given period. Each color represents the stringency response levels, months on y axis and new cases per week on x axis.

Visual #5 -Deaths per Total Cases by Time, Continent and Stringency Index (animation)



Analysis and Discussions

By extracting a massive dataset that has been constantly updated is a gruesome task. Not only that, this dataset consists of attributes like total cases, death, and stringency index collected from all over the world for COVID-19. We found that there are many ways to manipulate such a large chunk of dataset and create visuals that can give us a broad understanding of how things are happening and why. We started with the creation of the correlation plots. The discovery of the positive correlations between the Total Cases, new cases with Stringency and population has given the direction to explore the trends over time and by location. For our final report we have created five visualizations to explore the trends within the data.

To start, we wanted to see if there were any potential latent factors that contribute to the heavy multicollinearity within the dataset. Since we are dealing with a large number of numerical variables with different scales, a PC analysis helps group correlations into potential factors that could be used to understand the main drivers. An initial PCA analysis shows that 95% of the data can be explained into four potential latent factors contributing to COVID19 rates. These could be described as:

1: General Quality of Life

2: Prevalence of Senior Population

3: Hospital Quality and Prevalence of Smokers

4: Cases and Deaths

The largest explanatory factors are within the first two, which can logically be explained that continents that had a higher prevalence of quality of life also had a higher prevalence of a senior population and vice versa. This means that the combination of variables that describe those factors have a greater influence together than separately.

Then we explored data by visualizing new COVID-19 cases by dates, cases and deaths by continents. Most cases appear in Asia with nearly ten million, with India contributing eight million new cases between given dates. This would be understandable since populations in those continents are larger. In order to mitigate this, we applied a log base of ten to give greater weight towards the continents with a smaller population. Next, the mosaic plot we found the most stringent responses in Asia and South America. This detail can be further analyzed by smoothing the data. As shown from the mosaic plot, Asia and South America had the highest frequency of stricter responses. The result of these responses

throughout the pandemic can be highlighted in the locally weighted smoothing plots. These plots helped us to discover the relationship between death rate and stringency response. It can be easily noted that Asia and South America had a steady decrease in new deaths over new cases when compared to the other continents. This is incredibly important to understand since applying strict protocols within early onset of the pandemic showed dramatic reduction in cases. Whereas, continents that did not respond as quickly or as often can be shown with exponential increase in cases with a very slow decline. Finally, the animated density plots were useful in creating visualization deaths per total cases by time, continent and stringency level. The distributions of each of new cases can be seen in having dramatically different spreads between stringency responses. As shown in the still animation, the stringency responses that were stricter tended to have wider spreads, while low stringency responses had much narrower spreads indicating a much higher chance of getting additional, (predictable) new cases.

Overall, the pandemic has caused widespread chaos with each part of the world reacting and responding independently. The plots can clearly show that response levels are incredibly important to reduce potential cases.

Appendix

Rittam debnath - Individual Report

In this group,I have played an active part during this course project. I was responsible for creating initial documents to facilitate smooth project delivery. I was responsible for creating several charts including line graphs, maps and bar charts to display the Total cases per country**(RD1)**, Number of New cases registered per quarter **(RD2)**. state wise relationship of India**(RD3)**. The choropleth graph(RD1) shows the relationship between the number of total cases per country. This graph is highlighted using a divergent color schema which can easily be spotted by any type of audience, this signifies that the total cases in India is highest, according to our data set. The second plot(RD2) is a line graph that explains the number of new cases registered per million divided per quarter which shows that there was a significant increase in the new cases between june and july and then it gradually started decreasing in the mid-august. For my final set of bar graphs(RD3), I took a different approach to tie down our story of moving from the world data to exploring the statewise data in India and found out that the the number of positive cases in major cities like Mumbai (State: Maharashtra) and Delhi(UT) increase rapidly whereas the number of positive cases in other states were significantly low.

After the course of 10 weeks, I can spot the difference between different kinds of data visualization charts and know what chart to be used when. During this course, I have learned to use R studio that helped me to understand the importance of statistics in data visualization. However, it was challenging for me to learn a programming language. On the other hand, I have extensively used Tableau software to generate charts to find incredible relationships between various data variables. As an HCI student, I did research on colors, size and overall aesthetics of the visualizations. I am sure that this class will be a great addition to my skill set.

Viswa - Individual Report

I have played an active role in the project. I actively communicated with everyone to organize meetings and discuss the project deliverables. I have created the final presentation and presented it on the Voice Thread. I have created a few initial visualizations. VCS 1 is a geographical plot of total deaths that occurred in different countries around the world. From

the visualization, we can see that India had the highest number of deaths, followed by Mexico, Colombia, South Africa, and Indonesia.

I have created the second visualization VCS 2, to compare the relationship between Stringency Index and new cases by each week for four countries with the highest number of cases. From the visualization, we can see the countries had high stringency responses during the initial weeks of COVID. We can also see that India has reduced the stringency index from week 19, after which the cases started rising dramatically.

Being a part of the project group, I was able to learn how to use visualizations to find trends from a broad dataset. The correlation plots helped me to understand the overall relationship between the different categories of the data. With the use of the correlation plots, we have found significant relationships between a few categories and the COVID cases and deaths. During the course of the project, I have improved my knowledge of Tableau and R studio. I had a great learning experience through collaboration with my teammates, two data scientists, and one designer.

Gerardo - Individual Report

My primary role in the group was to explore the dataset and create charts using RStudio. Specifically, I was in charge of organizing the data to develop the PCA plot, Smoothed LOESS, the mosaic plot, and the animated ridge distributions. Each plot required careful customization to portray and highlight the correct information. For example, the PCA and LOESS plots required additional preprocessing to achieve the result. Since our complex dataset mainly consisted of indexes with little spacing between observations, it was necessary to create a new variable. In this case, the newly binned stringency index transforms into an ordered categorical variable with four levels. This transformation allowed the mosaic plot and the LOESS plots to reveal a higher detail level by clearly dividing the differences in stringency responses. Another leading transformation was summarizing and pivoting the data to reflect a more consistent time frame. In this case, I summarized the data using mean and mode for every single week. A summarized view of the data helps reduce potential noise and present cleaner plots. The summarized data gave way to create the LOESS line plots and the PCA plot. The plots help reveal trends over time and how different response levels can affect deaths and cases.

I have found data visualization incredibly helpful and challenging, especially with R programming's heavy use. During this project, I found it exciting to see how the project's

visualizations reflected those on the current news. The tutorials were incredibly helpful in differentiating useful information and wrong information when approaching a dataset with RStudio. This project became heavy with manipulation due to its different scales and uneven time intervals; however, RStudio helped facilitate all the necessary transformation. Much of it was hours spent in trial and error, but the outcome resulted in becoming incredibly comfortable with manipulating and displaying information solely out of RStudio.

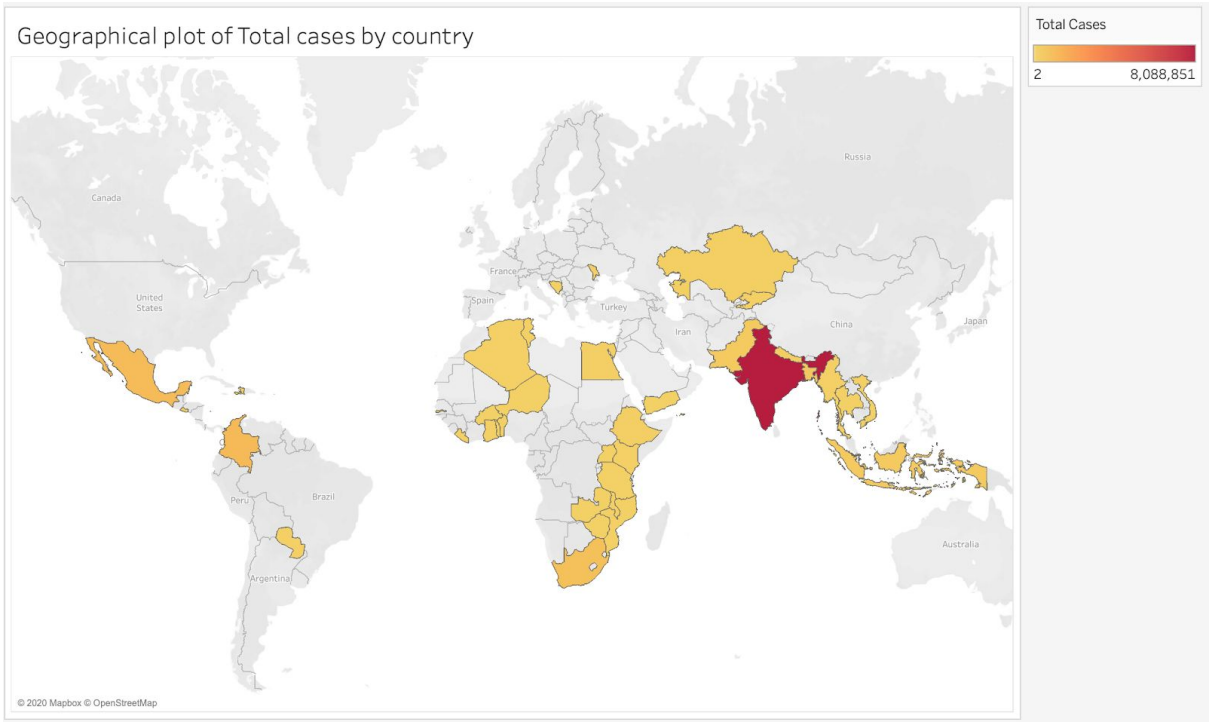
Azat - Individual Report

In this project my main task was to clean, preprocess the data set to make it more readable and to create graphs. I used RStudio in the cleaning process and Tableau to generate the longitude and latitude so we could use RStudio to experiment with some choropleth graphs. I created correlation plots (AD0 for codes) to investigate the dependence between multiple variables, for example, some strong positive relationships helped us to explore the insights. Animated line plots (submitted in additional .gif file) were used to show the timeline of new cases by each month in India. Animated world map (submitted in additional .gif file) shows the trends of the new cases within the given time. There were some other graphs such as density plots that were not included in the presentation. Density plots with “raster” geometry (AD1) and density plots with less dense versions (AD2) were applied to observe the distribution of new cases in the dataset.

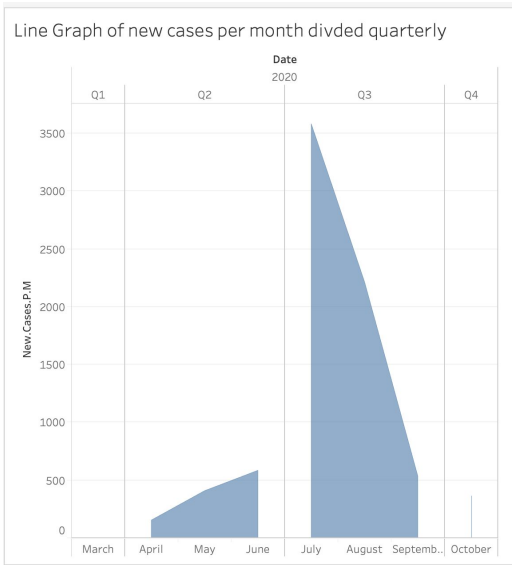
Being a part of the project group, I have learned how data visualizations make big and small data easier for the human brain to understand, and make it easier to detect patterns, trends, and outliers in groups of data. Class tutorials were really helpful in creating graphs, avoiding chart junks, checking for clarity and graphical integrity, and using statistical methods for business audiences. Color schemes and choosing right color palettes helps visualizations to look better. I found RStudio much powerful both in statistical and visualizational terms, and I believe I will apply the powerful tools we learned during the quarter to make presentations more efficient and beautiful.

Plots and Graphs with R codes

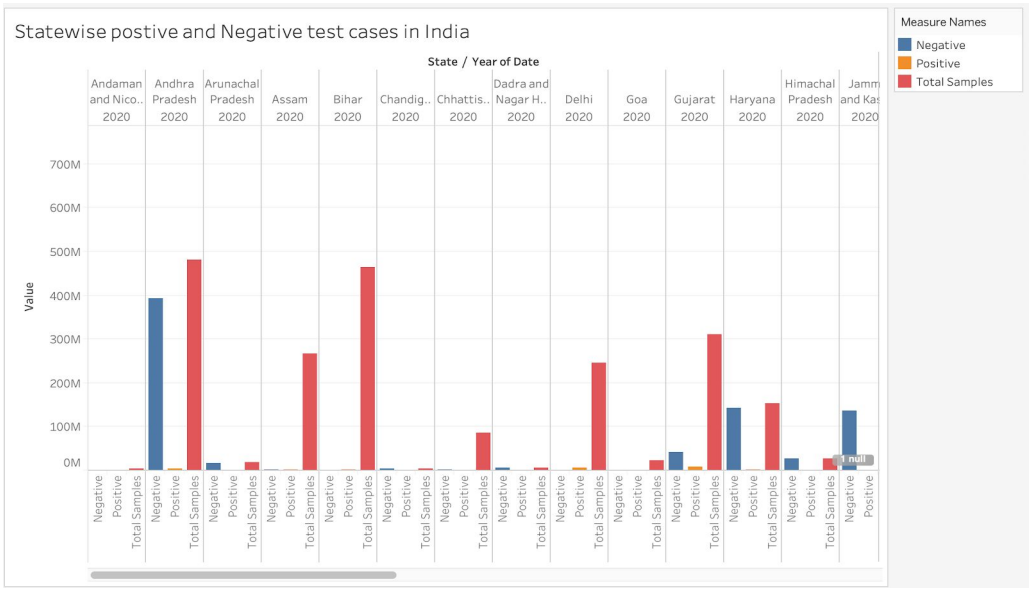
RD(1):

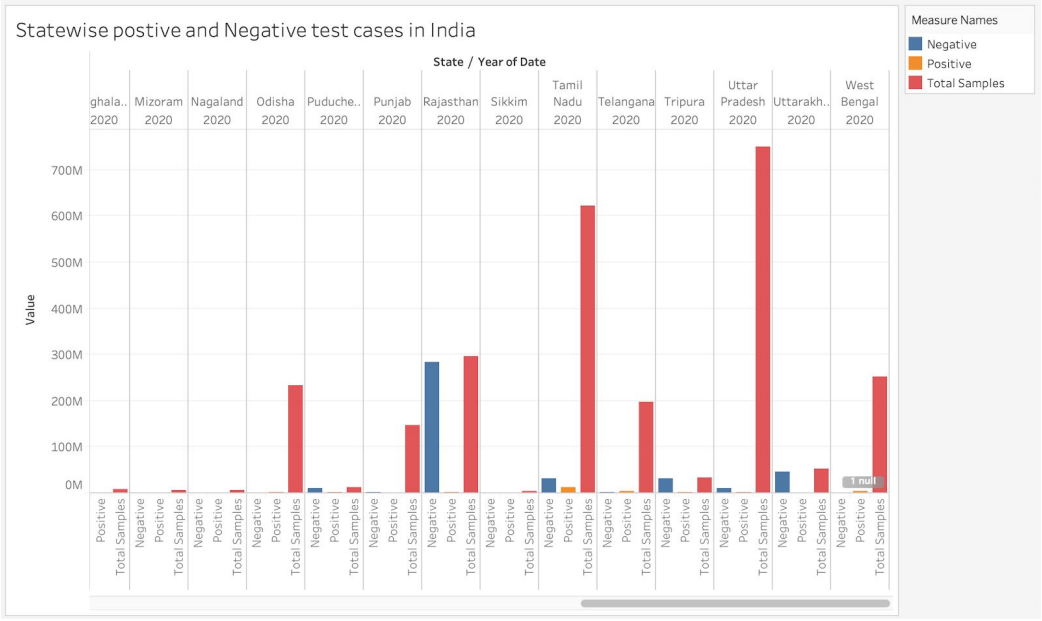


RD(2):

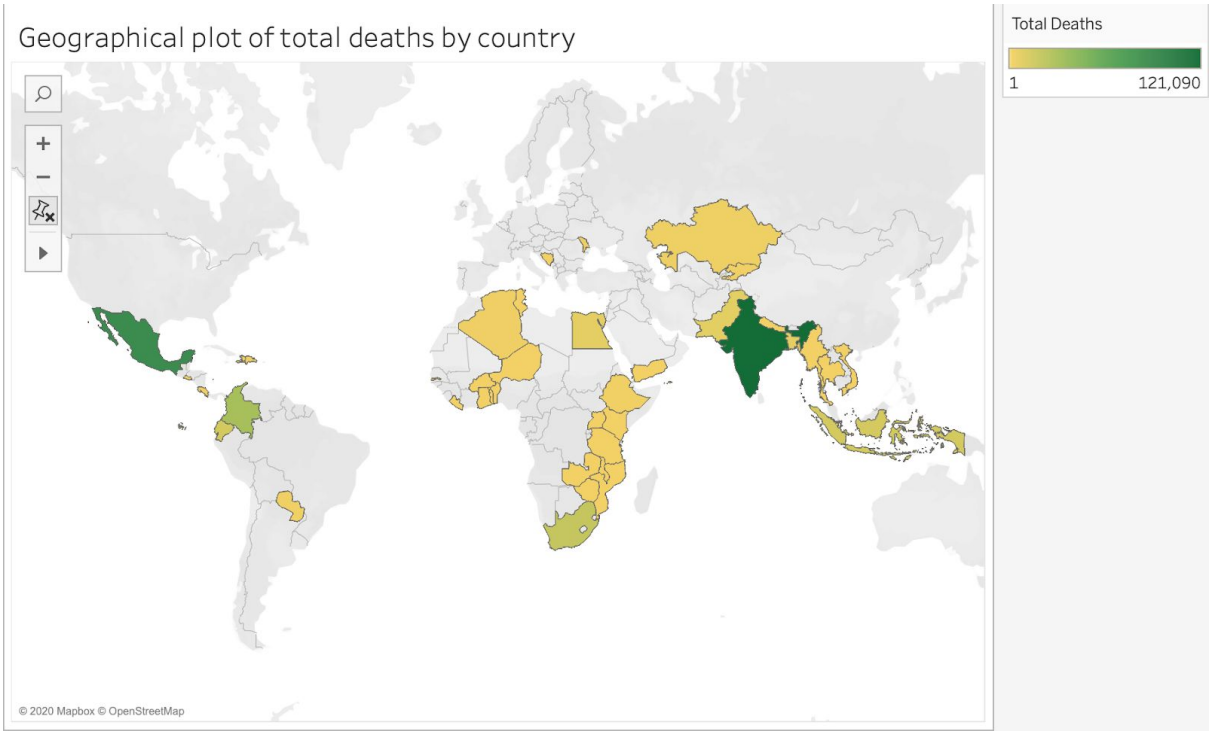


RD(3):

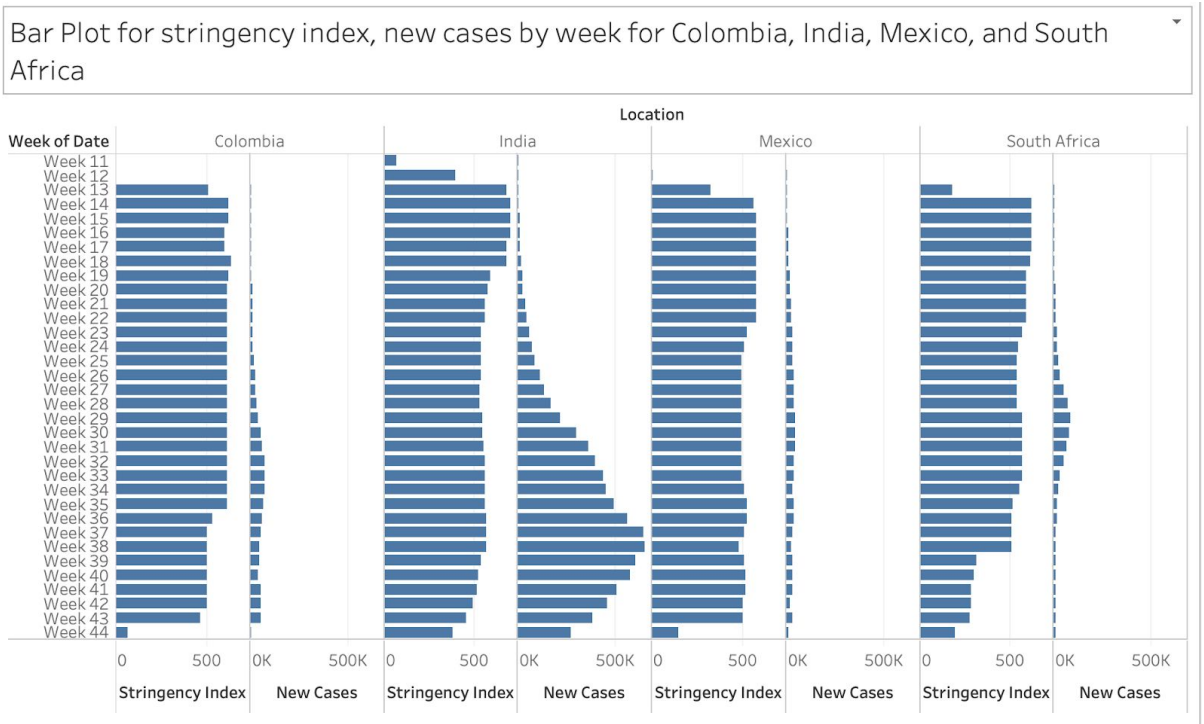

Statewise postive and Negative test cases in India
State / Year of Date
Andaman and Nico.. 2020
Andhra Pradesh 2020
Arunachal Pradesh 2020
Assam 2020
Bihar 2020
Chandig.. 2020
Chhattis.. 2020
Dadra and Nagar H.. 2020
Delhi 2020
Goa 2020
Gujarat 2020
Haryana 2020
Himachal Pradesh 2020
Jamm and Kas 2020
Value
700M
600M
500M
400M
300M
200M
100M
0M
Negative
Positive
Total Samples
1 null
Measure Names
Negative
Positive
Total Samples


Statewise postive and Negative test cases in India
State / Year of Date
ghala.. 2020
Mizoram 2020
Nagaland 2020
Odisha 2020
Puduche.. 2020
Punjab 2020
Rajasthan 2020
Sikkim 2020
Tamil Nadu 2020
Telangana 2020
Tripura 2020
Uttar Pradesh 2020
Uttarakh.. 2020
West Bengal 2020
Value
700M
600M
500M
400M
300M
200M
100M
0M
Positive
Negative
Total Samples
1 null
Measure Names
Negative
Positive
Total Samples

VCS 1


Geographical plot of total deaths by country
Total Deaths
1
121,090
© 2020 Mapbox © OpenStreetMap

VCS 2


Bar Plot for stringency index, new cases by week for Colombia, India, Mexico, and South Africa
Location
Week of Date
Colombia
India
Mexico
South Africa
Week 11
Week 12
Week 13
Week 14
Week 15
Week 16
Week 17
Week 18
Week 19
Week 20
Week 21
Week 22
Week 23
Week 24
Week 25
Week 26
Week 27
Week 28
Week 29
Week 30
Week 31
Week 32
Week 33
Week 34
Week 35
Week 36
Week 37
Week 38
Week 39
Week 40
Week 41
Week 42
Week 43
Week 44
0
500
0K
500K
Stringency Index
New Cases

AD0 - R codes for corrplot

Only those variables with numeric values

```
num_data <-num_data %>%
  rename(
    "New cases" = New.Cases,
    "New deaths" = New.Deaths,
    "Total deaths" = Total.Deaths,
    "Total cases" = Total.Cases,
    "Male Smokers" = Male.Smokers,
    "Female Smokers" = Female.Smokers,
    "Cardio death rate" = Cardio.Death.Rate,
    "Hospital beds" = Hosp.Beds.P.Thsd,
    "Age 70 older" = Aged.70.Older,
    "Age 65 older" = Aged.65.Older,
    "Median age"= Median.Age,
    "Life expectancy" = Life.Expectancy,
    "GDP per capita" = Gdp.Per.Capita,
    "Handwash facilities" = Handwash.Fac,
    "Stringency index" = Stry.Index,
    "Extreme poverty" = Extreme.Poverty,
    "Diabete prevention" = Diabet.Prev
  )
```

Spearman method with corrplot library:

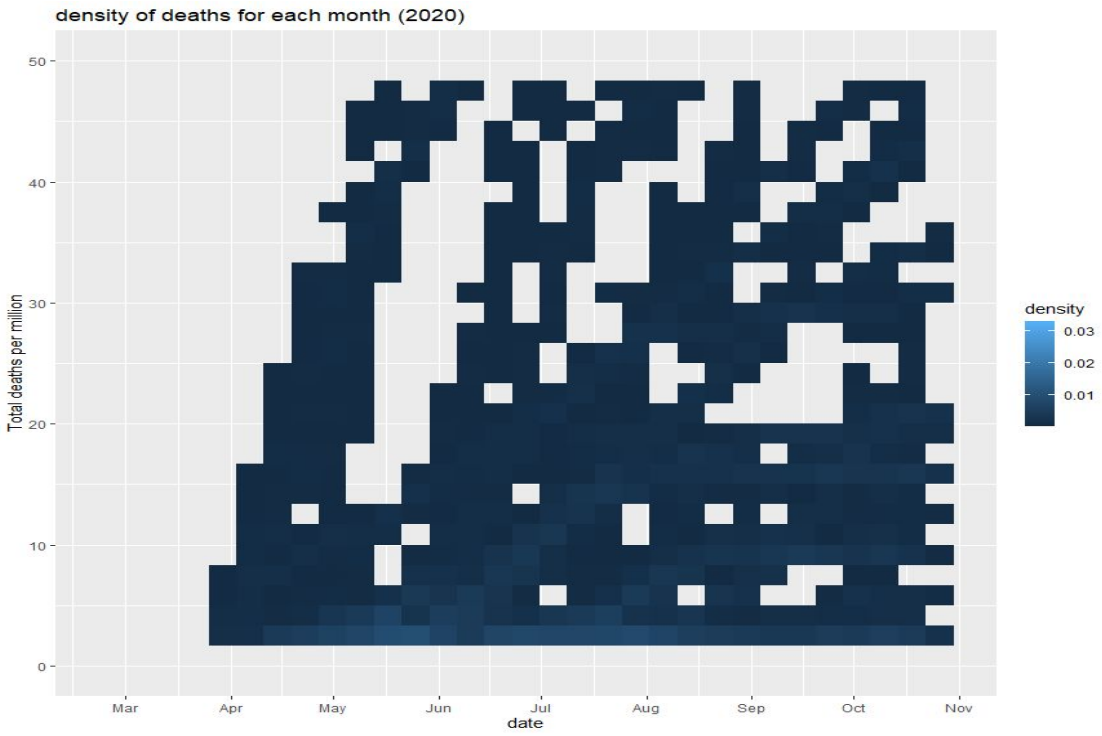
```
spearCor <- cor(num_data, method = 'spearman')
corrplot(spearCor, type = "upper", order="hclust",tl.cex = 0.9,tl.col="black")
```

AD1 plot + code

Created function to control large numbers.

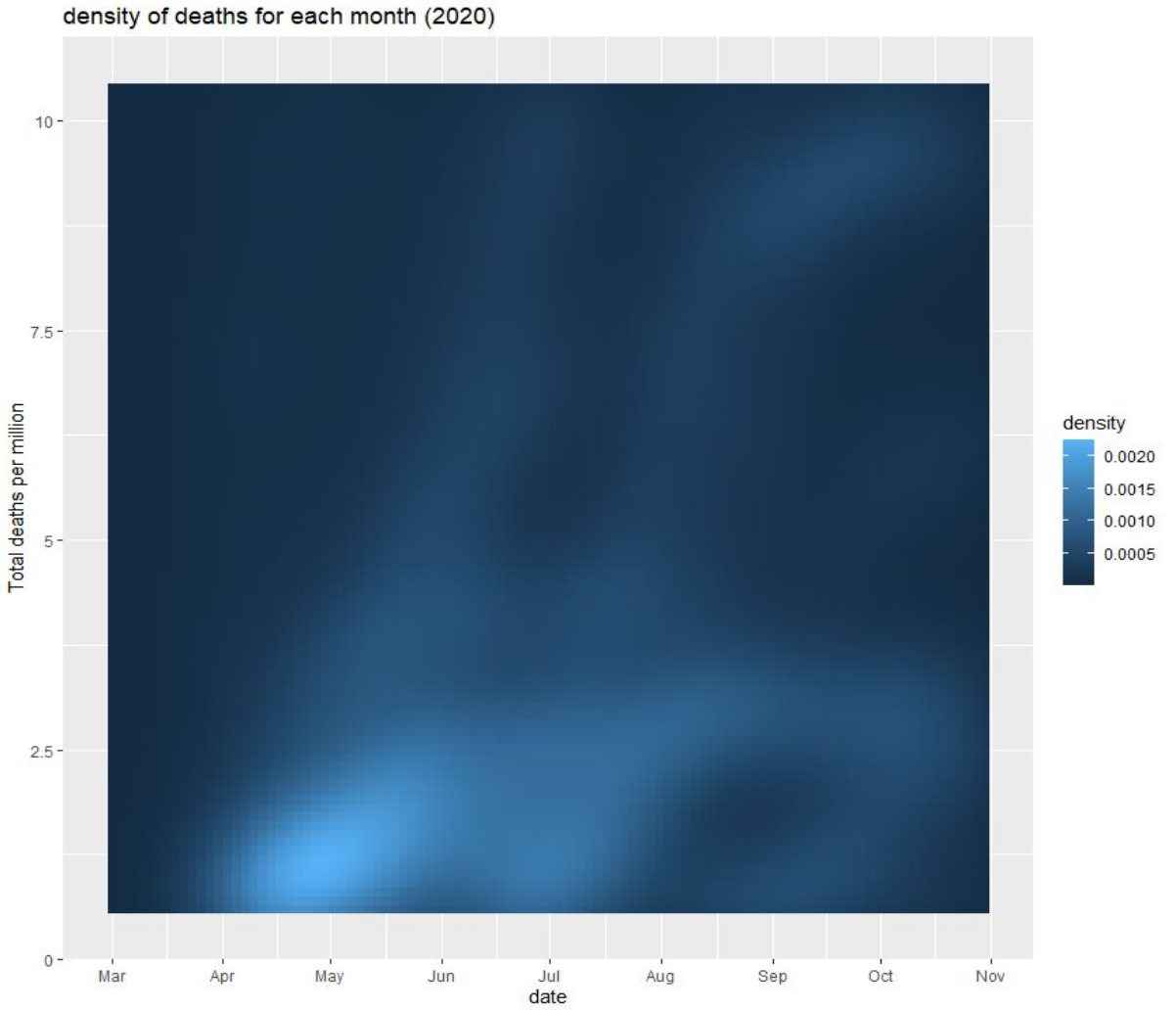
```
addUnits <- function(n) {
  labels <- ifelse(n < 1000, n,  # less than thousands
                   ifelse(n < 1e6, paste0(round(n/1e3), 'k'),  # in thousands
                          ifelse(n < 1e9, paste0(round(n/1e6), 'M'),  # in millions
                                 ifelse(n < 1e12, paste0(round(n/1e9), 'B'), # in billions
                                        ifelse(n < 1e15, paste0(round(n/1e12), 'T'), # in trillions
                                               'too big!'
                                        )))))
  return(labels)
```

```
}
plt1 <- ggplot(covid, aes(x=date, y=Tot.Deaths.P.M))
plt1 +  scale_y_continuous(labels =addUnits, limits = c(0, 50))+
  scale_x_date(date_breaks = "1 month",
               limits = as.Date(c('2020-02-01', '2020-10-30'), format="%d/%m"),
               date_labels="%b" )+
  stat_bin2d(aes(fill=..density..))+ylab("Total deaths per million")+
  ggtitle("density of deaths for each month (2020)")
```



AD2 plot + code

```
plt2 <- ggplot(covid, aes(x=date, y=Tot.Deaths.P.M))
plt2 +  scale_y_continuous(labels =addUnits, limits = c(0.5, 10.5))+
  scale_x_date(date_breaks = "1 month",
               limits = as.Date(c('2020-02-01', '2020-10-30'), format="%d/%m"),
               date_labels="%b" )+
  stat_density2d(aes(fill=..density..),
                 geom="raster",
                 contour=FALSE)+ylab("Total deaths per million")+
  ggtitle("density of deaths for each month (2020)")
```


density of deaths for each month (2020)
Total deaths per million
10
7.5
5
2.5
0
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
date
density
0.0020
0.0015
0.0010
0.0005

GP RStudio Code:

GP Custom Function Used:

```
PC_Analysis = function(df,exclude=NULL,factors=NULL,type="pearson",folder,PCAName){

 subfolder1 = "PrComp"
 subfolder2 = "Principal"

 folder1 = paste("./",folder,sep = "")
 subfolder1 = paste(folder,"/",subfolder1,sep = "")
 subfolder2 = paste(folder,"/",subfolder2,sep = "")


 path1 = paste(subfolder1,"/",sep = "")
 path2 = paste(subfolder2,"/",sep = "")

 pathCreator(path1)
 pathCreator(path2)


 if (is.null(exclude)==FALSE){
  pca_data <- dplyr::select_if(df,is.numeric)
  pca_data <- dplyr::select(pca_data, -exclude)

 }   else {
  pca_data <- dplyr::select_if(df,is.numeric)
 }

 nFactors              <- ncol(pca_data)

 if(type=="spearman") {
  pca_scale <- cor(pca_data,method = "spearman")
 }
 if(type=="pearson") {
  pca_scale <- cor(pca_data,method = "pearson")
 }
 if(type=="kendall") {
  pca_scale <- cor(pca_data,method = "kendall")
 }

 if(is.null(factors)==TRUE){factors=nFactors} else {factors = factors}
 if(is.null(factors)==TRUE){ncols=nFactors} else {ncols = factors}



 # PrComp Variables and Summaries
 pca1_notScaled        <- prcomp(pca_data,scale. = F,rank. = factors)
 pca1_Scaled           <- prcomp(pca_scale,scale. = T,rank. = factors)


 s1 <- pOutput(
  output = "png",
```

```
  plot = plot(pca1_notScaled,main="Unscaled"),
  vName =  paste(PCAName," - Unrotated - Unscaled",sep = ""),
  folder = folder,subfolder = "PrComp")

s2 <- pOutput(
  output = "png",
  plot = plot(pca1_Scaled,main="Scaled") + abline(h=1,col="red"),
  vName =  paste(PCAName," - Unrotated - Scaled",sep = ""),
  folder = folder,subfolder = "PrComp")

s3 <- plot_grid(
  ggdraw() + draw_label("Scree Plots - Unscaled vs Scaled", fontface = 'bold',x = 0,hjust = 0),
  plot_grid(
    ggdraw() + draw_image(s1),
    ggdraw() + draw_image(s2),
    align = "h",nrow = 1,ncol = 2),
  align = "v",ncol = 1,rel_heights = c(.05,1))



# Principal Variables
pca2_notScaled       <- principal(pca_data,rotate = "none",cor = "cov",nfactors = factors )
pca2_Scaled          <- principal(pca_scale,rotate = "none",nfactors = factors )
pca3_notScaled       <- principal(pca_data,rotate = "varimax",cor = "cov",nfactors = factors )
pca3_Scaled          <- principal(pca_scale,rotate = "varimax",nfactors = factors )



principle1.cv.t1 <- kable(round(pca2_notScaled$Vaccounted,2)[1:3,],booktabs=T, caption = "PCA Importance Table Unrotated - Unscaled (Co-variance)",linesep = "\\addlinespace" ) %>% column_spec(1, width = "1in") %>%  column_spec(c(1:ncols+1), width = ".3in") %>% column_spec(2:(factors),color = "black", background = "#5bc0de",bold=T) %>% row_spec(c(1:2),color = "black", background = "white") %>% column_spec(c(1,(factors):ncols),color = "black", background = "white",bold = TRUE) %>% kable_styling(latex_options = "hold_position")
principle1.cr.t2 <- kable(round(pca2_Scaled$Vaccounted,2)[1:3,],booktabs=T, caption = "PCA Importance Table Unrotated - Scaled (Correlation)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width = "1in") %>%  column_spec(c(1:ncols+1), width = ".3in") %>% column_spec(2:(factors),color = "black", background = "#5bc0de",bold=T) %>% row_spec(c(1:2),color = "black", background = "white") %>% column_spec(c(1,(factors):ncols),color = "black", background = "white",bold = TRUE) %>% kable_styling(latex_options = "hold_position")
principle1.cv.t3 <- kable(round(t(pca2_notScaled$loadings[c(1:ncols),c(1:factors)]),2),booktabs=T, caption = "PCA No Rotation Formulas - Unscaled (Co-variance)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width = ".5in") %>%  column_spec(c(1:ncols+1), width = ".4in") %>% kable_styling(latex_options = "hold_position")
principle1.cr.t4 <- kable(round(t(pca2_Scaled$loadings[c(1:ncols),c(1:factors)]),2),booktabs=T, caption = "PCA No Rotation Formulas - Scaled (Correlation)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width = ".5in") %>%  column_spec(c(1:ncols+1), width = ".4in") %>% kable_styling(latex_options = "hold_position")

principle2.cv.t1 <- kable(round(pca3_notScaled$Vaccounted,2)[1:3,],booktabs=T, caption = "PCA Importance Table Rotated - Unscaled (Co-variance)",linesep = "\\addlinespace" ) %>% column_spec(1, width = "1in")
```

```
%>%  column_spec(c(1:ncols+1), width = ".3in") %>% column_spec(2:(factors),color = "black", background =
"#5bc0de",bold=T) %>% row_spec(c(1:2),color = "black", background = "white") %>%
column_spec(c(1,(factors):ncols),color = "black", background = "white",bold = TRUE) %>%
kable_styling(latex_options = "hold_position")
  principle2.cr.t2 <- kable(round(pca3_Scaled$Vaccounted,2)[1:3,],booktabs=T, caption = "PCA Importance
Table Rotated - Scaled (Correlation)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width = "1in") %>%
column_spec(c(+1), width = ".3in") %>% column_spec(2:(factors),color = "black", background =
"#5bc0de",bold=T) %>% row_spec(c(1:2),color = "black", background = "white") %>%
column_spec(c(1,(factors):ncols),color = "black", background = "white",bold = TRUE) %>%
kable_styling(latex_options = "hold_position")
  principle2.cv.t3 <- kable(round(t(pca3_notScaled$loadings[c(1:ncols),c(1:factors)]),2),booktabs=T, caption =
"PCA Rotation Formulas - Unscaled (Co-variance)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width
= ".5in") %>%  column_spec(c(1:ncols+1), width = ".4in") %>% kable_styling(latex_options =
"hold_position")
  principle2.cr.t4 <- kable(round(t(pca3_Scaled$loadings[c(1:ncols),c(1:factors)]),2),booktabs=T, caption =
"PCA Rotation Formulas - Scaled (Correlation)",linesep = "\\addlinespace" ) %>% column_spec(c(1), width =
".5in") %>%  column_spec(c(1:ncols+1), width = ".4in") %>% kable_styling(latex_options = "hold_position")

  PCAs = list()

  PCAs[[1]] <-  pca1_notScaled
  PCAs[[2]] <-  pca1_Scaled
  PCAs[[3]] <-  pca2_notScaled
  PCAs[[4]] <-  pca2_Scaled
  PCAs[[5]] <-  pca3_notScaled
  PCAs[[6]] <-  pca3_Scaled

  names(PCAs) <- c(
    "PrComp-notScaled",
    "PrComp-Scaled",
    "Principal Unrotated - Not Scaled",
    "Principal Unrotated - Scaled",
    "Principal Rotated - Scaled",
    "Principal Rotated - Not Scaled"
  )

  latexTables = list()

  latexTables[1] <- principle1.cv.t1
  latexTables[2] <- principle1.cr.t2
  latexTables[3] <- principle1.cv.t3
  latexTables[4] <- principle1.cr.t4

  latexTables[5] <- principle2.cv.t1
  latexTables[6] <- principle2.cr.t2
  latexTables[7] <- principle2.cv.t3
  latexTables[8] <- principle2.cr.t4

  names(latexTables) <- c(
    "PCA Importance Table Unrotated - Unscaled (Co-variance)",
    "PCA Importance Table Unrotated - Scaled (Correlation)",
```

```
  "PCA No Rotation Formulas - Unscaled (Co-variance)",
  "PCA No Rotation Formulas - Scaled (Correlation)",
  "PCA Importance Table Rotated - Unscaled (Co-variance)",
  "PCA Importance Table Rotated - Scaled (Correlation)",
  "PCA Rotation Formulas - Unscaled (Co-variance)",
  "PCA Rotation Formulas - Scaled (Correlation)"
)

plots = list()
plots[[1]]  <- s3
plots[[2]]  <- PCA_Plot(pca1_notScaled)
plots[[3]]  <- PCA_Plot_Secondary(pca1_notScaled)
plots[[4]]  <- PCA_Plot(pca1_Scaled)
plots[[5]]  <- PCA_Plot_Secondary(pca1_Scaled)

plots[[6]] <- PCA_Plot_Psyc_R(pca3_notScaled)
plots[[7]] <- PCA_Plot_Psyc_Secondary_R(pca3_notScaled)
plots[[8]] <- PCA_Plot_Psyc_R(pca3_Scaled)
plots[[9]] <- PCA_Plot_Psyc_Secondary_R(pca3_Scaled)

plots[[10]] <- PCA_Plot_Psyc(pca2_notScaled)
plots[[11]] <- PCA_Plot_Psyc_Secondary(pca2_notScaled)
plots[[12]] <- PCA_Plot_Psyc(pca2_Scaled)
plots[[13]] <- PCA_Plot_Psyc_Secondary(pca2_Scaled)


  plotNames <- c(
  "Scree plot Scaled vs Unscaled",
  "PCA Plot Unscaled - P1 and P2",
  "PCA Plot Unscaled - P3 and P4",
  "PCA Plot Scaled - P1 and P2",
  "PCA Plot Scaled - P3 and P4",
  "PCA Plot Unscaled and Rotated - P1 and P2",
  "PCA Plot Unscaled and Rotated - P3 and P4",
  "PCA Plot Scaled and Rotated - P1 and P2",
  "PCA Plot Scaled and Rotated - P3 and P4",

  "PCA Plot Unscaled and Unrotated - P1 and P2",
  "PCA Plot Unscaled and Unrotated - P3 and P4",
  "PCA Plot Scaled and Unrotated - P1 and P2",
  "PCA Plot Scaled and Unrotated - P3 and P4"


)

names(plots) <- plotNames

for ( i in seq(1,5,1)){

  png(filename=paste(path1,PCAName," - ",plotNames[i],".png",sep=""),width = 800, height = 800, units =
"px", pointsize = 12, bg = "white", res = NA, restoreConsole = TRUE)
```

```
  print(plots[[i]])
  dev.off()
 }

 for ( i in seq(6,13,1)){

  png(filename=paste(path2,PCAName," - ",plotNames[i],".png",sep=""),width = 800, height = 800, units =
"px", pointsize = 12, bg = "white", res = NA, restoreConsole = TRUE)
  print(plots[[i]])
  dev.off()
 }


 results <- list(PCAs,latexTables,plots)

 names(results) <- c(
  "PCA results",
  "Latex Tables",
  "Plots"
 )
 results

}
```

GP RStudio Data Preprocessing:

```
# Import data
data <- read.table("cleandata_rm.csv", header=TRUE, sep=",")

look_at <- c(
 "continent",
 "location",
 "date",
 "total_cases",
 "new_cases",
 "total_deaths",
 "new_deaths",
 "stry_index",
 "population",
 "pop.index",
 "median_age",
 "aged_65_older",
 "aged_70_older",
 "gdp_per_capita",
 "extreme_poverty",
 "cardio.death.rate",
 "diabet.prev",
 "female_smokers",
 "male_smokers",
 "handwash.fac",
 "hosp.beds.p.thsd",
 "life_expectancy",
 "human.dev.index"
)
rename <- c(
```

```
  "Continent",
  "Date",
  "Location",
  "Total Cases",
  "New Cases",
  "Total Deaths",
  "New Deaths",
  "Stringency Response",
  "Stringency Index",
  "Population",
  "Population Index",
  "Median Age",
  "Age 65 and Over",
  "Age 75 and Over",
  "GDP per Capita",
  "Extreme Poverty",
  "Cardio Death Rate",
  "Diabetes Prevelence",
  "Female Smokers",
  "Male Smokers",
  "Handwashing Facilities",
  "Hospital Beds per Thousand",
  "Life Expectency",
  "Human Development Index",
  "Deaths per Total Cases"
data <- data[look_at]
colnames(data) <- rename[-c(8,25)]
numeric_columns <- as.vector(colnames(select_if(data,is.numeric)))
colnames(data) <- look_at

# Convert to date types
data$date <- as.Date(as_date(data$date,format = '%Y-%m-%d'))

# Change date format
data$date <- as.Date(as.yearmon(data$date,
                    format = "%Y-%m-%d"),
                    format = "%m-%d-%Y")

# Remove special characters
data$continent <- gsub("\n"," " , data$continent ,ignore.case = TRUE)


# Bin the stringency index into 4 parts
data$stry_index_cat <- as.factor(ifelse(
  data$stry_index <= 25, 'No Response' , ifelse(
   data$stry_index > 25 & data$stry_index <= 50 , 'Least Stringent', ifelse(
    data$stry_index > 50 & data$stry_index <= 75 ,'Moderately Stringent',ifelse(
     data$stry_index > 75 ,  'Most Stringent','None')))))

# Convert categorical into integer representations
data$stry_index_cat_int <- as.integer(factor(data$stry_index_cat,
                      levels = c("No Response", "Least Stringent", "Moderately Stringent", "Most Stringent"),
                      ordered = TRUE))


# Copy standardized raw data set
data_raw <- data
```

```
# Pivot and summarize data by 7 days
data <- data %>% group_by(
  continent= continent,
  date=floor_date(date, "7 days"),

  ) %>%
   summarize(
  location = getmode(location),
  total_cases = mean(total_cases),
  new_cases = mean(new_cases),
  total_deaths = mean(total_deaths),
  new_deaths = mean(new_deaths),
  stry_index_cat_int = getmode(stry_index_cat_int),
  stry_index = mean(stry_index),
  population = mean(population),
  pop.index = mean(pop.index),
  median_age = mean(median_age),
  aged_65_older = mean(aged_65_older),
  aged_70_older = mean(aged_70_older),
  gdp_per_capita = mean(gdp_per_capita),
  extreme_poverty = mean(extreme_poverty),
  cardio.death.rate = mean(cardio.death.rate),
  diabet.prev = mean(diabet.prev),
  female_smokers = mean(female_smokers),
  male_smokers = mean(male_smokers),
  handwash.fac = mean(handwash.fac),
  hosp.beds.p.thsd = mean(hosp.beds.p.thsd),
  life_expectancy = mean(life_expectancy),
  human.dev.index = mean(human.dev.index)
)


# Create new variable
data$deaths_per_cases <- (data$total_deaths/data$total_cases) * 100
data_raw$deaths_per_cases <- (data_raw$total_deaths/data_raw$total_cases) * 100

# Manually Transform all numeric variables to log base 10
log_base = 10
data <- data %>% mutate_if(is.numeric,funs(log(.,log_base)))

colnames(data) <- rename
```

GP PCA Analysis:

```
exclude <- c('Total Cases','Total Deaths')

correlations_scaled <- SummaryCorrelations(df=data[numeric_columns],strong=.7,corMethods =
c("pearson","spearman"), name="cor_1_scaled",z=1,folder='plots')

PCA_2 <- PC_Analysis(data[correlations_scaled$spearman$StrongCorr],factors = 4,type='spearman',
folder='plots',PCAName = 'PCA_2',exclude = exclude)

## PCA Analysis
print(PCA_2$`PCA results`$`Principal Rotated - Not Scaled`$loadings,cutoff=.4,sort=T)
```

GP PCA Plot:

```
##############################################################################
# PCA_Plot functions
# Courtesy of Dr. McDonald from DePaul University
##############################################################################
PCA_Plot_Psyc_R = function(pcaData)
{
 library(ggplot2)

 theta = seq(0,2*pi,length.out = 100)
 circle = data.frame(x = cos(theta), y = sin(theta))
 p = ggplot(circle,aes(x,y)) + geom_path()

 loadings = as.data.frame(unclass(pcaData$loadings))
 s = rep(0, ncol(loadings))
 for (i in 1:ncol(loadings))
 {
  s[i] = 0
  for (j in 1:nrow(loadings))
   s[i] = s[i] + loadings[j, i]^2
  s[i] = sqrt(s[i])
 }

 for (i in 1:ncol(loadings))
  loadings[, i] = loadings[, i] / s[i]

 loadings$.names = row.names(loadings)

 p + geom_text(data=loadings, mapping=aes(x = RC1, y = RC2, label = .names, colour = .names,
fontface="bold")) +
  coord_fixed(ratio=1) + labs(x = "PC1", y = "PC2",title = pcatitles$title5,subtitle = pcasubtitles$subtitle5) +
guides(fill=guide_legend(title="Variables"))
}
```

GP Mosaic Plot

```
```{r p1, include = TRUE, echo = FALSE, fig.align='center',fig.width= 8,fig.height= 8,fig.cap="Covid 19 - Stringency Level vs Continent",warning=FALSE}

fill_1 <-   scale_fill_brewer(
            name="Stringency\nResponse\nLevel",
            direction = 1,
            palette = pallets$qualitative[7])

p1 <- ggplot(data = data_raw) +
     geom_mosaic(aes(x=product(stry_index,continent),fill=stry_index,size=3)) +

     annotate(
     geom="text", x=.225,y=-0.05,
     label="Africa", color="black", size=5) +
     annotate(
     geom="text", x=.59,y=-0.05,
     label="Asia", color="black", size=5) +
     annotate(
     geom="text", x=.75,y=-0.05,
     label="Europe", color="black", size=5) +
     annotate(
     geom="text", x=.85,y=-0.05,
     label="North\nAmerica", color="black", size=5) +
     annotate(
     geom="text", x=.97,y=-0.05,
     label="South\nAmerica", color="black", size=5) +
  labs(fill='Stringency Index',title = titles$title1, subtitle = subtitles$subtitle1) + theme(legend.position = 'bottom',axis.title = element_text(size = 7)) + fill_1
```

GP Animated Ridge Distribution:

```
```{r p2, include = TRUE, echo = FALSE, fig.align='center',fig.width= 8,fig.height= 7,fig.cap="Distributions by Stringency Level and Continent",warning=FALSE}
sec_breaks <- c(10,100, 1000, 10000, 100000, 1000000)

p2 <- ggplot(data=data_raw, aes(y = date2, x = new_cases,fill=stry_index)) +
     geom_density_ridges() +
     scale_x_continuous(
       sec.axis = sec_axis(trans = ~10^.,
       breaks = sec_breaks,
       labels = format(sec_breaks, big.mark=",", scientific=FALSE),
       name = 'New Cases Per Week - Exponential scale')) +
     labs(title = titles$title2, subtitle = subtitles$subtitle2,fill='Stringency Index') + ylab('Date') +
     xlab('New Cases Per Week - Logorithmic scale (Log base 10)') + theme(legend.position = 'bottom') + fill_1


p2_animated <- p2 + transition_time(date2) + labs(subtitle = subtitles$subtitle2a)
animate(p2_animated,fps = 20)
anim_save(paste(titles$title2,'gif',sep = '.'))
```

GP Layered LOESS by Time, Continent and Stringency Level

```{r p3, include = TRUE, echo = FALSE, fig.align='center',fig.width= 8,fig.height= 10,fig.cap="Smoothed LOESS by Continent and Stringency Level ",warning=FALSE}

color_2 <-   scale_color_brewer(
            name="Continents",
            direction = 1,
            palette = pallets$qualitative[2])


lwd = 2.25
p3 <- ggplot() +
  geom_smooth(data=data[which(data$continent=='Africa'),],aes(x=date,y=deaths_per_cases,linetype
=stry_index,color=continent),lwd =lwd) +
geom_smooth(data=data[which(data$continent=='Asia'),],aes(x=date,y=deaths_per_cases,linetype
=stry_index,color=continent),lwd =lwd)  +
  geom_smooth(data=data[which(data$continent=='Europe'),],aes(x=date,y=deaths_per_cases,linetype
=stry_index,color=continent),lwd =lwd)+
  geom_smooth(data=data[which(data$continent=='North America'),],aes(x=date,y=deaths_per_cases,linetype
=stry_index,color=continent),lwd =lwd)  + geom_smooth(data=data[which(data$continent=='South
America'),],aes(x=date,y=deaths_per_cases,linetype =stry_index,color=continent),lwd =lwd) +
  scale_x_date(date_breaks = '1 month') +
  scale_y_continuous(n.breaks = 15) + theme(legend.position = 'bottom') +
  labs(title = titles$title3,subtitle = subtitles$subtitle3,linetype='Stringency Response') +
  ylab('Weekly Deaths Per Weekly Total Cases (%)') +
  xlab('Date') + theme(axis.text.x.bottom = element_text(angle = 45,hjust = 1),legend.position = 'left') + color_2
```

Table of attributes

Attribute	Description
iso_code	ISO 3166-1 alpha-3 – three-letter country codes
continent	Continent of the geographical location
location	Geographical location
date	Date of observation
total_cases	Total confirmed cases of COVID-19
new_cases	New confirmed cases of COVID-19
new_cases_smoothed	New confirmed cases of COVID-19 (7-day smoothed)
total_deaths	Total deaths attributed to COVID-19
new_deaths	New deaths attributed to COVID-19
new_deaths_smoothed	New deaths attributed to COVID-19 (7-day smoothed)
total_cases_per_million	Total confirmed cases of COVID-19 per 1,000,000 people
new_cases_per_million	New confirmed cases of COVID-19 per 1,000,000 people
new_cases_smoothed_per_million	New confirmed cases of COVID-19 (7-day smoothed) per 1,000,000 people

total_deaths_per_million	Total deaths attributed to COVID-19 per 1,000,000 people
new_deaths_per_million	New deaths attributed to COVID-19 per 1,000,000 people
new_deaths_smoothed_per_million	New deaths attributed to COVID-19 (7-day smoothed) per 1,000,000 people
tests_units	Units used by the location to report its testing data
stringency_index	Government Response Stringency Index: composite measure based on 9 response indicators including school closures, workplace closures, and travel bans, rescaled to a value from 0 to 100 (100 = strictest response)
population	Population in 2020
population_density	Number of people divided by land area, measured in square kilometers, most recent year available
median_age	Median age of the population, UN projection for 2020
gdp_per_capita	Gross domestic product at purchasing power parity (constant 2011 international dollars), most recent year available
cardiovasc_death_rate	Death rate from cardiovascular disease in 2017 (annual number of deaths per 100,000 people)

diabetes_prevalence	Diabetes prevalence (% of population aged 20 to 79) in 2017
life_expectancy	Life expectancy at birth in 2019
human_development_index	Summary measure of average achievement in key dimensions of human development: a long and healthy life, being knowledgeable and have a decent standard of living