****

**Visualization of Global Competitiveness Report**

**B9DA106 Data Visualisation**

**Basel Magableh**

**Brijith Elizabeth Joseph**

**10393634**

**Introduction**

The Global Competitiveness Report (GCR) is a yearly report published by the World Economic Forum. the Global Competitiveness Report ranks countries based on the Global Competitiveness Index. It is developed by Xavier Sala-i-Martin and Elsa V. Artadi in 2004. The report assesses the ability of countries to provide high levels of prosperity to their citizens. This in turn depends on how productively a country uses available resources. Therefore, the Global Competitiveness Index measures the set of institutions, policies, and factors that set the sustainable current and medium-term levels of economic prosperity.Report assesses the competitiveness landscape of 137 economies

The report has twelve pillars of competitiveness.

They are:

* Institutions
* Appropriate infrastructure
* Stable macroeconomic framework
* Good health and primary education
* Higher education and training
* Efficient goods markets
* Efficient labor markets
* Developed financial markets
* Ability to harness existing technology
* Market size—both domestic and international
* Production of new and different goods using the most sophisticated production processes
* Innovation

**Visualizations**

We have made nine visualizations including 3 svgs using the GCI data. The data fields that were taken into consideration are Country, Year, 1st pillar Institutions, 2nd pillar Infrastructure, 3rd pillar Macroeconomic environment, 4th pillar Health and primary education,5th pillar Higher education and training,6th pillar Goods market efficiency,7th pillar Labor market efficiency, 8th pillar Financial market development,9th pillar Technological readiness,10th pillar Market size,11th pillar Business sophistication ,12th pillar Innovation, Global Competitiveness Index, Population, GDP, Income group, Region, Forum classification. The report contains the data from 2007 - 2017

We have used three different tools.

* D3.js
* Python
* R Shiny

**Visualization 1**

**Tool :D3.JS**

Hans Rosling is a data visualization legend. His 2006 TED talk, The Best Stats You’ve Ever Seen, is one of the most viewed videos on the TED website (<http://bit.ly/2doLzAY>). We were highly inspired by his work and we have createda visualization as homage to him.

SVG stands for **Scalable Vector Graphics**. SVG is an XML-based vector graphics format. It provides options to draw different shapes such as Lines, Rectangles, Circles, Ellipses, etc. Hence, designing visualizations with SVG gives you more power and flexibility compared to other visualization tools like tableau.

The First Visualization consists of three SVG’s. The first one is a bubble plot. Second and third are bar plots. Second and third svgs will get activated only after selecting a country from 1st plot.

First svg is to select a country. It gives the GCI vs GDP plot . The user can get the data from 2007-2017. User can see all together using the play all button or can select a particular year and country

Second svg is to display the pillars of that selected country in a particular year.The pillar information displayed in this Svg is dependent upon the country and year selected in the previous svg. If the customer is selecting ‘play all’ the pillar values of 2007 to 2017 of the selected country will be displayed

Third SVG is for comparing a particular pillar of a country with the selected country in svg 1

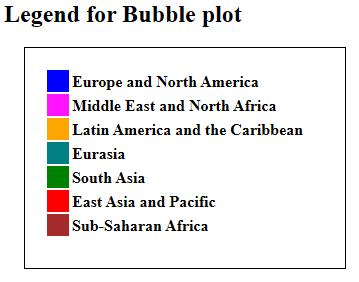
A video has been captured and has been uploaded in YouTube.

The link is [**https://youtu.be/8EzydiN7y9s**](https://youtu.be/8EzydiN7y9s)

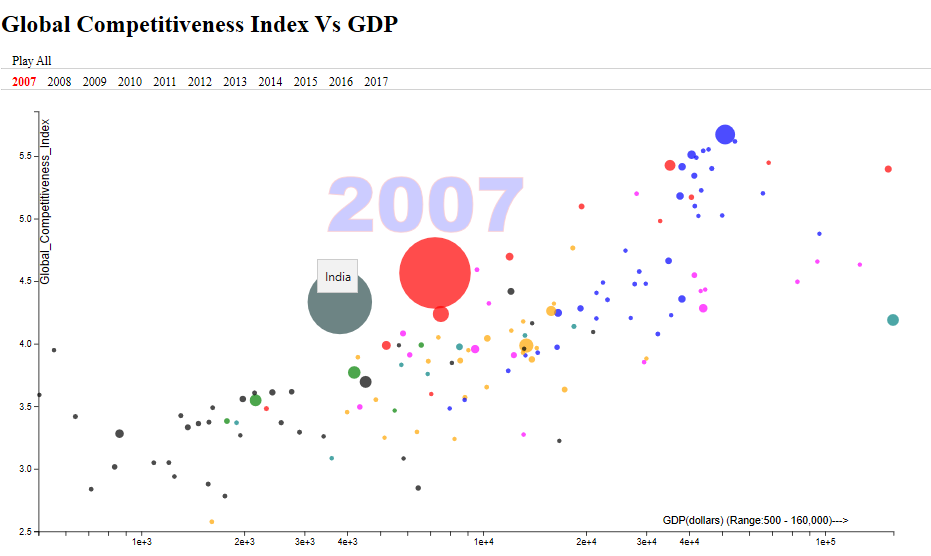
The github url is The url is <https://pallavije.github.io/>

**1st SVG:**

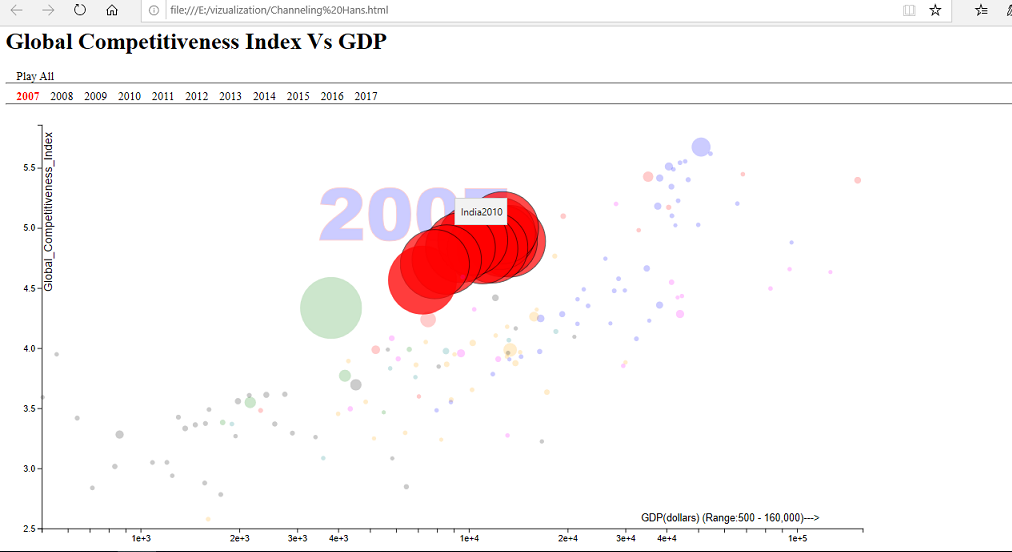
* Bubble plot **GCI** VS **GDP** of 137 countries
* A clear bubble chart with’ Play All’function and provision to view data based on particular year.
* Click on any bubble to invoke TRACE functionality.
* The countries are colored based on their region
* The size of the bubble represents the population
* When a country is selected, it will be highlighted along with its trace from 2007 to 2017



**Fig 1. Legend of the bubble Plot**



**Fig 2. SVG 1 result Bubble plot – GCI Vs GDP**



**Fig 3. SVG 1 result:Trace of india from 2007-17(selected bubble- india in 2010)**

**2nd SVG:**

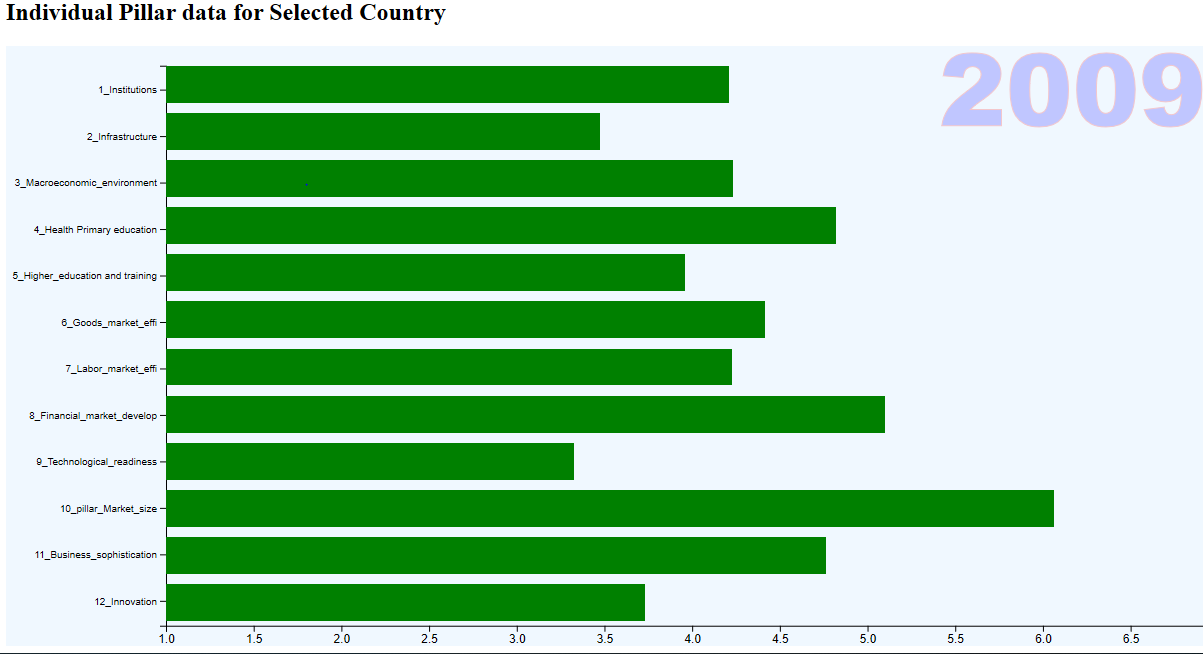
-Activates when a Country bubble is selected from 1st SVG

-Dynamic and changes with interactions in 1st SVG. Also supports "Play All" functionality (on the top ).

-This is a horizontal bar plot(pillars VS value). The 11 pillar values of the selected country (in first svg) will be displayed here. If the user is selecting the play all button the pillars of the country from 20007 to 2017 will be displayed as series or else if a particular year is selected, the pillars for that particular year of the selected country will be displayed



**Fig 4. Svg 2 result before selecting a country**

** fig 5svg2 result :selected country-India**

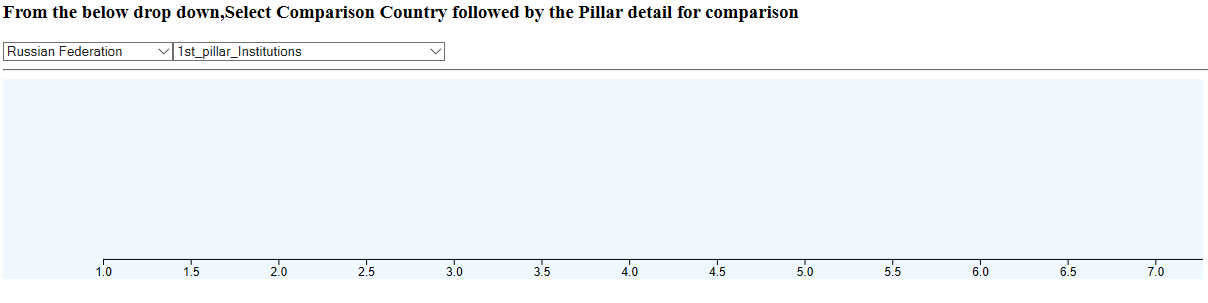
**3rd SVG(Comparion of pillar):**

-Activates after displaying 2nd SVG.

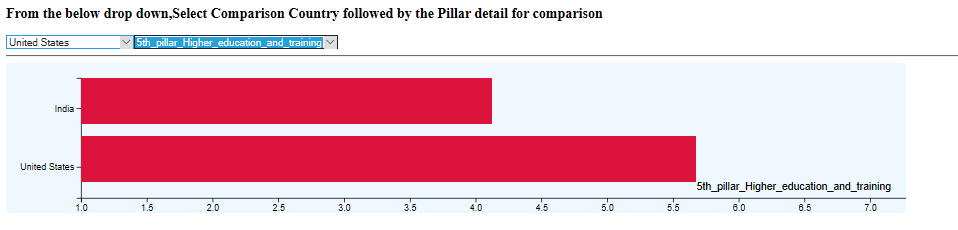
-Need to select Country and Pillar from drop down provided for comparison.

-Comparison is done between the First Country, selected from Bubble chart and the country selected in drop down list based on Pillar Condition selected in the second drop down.

-Comparison can be done multiple number of times by changing vales in Drop down(Pillar detail has to be selected(or reselected, if no change) to generate new comparison).



**Figure 6: SVG 3 result Before selecting Country**



**Figure 7: SVG 3 result Comparison of 5th pillar between India and US**

. The webpage it is hosted in github. The url is <https://pallavije.github.io/>

Source code : <https://github.com/pallavije/pallavije.github.io>

**Visualization 2**

**Tool:Python**

In total we have done four visualizations in python. And code for each visualization is shown below apart from deploying it on GitHub.

Below is the code from which we loaded the essential python libraries that were needed for visualization.

*import matplotlib.pyplot as plt*

*import pandas as pd*

*import seaborn as sns*

*sns.set()*

*import numpy as np*

For reading the csv file of the dataset the below script was used.

*dfg = pd.read\_csv('E:\Datasets\GCI\_CompleteData4.csv')*

*dfg*

where ‘dfg’ is a variable defined to input the dataset as a dataframe.

So, coming to the visualization, below is the code for it.

*bplot = sns.boxplot(y='Region', x='Population',*

*data=dfg,*

*width=0.5,*

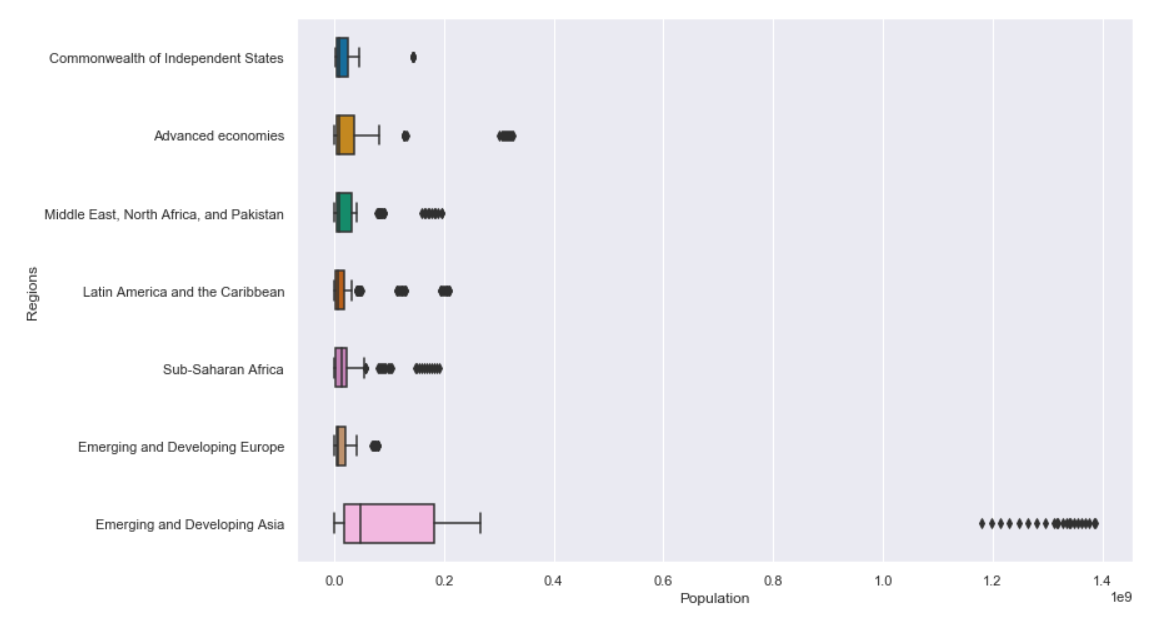
*palette="colorblind")*

*plt.xlabel('Population')*

*plt.ylabel('Regions')*

*plt.rcParams['figure.figsize'] = (12, 8)*

It is a boxplot showing the population in different regions.



The X-axis here is Population and the Y-axis displays the 7 different Regions. Fromthis plot it can be clearly seen the Emerging and Developing Asia has the highest population whereas Emerging and Developing Europe has the least population. There is a large range of population gap between the region with highest population and the regions with low population (i.e., the remaining 6 regions), we can see that clearly by looking at the upper limits of each region’s boxplot.

**Visualization 3**

**Tool: Python**

This Visualization is a Density Plot. Below is the script used to plot this.

*sns.kdeplot(dfg['Global\_Competitiveness\_Index'], shade=True, bw=0.8, color="black")*

*sns.kdeplot(dfg['1st\_pillar\_Institutions'], shade=True, bw=0.8, color="orange")*

*sns.kdeplot(dfg['2nd\_pillar\_Infrastructure'], shade=True, bw=0.8, color="red")*

*sns.kdeplot(dfg['3rd\_pillar\_Macroeconomic\_environment'], shade=True, bw=0.8, color="blue")*

*sns.kdeplot(dfg['4th\_pillar\_Health\_and\_primary\_education'], shade=True, bw=0.8, color="yellow")*

*sns.kdeplot(dfg['5th\_pillar\_Higher\_education\_and\_training'], shade=True, bw=0.8, color="brown")*

*sns.kdeplot(dfg['6th\_pillar\_Goods\_market\_efficiency'], shade=True, bw=0.8, color="purple")*

*sns.kdeplot(dfg['7th\_pillar\_Labor\_market\_efficiency'], shade=True, bw=0.8, color="white")*

*sns.kdeplot(dfg['8th\_pillar\_Financial\_market\_development'], shade=True, bw=0.8, color="lightblue")*

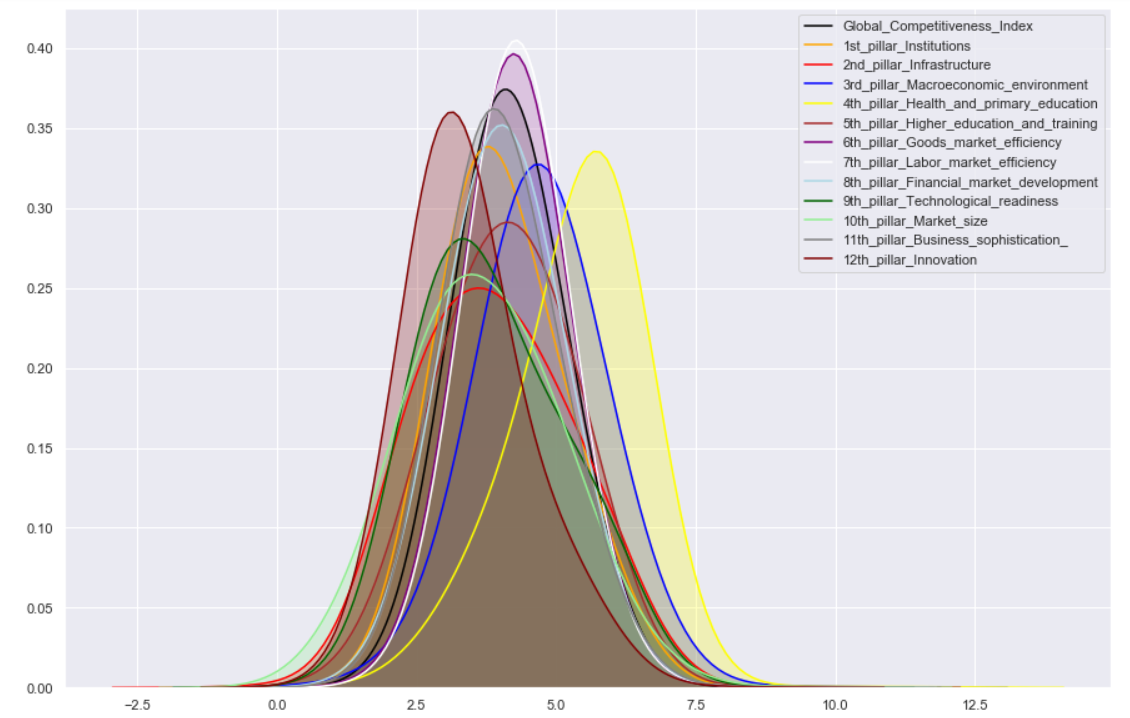
*sns.kdeplot(dfg['9th\_pillar\_Technological\_readiness'], shade=True, bw=0.8, color="darkgreen")*

*sns.kdeplot(dfg['10th\_pillar\_Market\_size'], shade=True, bw=0.8, color="lightgreen")*

*sns.kdeplot(dfg['11th\_pillar\_Business\_sophistication\_'], shade=True, bw=0.8, color="gray")*

*sns.kdeplot(dfg['12th\_pillar\_Innovation'], shade=True, bw=0.8, color="maroon")*

The idea of this plot is to compare the 12 pillars of the GCI data with the main Global Competitive Index. This plot is shown below.

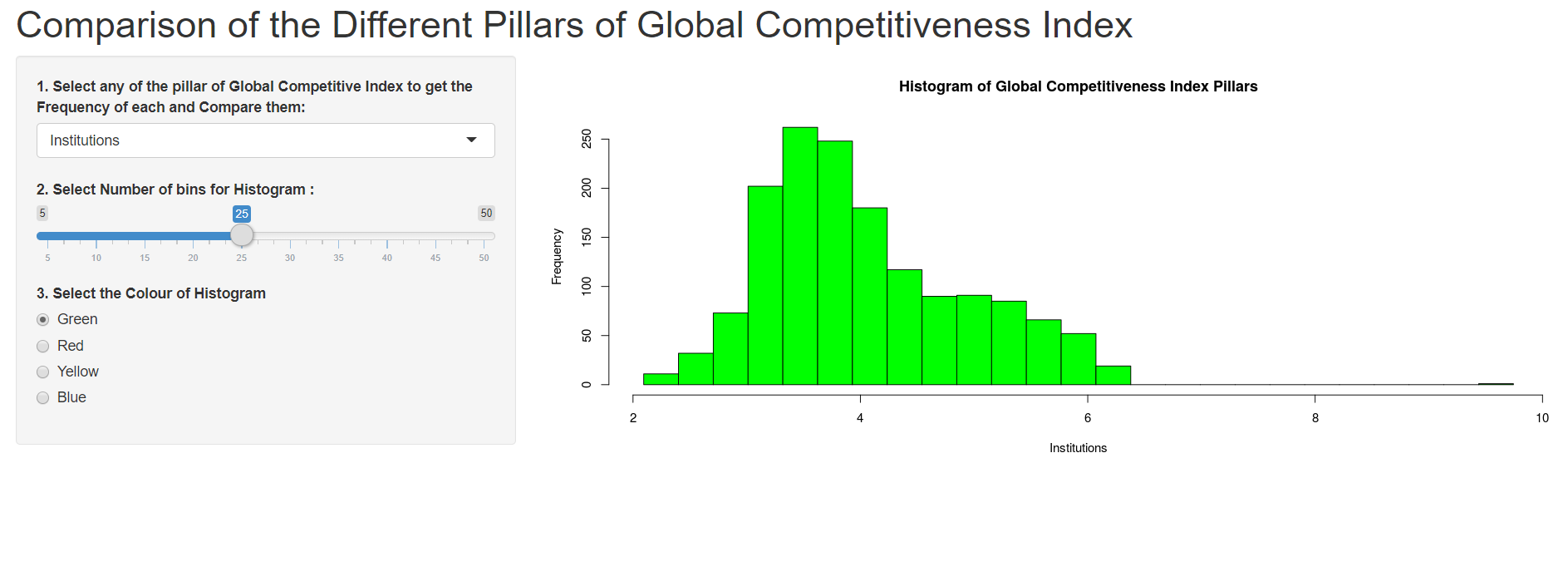


The Y-axis here shows the range of ratings of all the pillars and the X-axis is range of normal distribution curve and it varies as per the bandwidth value provided. There are total 13 different density plots overlapping each other and each of them are represented by different colors. From this plot it can be seen that the goods market efficiency(purple) and labor market efficiency(white) are exceeding the global competitive index standard range.

**Visualization 4**

**Tool: R Shiny**

R shiny is a package in R through which we can make interactive and automated plots and these plots works like a mini application which can be deployed on the web. Coming to the visualization it’s a R shiny application which shows histograms for each pillar of the GCI dataset to compare frequency of each pillar with other. This app has a slider input for changing the number of bins of histograms for better understanding. In addition to this, there is a radio button panel to change the color of histogram. Below is one of the screenshot of the visualization app.



This R shiny app has been deployed on the R shiny web server and it can be accessed from the below link:

<https://ahmedkhanak.shinyapps.io/dataviz1/>

This app is a combination of two R scripts namely UI.R and SERVER.R. Codes for each are given below.

**UI.R**

*shinyServer(*

*pageWithSidebar(*

*headerPanel( "Comparison of the Different Pillars of Global Competitiveness Index"),*

*sidebarPanel(*

*selectInput("gdp", "1. Select any of the pillar of Global Competitive Index to get the Frequency of each and Compare them:",*

*choices = c("Institutions"= 1, "Infrastructure"= 2, "Macroeconomic Environment"= 3, "Health and Primary Education"= 4,*

*"Higher Education and Training"= 5, "Goods Market Efficiency"= 6, "Labor Market Efficiency"= 7,*

*"Financial Market Development"= 8, "Technological Readiness"= 9, "Market Size"= 10,*

*"Business Sophistication"= 11, "Innovation"= 12)),*

*sliderInput("bins",*

*"2. Select Number of bins for Histogram :",*

*min = 5, max = 50, value = 30),*

*radioButtons("color", "3. Select the Colour of Histogram", choices = c("Green", "Red", "Yellow", "Blue"), selected = "Green")*

*),*

*mainPanel(*

*plotOutput("myPlot")*

*) ))*

**SERVER.R**

*function(input, output, session) {*

*gdp=read.csv("GCI\_CompleteData4.csv")*

*df = data.frame(gdp$X1st\_pillar\_Institutions, gdp$X2nd\_pillar\_Infrastructure,*

*gdp$X3rd\_pillar\_Macroeconomic\_environment,*

*gdp$X4th\_pillar\_Health\_and\_primary\_education,*

*gdp$X5th\_pillar\_Higher\_education\_and\_training, gdp$X6th\_pillar\_Goods\_market\_efficiency,*

*gdp$X7th\_pillar\_Labor\_market\_efficiency,*

*gdp$X8th\_pillar\_Financial\_market\_development, gdp$X9th\_pillar\_Technological\_readiness,*

*gdp$X10th\_pillar\_Market\_size,*

*gdp$X11th\_pillar\_Business\_sophistication\_, gdp$X12th\_pillar\_Innovation)*

*names(df) = c("Institutions", "Infrastructure", "Macroeconomic Environment", "Health and Primary Education",*

*"Higher Education and Training", "Goods Market Efficiency", "Labor Market Efficiency",*

*"Financial Market Development", "Technological Readiness", "Market Size",*

*"Business Sophistication", "Innovation")*

*output$myPlot = renderPlot(*

*{*

*colm = as.numeric(input$gdp)*

*hist(df[,colm], breaks = seq(min(df[,colm]), max(df[,colm]), length.out = input$bins+1), col = input$color,*

*main = "Histogram of Global Competitiveness Index Pillars", xlab = names(df[colm]))*

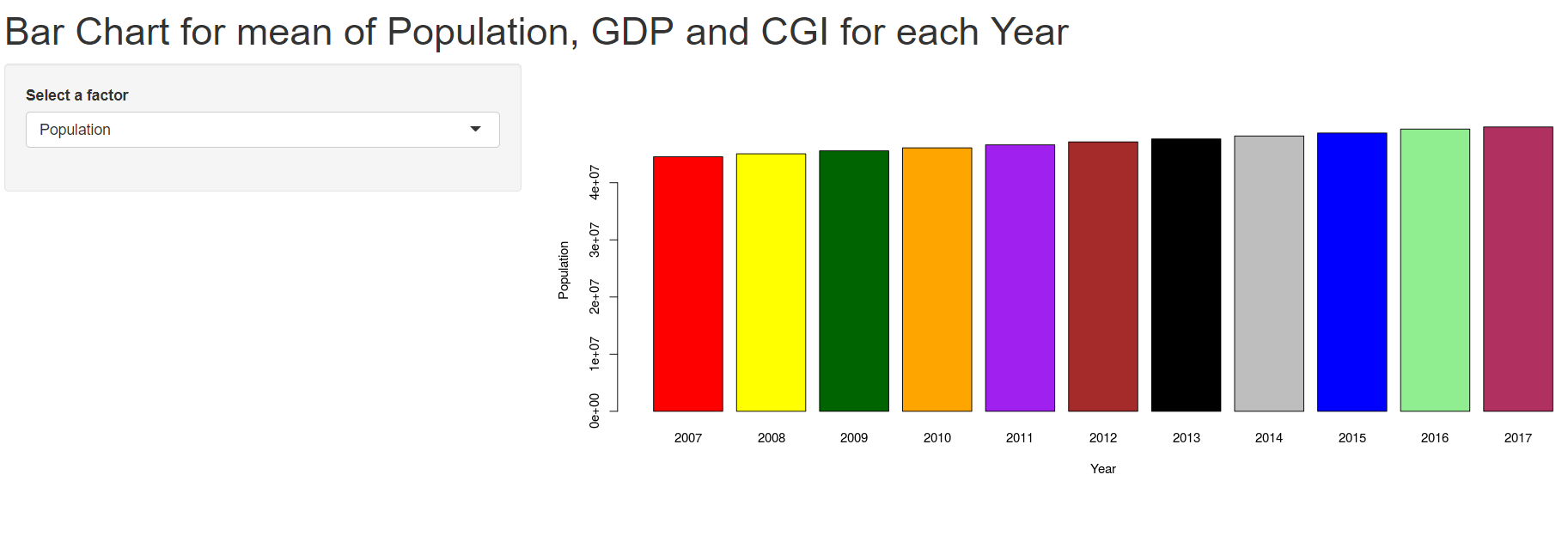
*}*

*) })*

**Visualization 5**

**Tool: R Shiny**

This visualization is also a R shiny based application. It is a bar plot displaying the comparison of Global competitive index (GCI), Population and GDP from 2007 to 2017. Each year is presented with different colored bar. And measures the mean of GCI, Population and GDP using aggregate function to average up all the data of one variable to plot a separate bar for each year. Below is the screenshot of the visualization app.



As seen above Y-axis is the factor/variable/column to be selected with the drop down select inputand X-axis is the Year. It can be observed from the above screenshot, the population goes on increasing from year 2007 to 2017. This visualization app is deployed on the R shiny web server and can be accessed from the below link:

<https://ahmedkhanak.shinyapps.io/dataviz2/>

This also has two files namely UI.R and SERVER.R. Code for each is given below.

**UI.R**

*shinyServer(*

*pageWithSidebar(*

*headerPanel( "Bar Chart for mean of Population, GDP and CGI for each Year"),*

*sidebarPanel(*

*selectInput("gpg", "Select a factor", choices = names(gdp[,15:17]))*

*),*

*mainPanel(*

*plotOutput("barchart")*

*) ))*

**SERVER.R**

*shinyServer(*

*function(input, output, session) {*

*gdp=read.csv("GCI\_CompleteData4.csv")*

*output$barchart = renderPlot({*

*c = aggregate(gdp[,input$gpg] ~ Year, gdp, mean)*

*barplot(c[,2], names.arg = c$Year, xlab = "Year", ylab = names(gdp[input$gpg]), col = c("Red", "Yellow", "darkgreen", "Orange", "Purple", "Brown", "Black", "Gray", "Blue", "lightgreen", "maroon"))*

*}) })*

UI.R is for developing the user interface of the app and the SERVER.R decides how the app is going to work and what should be the output based on the inputs.

**Visualization 6**

**Tool: Python**

In total we have done four visualizations in python. And code for each visualization is shown below apart from deploying it on GitHub.

Below is the code from which we loaded the essential python libraries that were needed for visualization

Matplotlib and bokeh are the most popular and user friendly package for data visualization projects.

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from \_\_future\_\_ import division

from plotly.offline import init\_notebook\_mode, iplot

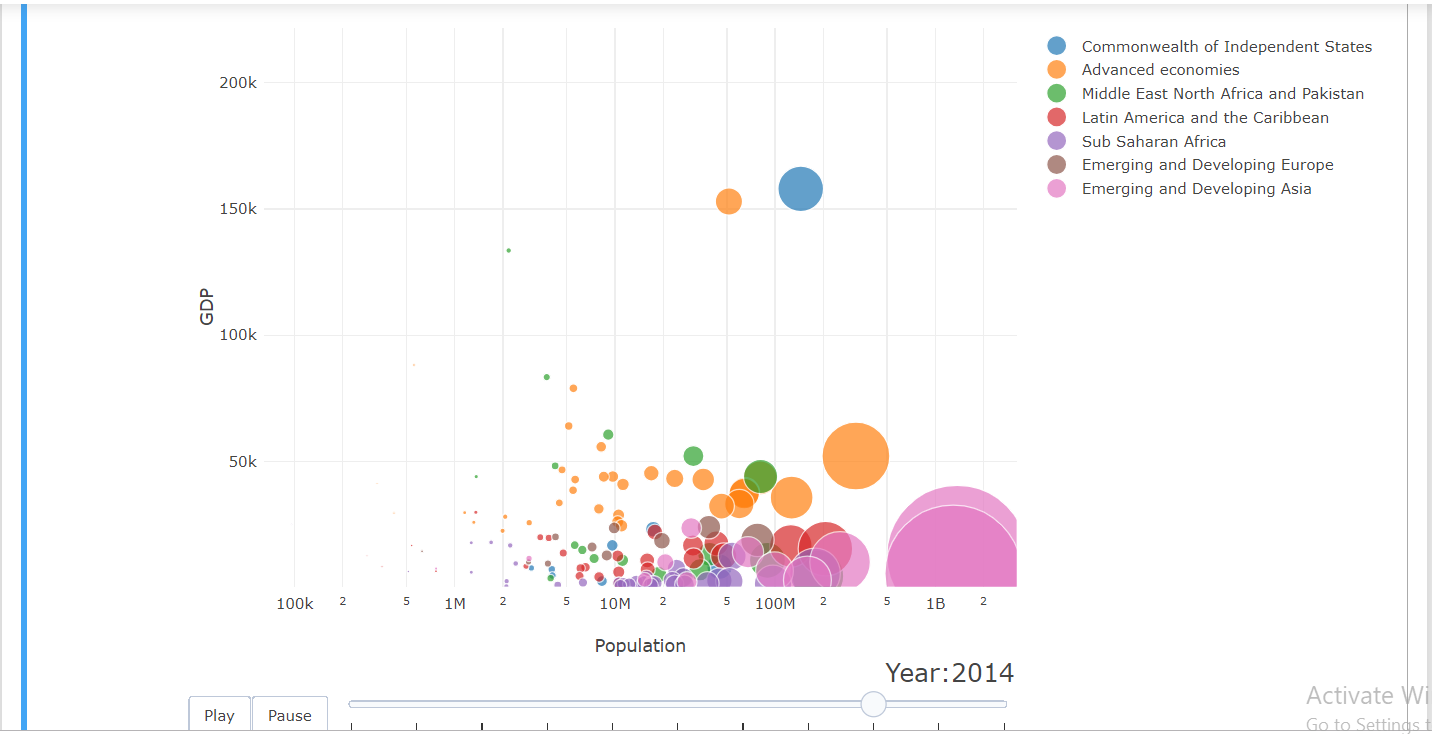
init\_notebook\_mode()

from bubbly.bubbly import bubbleplot

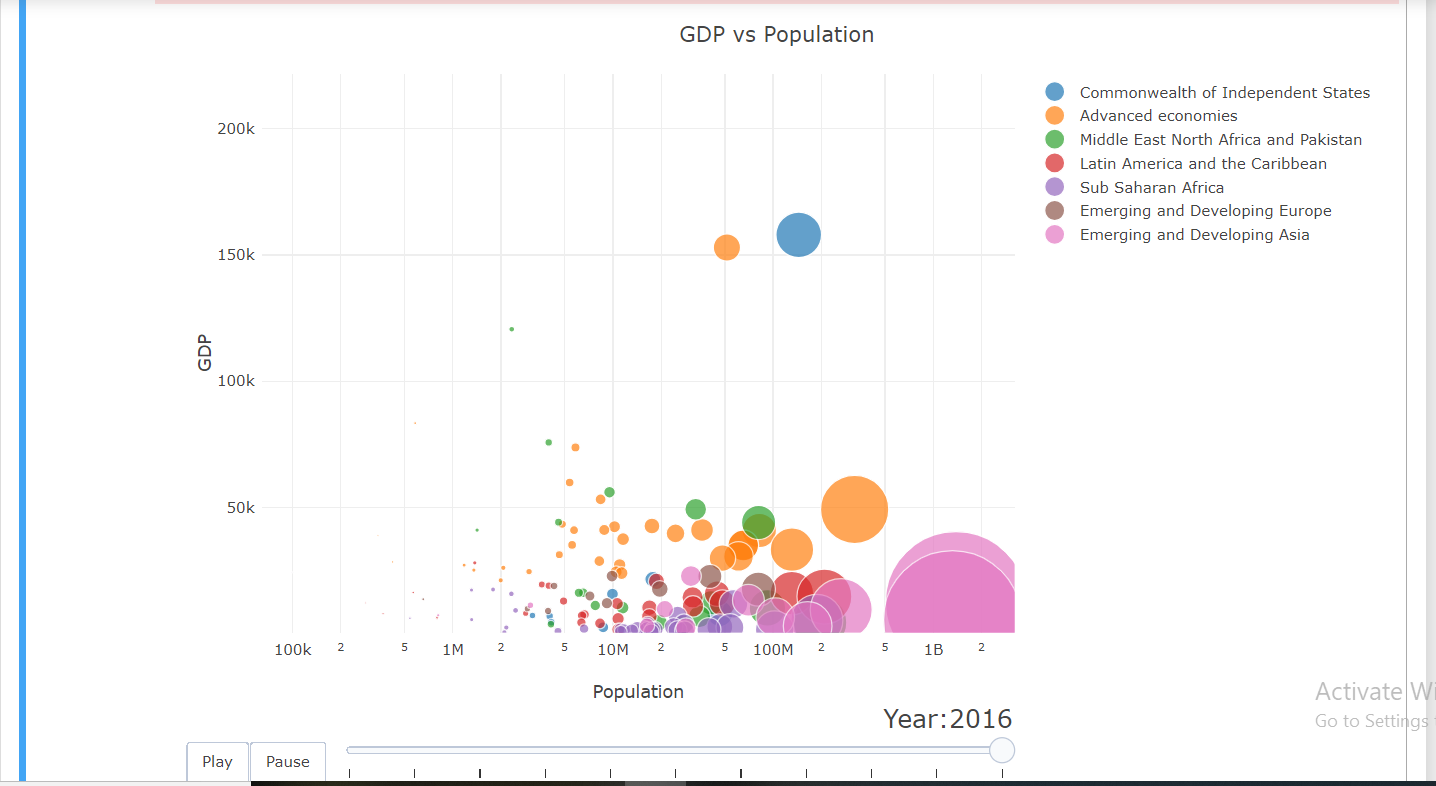
For reading the csv file of the dataset the below script was used.

data = pd.read\_csv("C:/Users/hp/Desktop/2nd Semester Assignments/Data Visualization/GCI\_Data.csv", delimiter=',')

We are visualizing 4 specific variables like, GDP, Population, Region, Year. This graph is an interactive graph and we added year as a input parameter and based on the year input the visualization changes automatically when we press the play button.



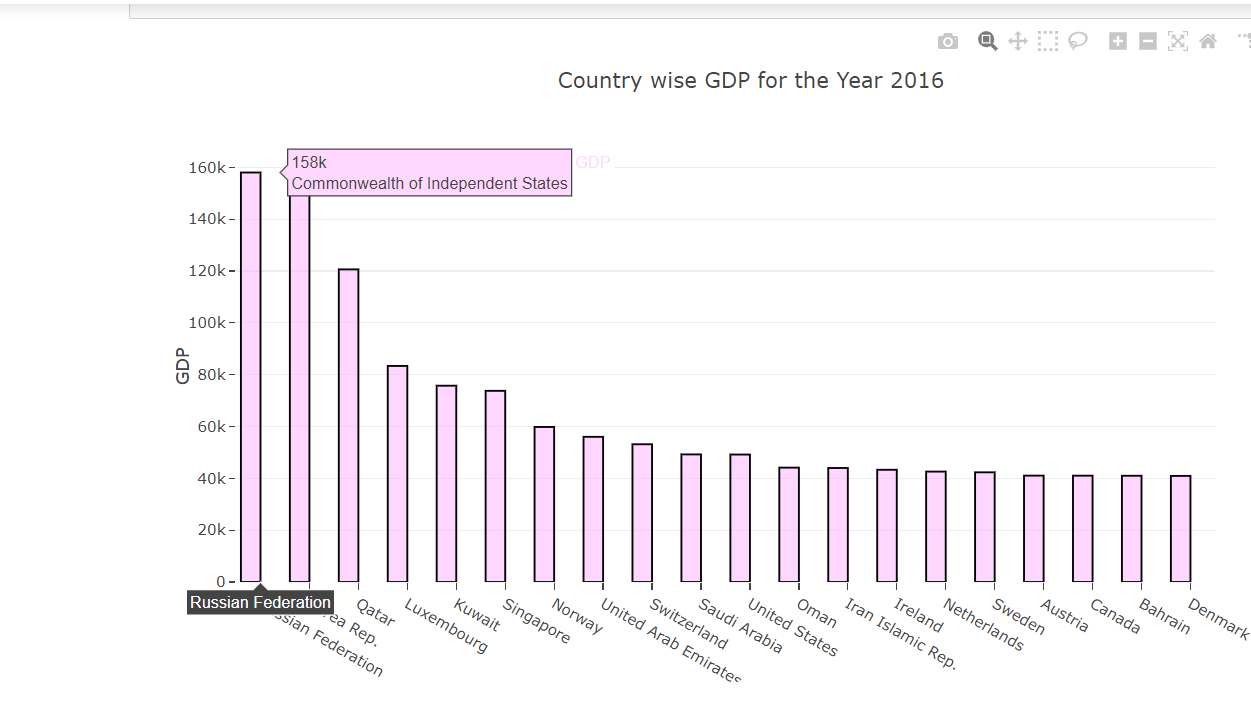
When we move the slide from left to right the graph interact dynamically



**Visualization 7**

**Tool: Python**

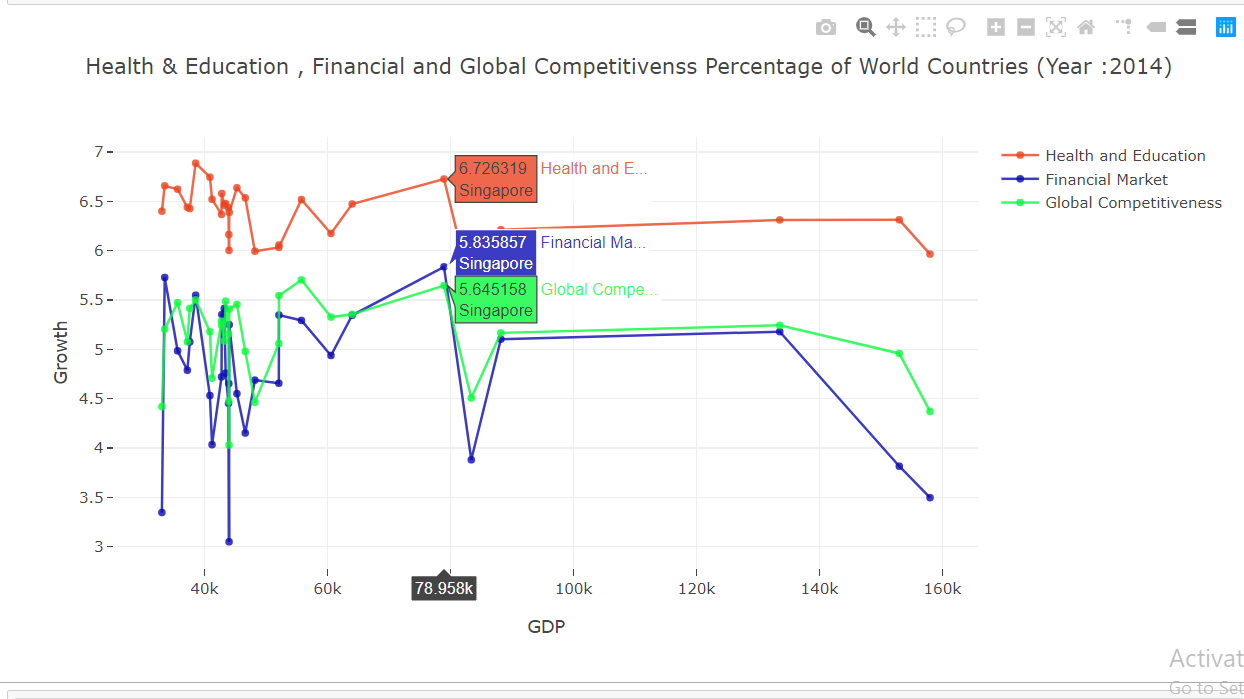
In this visualization we are considering the GDP rate for top 20 countries. Russian federation has height GDP in this top 20 and Denmark has lowest .

****

**Visualization 8**

**Tool: Python**

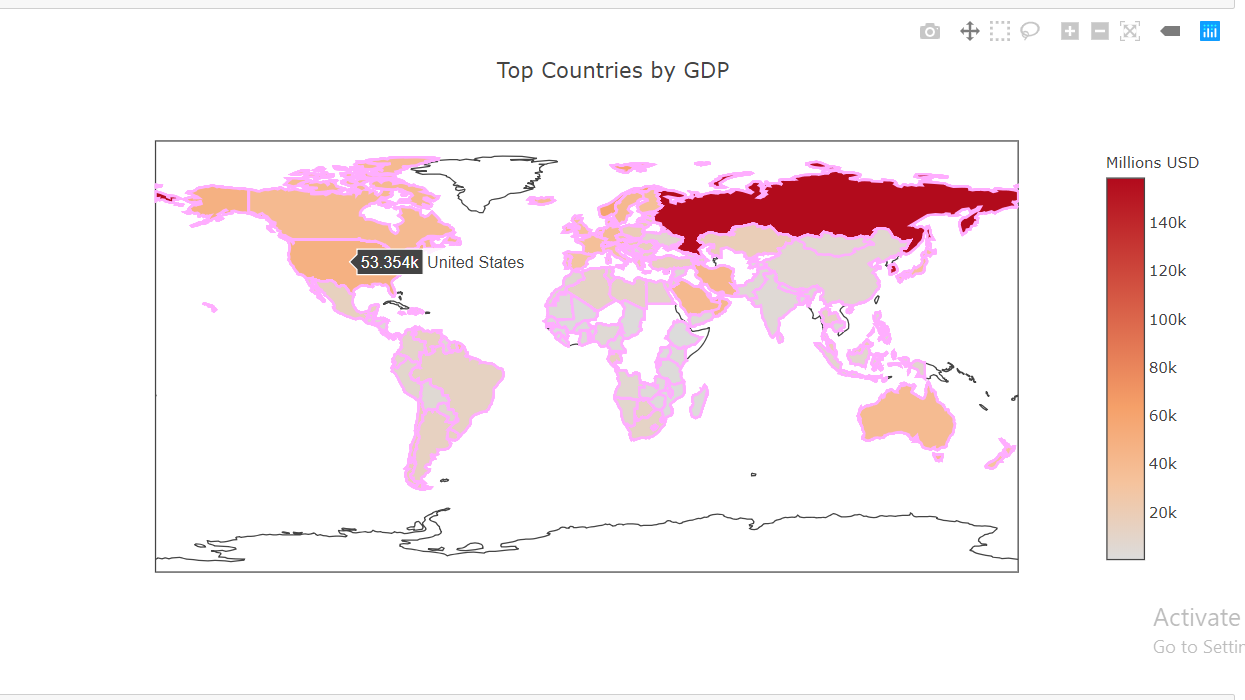
We are visualizing growth of health and education, financial market and GCI in a line graph. The graph is interactive when we click on the line, its shows the details

****

**Visualization 9**

**Tool: Python**

We are visualizing the GDP for different countries. The countries are colored according to there GDP. The scale is also mentioned in the graph



This visualization is done by Python and can be accessed from the below link:

<https://github.com/Jay0421/Visualization.of.Global.Competitiveness>

**Project Development Cycle.**

First we selected global terrorism data but because of the lack of numerical values, the planned to select a new topic. Since we clearly knew the shortcomings on the previous dataset, we were able to choose out next dataset ‘Global Competitive report’. First we thought of implementing in python. Later we decided to do in three different tools so that each member can implement their ideas fully and learn it in detail.

For three of us, It was completely new experience

**I,Brijith Elizabeth Joseph** was really fascinated by the TED talk ‘The best stats you've ever seen’ by Hans Rosling. i wanted to make a Dynamic Chart and I choose D3.js .My past experience in web development and Java script made the task easier for me.

Even though I had experience in web development. I was completely new to d3.js and SVG. I referred a website (<https://www.tutorialsteacher.com/d3js>) for understanding the basics. It took me 2 days to understand it. The development was done alone with the demo examples.

Maintaining the features of the old svg unaltered after the addition on new svg was a challenging task. Also capturing the values in variables from first svg and using that in the next svg took a good amount of time in development process..

The development of two svgs took 30 Hours.

**Ahmed Afzal Khan** really likes the part of Data visualization in Data Analysis. As he did good coding in R as well as Python he I choose these two tools to proceed further. Also he wanted to learn something new in R and in depth so I decided to do some visualization using R Shiny package. It took me 2-3 days to refer all the online materials for Python and R Shiny and to understand what are the libraries and packages to be used and what are the types of plots that can be implemented and how using our chosen dataset. The development of the two python visualizations almost took 4-5 hours. And for the two Shiny apps it took 24 hours to implement. For shiny apps he referred the R shiny gallery (<https://shiny.rstudio.com/gallery/>)

**Jayaramakakrishnan Balakrishnan** is a good python programmer. So he planned to use Python. he have used the packages Matplotlib, bubbleplot, plotly, beautifulsoup and seaborn for the vizualisation.The 3rd svg was also done by Jayaram . It took total 30 hours to complete the five visualizations. the dataset that we were using was not having the geographical coordinates. So the coordinates were updated by him to generate the map. Updating this data took a larger amount of time.