A REPORT ON

DATA ANALYTICS OF THE GOFORFIT APP

BY

Name of the Student	ID.No	Discipline
ASHISH GUPTA	2017A7PS0056H	COMPUTER SCIENCE
SIMRAN MALIK	2017A7PS1631H	COMPUTER SCIENCE
ABHISHEK DHANWANI	2017A7PS0161H	COMPUTER SCIENCE
KUNWAR RANANJAY SINGH	2017B4A70504G	COMPUTER SCIENCE

AT

CT Software Solutions Pvt. Ltd. , Gurgaon

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE (July, 2019)

A REPORT ON

DATA ANALYTICS OF THE GOFORFIT APP

BY

Name of the Student	ID.No	Discipline
ASHISH GUPTA	2017A7PS0056H	COMPUTER SCIENCE
SIMRAN MALIK	2017A7PS1631H	COMPUTER SCIENCE
ABHISHEK DHANWANI	2017A7PS0161H	COMPUTER SCIENCE
KUNWAR RANANJAY SINGH	2017B4A70504G	COMPUTER SCIENCE

Prepared in fulfilment of the Practice School-I Course No. BITS F221

At

CT Software Solutions Pvt. Ltd. , Gurgaon

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (July, 2019)

Acknowledgements

We thank the management of BITS-Pilani and the PS division for giving us such a great opportunity to do an internship in a place like **CT Software Solutions Pvt. Ltd. , Gurgaon**

We would like to thank **Rashmi Sharma(C.E.O) and Sujit Panigrahi (C.T.O),** for giving us the invaluable opportunity to undergo our Practice-School-1 program at CT Software Solutions Pvt. Ltd. , Gurgaon

We would also like to thank **Arun Kumar (Mentor in-charge) and Sandeep Singh (Mentor in-charge)** for helping us out at different stages of the project and for providing valuable insight and guiding us throughout the entire project duration**.**

We are grateful to our **PS-Instructor Dr. Arghya Banerjee Sir** for his guidance and assistance in both, project related as well as non-academic matters.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

Practice School Division

Station : CT Software Solutions Pvt. Ltd. **Centre** : Gurugram

Duration:53 days **Date of start** : 21 May, 2019

Date of Submission: 12 July, 2019

Title of the Project :GoForFit Student Analysis

Name	**ID**	**Discipline**
Kunwar Rananjay Singh	2017B4A70504G	Computer Science
Ashish Gupta	2017A7PS0056H	Computer Science
Simran Malik	2017A7PS1631H	Computer Science
Abhishek Dhanwani	2017A7PS0161H	Computer Science

Mentors	**Designation**
Mr.Arun Kumar	Manager
Mr.Sandeep Singh	Software Engineer

PS Faculty : Dr. Arghya Banerjee

Key Words : Data Science and Analytics, Database Management ,SQL Server, SQL Server Management Studio, R Studio

Project Areas : Data Analytics

Abstract:

Using the GoForFit Database,SQL,Rstudio plots were drawn between various fitness indicators like weight,height,test score,percentile against region,age etc.Before plotting the graph we needed to merge the tables after fetching them from database.We needed to filter the data according to benchmark table .Functions from packages like ggplot2 ,dplyr were used for plotting the graphs.Shiny Package was used to make Web App to make program interactive.Main aim of this project was to learn how data analytics can be used in large scale.Today data analytics has become very important and is used for comparison,clustering/grouping/Co-occurrence,Classification,Prediction using the data from various sources.By working on large database (GoForFit) we got experience about data science and work of data scientists.Skill acquired after working on GoForFit database can be used while working on any dataset irrespective of domain of the data.

Table of Contents

Introduction

This report has been compiled to describe the details of the undertaken project. It contains the description of the nature of the problem, the literature surveyed for the project and the methodology adopted. It details the procedures used to obtain the required results and ends with a conclusion.

The Problem Statement:

The GoForFit App was created with the purpose of "Sports Education and Fitness Assessment, Intervention and Monitoring for School Children". As with any app, this app is supported by a database which contains the fitness assessment results of thousands of school students from all over India. However, this data, due to its enormity, virtually serves no purpose unless it is ensured of its consistency and is analysed scrupulously. This project aims at solving this problem of data analysis on the GoForFit App Database by cleaning data and analysing it to reach important conclusions.

History of the problem:

India, with her vast diversity, is yet to showcase extraordinary talent in a large number, when it comes to sports and fitness, save some accomplished sportspersons that she has contributed to the sports industry. A major reason for this is the lack of technical and financial support provided to students at the school level as well as a lesser amount of amenities for

sportspersons. One requirement is for talent from all corners of the country to be recognized and nurtured, as well as training students with sports potential.

Another aspect is the growing concern for the decreased levels of fitness across all ages, owing to the availability of a variety of fast food, adulteration in foods and the increasing prevalence of a sedentary lifestyle. This, in turn, has led to a surge in the number of lifestyle-related conditions like diabetes mellitus. Diabetes currently affects more than 62 million Indians, which is more than 7.1% of the adult population. Nearly 1 million Indians die due to diabetes every year. With startling figures such as these, it is a necessity to focus on increasing the level of awareness about lifestyle choices and fitness.

Hence, the promotion of sports and fitness at the school level is a large step taken towards the growth of the sports culture in the country. The GoForFit App, specifically, helps conduct fitness assessments with greater ease and makes way for their analysis with efficiency. Using the data obtained from a huge number of fitness tests, it is possible to analyse various aspects of one's fitness, to compare students and recognize extraordinary talent.

Reasons for undertaking the project:

As described in the history of the problem, two reasons which govern the need for such an app are promotion of the sports and fitness culture of the country to avail the benefits of an increased number of sportspersons as well as higher levels of overall health and fitness. In addition, the task at hand requires one to handle and analyse data using various technologies and the R language, designed especially for statistical analyses. This would aid one to increase

one's skill set and have hands-on experience in data analytics, given its surmounting significance.

Methodology (Sources and Procedure):

We began with studying the technologies and languages needed by us. We were supplemented with video tutorials for the same. The following were studied:

1. Data Analytics: What is it and what is its significance?
2. R v/s Python: Which is better for statistical analysis?
3. R language: Using R Studio - basic language syntax, packages like ggplot2 and plotly
4. SQL Server and SQL Management Studio
5. Open Database Connectivity: An alternative to the SSMS approach

These topics are discussed in the next section.

Next, we implemented various analyses on the database and displayed our results graphically.

Theoretical requirements of the project:

1. **Data Analytics: What is it and what is its significance?**

Data analytics, as the name suggests the process of cleaning, filtering and examining data to reach important conclusions regarding the information it contains. Raw data, by itself, has little value; this helps use data effectively, helping us derive relationships and conclusions which are not obvious. These conclusions further help take informed decisions.

Some applications data analytics in a variety of fields are briefed below.

i. *Psychology*

The entire branch of psychology and its theories are based on data and its analysis. Experiments are conducted in which data is obtained from subject. That data is then analysed to verify or falsify the hypothesis.

ii. *Advertisements*

Advertisements that appear on the websites we surf are tuned to our preferences. Websites record our search history and analyse it to determine our likes so that they can accordingly display advertisements on our pages. This helps users because it prevents irrelevant ads from showing up. The advertising company is benefited as it can now target potential customers. The ad hosting website's company generates higher revenue as the more user-preferred ads are shown to a user, higher is his/her tendency to view the advertisement.

iii. *Financial Analysis*

The complete stock market is based on predictions. These predictions are made by analysing financial data of various financial entities in the market. With the help of these, more accurate decisions can be taken.

iv. *Feedback Analysis*

In any service providing sector, service providers take customer feedback. That feedback is analysed and interpreted. Changes are made, if required. This helps customers to have a higher level of satisfaction and the service providers to attract a larger number of customers.

v. *Government initiatives*

Various kinds of data is collected pertaining to the country. From it, various conclusions can be drawn, for instance, finding the poorest region in the country. Data pertaining to the daily income can be analysed and averaged to find out the poorest region. The govt can then allot

resources to resolve the issue.

2. **R v/s Python: Which is better for statistical analysis?**

For a growing number of people, data science is a central part of their job. Increased data availability, more powerful computing, and an emphasis on analytics-driven decision in business has made it a heyday for data science. According to a report from IBM, in 2015 there were **2.35 million openings** for data analytics jobs in the US. It estimates that number will rise to 2.72 million by 2020. The two **most popular programming tools for data science work are Python and R at the moment**.

R was developed in 1992 and was the preferred programming language of most data scientists for years. It is a **procedural language** which works by breaking down a programming task into a series of steps, procedures, and subroutines. This is a plus when it comes to building data models because it makes it relatively easy to understand how complex operations are carried out; however, it is often at the expense of performance and code readability.

R's analysis-oriented community has developed **open-source packages for specific complex models** that a data scientist would otherwise have to build from scratch. R also emphasizes quality reporting with support for **clean visualizations** and frameworks for creating interactive web applications. On the other hand, slower performance and a lack of key features like unit testing and web frameworks are common reasons that some data scientists prefer to look elsewhere.

Python was released in 1989 with a philosophy that emphasizes **code readability** and efficiency. It is an **object-oriented programming language**, which means it groups data and code into objects that can interact with and modify one another. Java, C++, and Scala are other examples. This sophisticated approach allows data scientists to execute tasks with better stability, modularity, and code readability.

Data science is only a small portion within the **diverse Python ecosystem**. Python's suite of specialized deep learning and other **machine learning** libraries includes popular tools like Panda, TensorFlow etc which enable data scientists to develop sophisticated data models that plug directly into a production system.

It is hard to pick one out of those two amazingly **flexible** data analytics languages. Both are **free and open source**, and were developed in the early 1990s — **R for statistical analysis and Python as a general-purpose** programming language. For anyone interested in machine learning, working with large datasets, or creating complex data visualizations, they are absolutely essential.

3. **R language: Using R Studio - basic language syntax, ggplot2 and plotly**

RStudio and R: RStudio provides popular open source and enterprise-ready professional software for the R statistical computing environment.RStudio makes R easier to use. It includes a code editor, debugging & visualization tools. As R is a programming language and free

software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing let's discuss a few of the important terms used in R

vector:Vectors are the most basic R data objects.Vectors are the most basic R data objects

Data Frame:A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column

package:Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library

Why need packages?:R packages provide a simple way to distribute R code and documentation.

The packages that were used in our projects were ggplot2 and plotly.

ggplot2:ggplot2 is a data visualization package for the statistical programming language R. It is a very powerful tool to plot the graph

plotly:Plotly is an R package for creating interactive web-based graphs via the open source JavaScript graphing library plotly.

4. **SQL Server and SQL Management Studio(SSMS):**

SQL Server:Microsoft SQL Server is a relational database management system developed by Microsoft .It is used for storing and retrieving the data by other software applications—which may run either on the same computer or on another computer across a network .This server is used for connecting our Rstudio to the live database of GoForFit.

SSMS:GUI Administration tool by microsoft for working against the SQL server engine which makes life easier to use sql server

5. **Open Database Connectivity: An alternative to the SSMS approach**

We can export data from SQL server or Microsoft SQL Server Management Studio in 2 ways:

1) Exporting the table that is obtained from the query executed in the Microsoft SQL server management studio to the excel file in the form of .csv format

2)By directly connecting SQL server with RStudio by using Packages like RODBC,ODBC e.t.c i.e.Open Database Connectivity

The second method is more effective than the second method because in the first method we have to export the data into the .csv format which results in the data redundancy and wastage of memory and there is wastage of time in joining tables in SSMS and exporting the data to an excel file and then reading them in the RStudio but the above process is not required in Open Database Connectivity i.e we can directly connect our RStudio to the SQL server and extract data into a dataframe by writing query in RStudi0 itself which reduces the wastage of memory and data redundancy.

System Requirements to setup the Shiny App

1. Windows platform
2. R Studio with the following libraries installed:
    a. RODBC
    b. RODBCext
    c. ggplot2
    d. plotly
    e. shiny
    f. shinyWidgets

3. Connection to CT Software Solutions Pvt Ltd Local WiFi or LAN

R Packages and their methods used to design our application

a. RODBC
    i. odbcDriverConnect() : To connect to the database
    ii. sqlGetResults() : Returns the results of a query (in the form of a data frame) associated with the passed database connection handle
    iii. sqlDrop() : Removes the table SQL table (if permitted)
    iv. sqlSave() : Write or update a table in the database
b. RODBCext
    i. sqlPrepare() : To pass a query (can be parameterised) via the database connection handle
    ii. sqlExecute() : To pass parameters to a query prepared by sqlPrepare()
c. ggplot2
    i. ggplot() : To initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.
    ii. geom_bar() : To create a bar chart
    iii. position_dodge() : Dodging preserves the vertical position of an geom while adjusting the horizontal position.
    iv. scale_fill_manual() : To specify our own set of mappings from levels in the data to aesthetic values
    v. labs() : To label the graph

vi. theme() : To format visual parameters of the graph such as background colour of the graph, text size and so on.

d. plotly

i. ggplotly() : To convert a ggplot2::ggplot() object to a plotly object.

ii. plotlyOutput() : To output and render functions for using plotly within Shiny applications in ui.R

iii. renderPlotly(): Output and render functions for using plotly within Shiny applications in server.R

e. shiny

i. shinyUI() : to register a user interface with Shiny

ii. fluidPage() : To scale components of a Shiny app in realtime to fill all available browser width.

iii. titlePanel() : Create a panel containing an application title.

iv. sidebarLayout() : Create a layout with a sidebar and main area

v. sidebarPanel() : Create a sidebar panel containing input controls that can in turn be passed to sidebarLayout.

vi. selectInput() : Create a select list that can be used to choose a single or multiple items from a list of values.

vii. uiOutput() : Render a reactive output variable within an application page.

viii. actionButton(): Creates an action button or link whose value is initially zero, and increments by one each time it is pressed.

ix. textOutput() : Render a reactive output variable as text within an application page.

x. reactive() : Wraps a normal expression to create a reactive expression.

xi. renderUI(): Renders reactive HTML using the Shiny UI library

xii. observeEvent(): Respond to "event-like" reactive inputs, values, and expressions.

xiii. showNotification(): To display a notification in a Shiny app

xiv. renderText(): Makes a reactive version of the given function to turn its result into a single-element character vector.

xv. isolate(): Executes the given expression in a scope where reactive values or expression can be read, but they cannot cause the reactive scope of the caller to be re-evaluated when they change.

f. shinyWidgets

i. setBackgroundColor() : To change the background colour of our shiny app

g. ODBC

i. odbc(): To connect to the database.

ii. dbGetQuery(): To send query, retrieve results and then clear result set

h. DPLYR

i. mutate(): To add new variables and preserves existing; transmute drops existing variables.

Task 1 : Percentile ranges and Percentage of students

We plotted bar graphs representing percentile ranges(0-20,20-40,40-60,60-80,80-100) on x-axis and percentage of students on y-axis .

We did this in following steps -

1.first we created excel sheet containing senior test result table obtained from GoForFit database in Microsoft SQL

2.Then CSV file was obtained from excel file(because R can read CSV file).

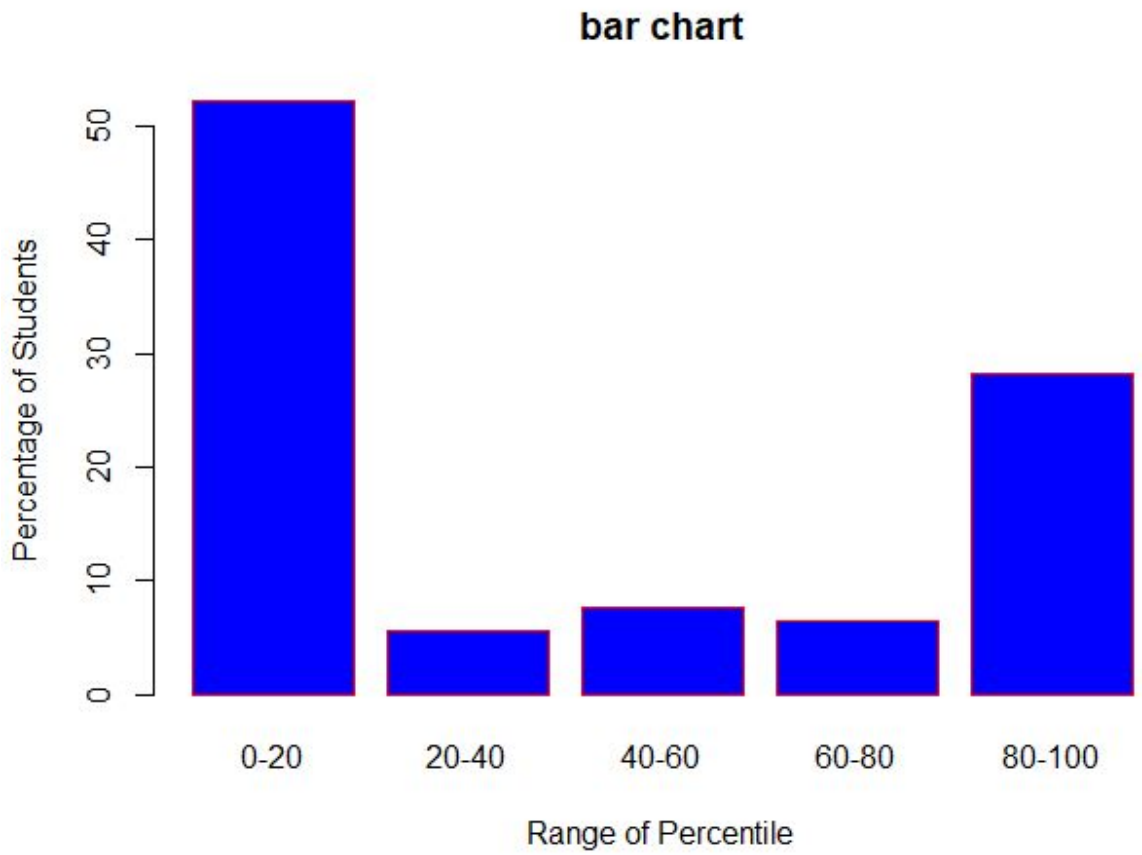
3.Column containing the Percentile of Students is fetched from the dataframe.

4.Then sum function was used to find the total number of students in each of the ranges (0-20,20-40,40-60,60-80,80-100).

5.Percentage was calculated using data obtained in previous step.

6.Barplot function was used to obtain the plot .

Following is the barplot we obtained -


bar chart
Percentage of Students
0
10
20
30
40
50
0-20
20-40
40-60
60-80
80-100
Range of Percentile

Task 2 : Individual Age and Average Percentile :

Purpose of this graph

To analyse the relationship between the age of a student and the average performance as demonstrated by students that age. This will help us compare the performance of students of different ages and display which set of students lack in performance and hence, need extra training.

Requirements

We want to demonstrate the relationship between a particular age and the percentile obtained by students that age on an average, by plotting a barplot for the same.

Data Extraction and Cleaning

The GoForFit database is a large database containing many relations. We want to obtain the age of a student and his/her average percentile. This information is present in a single table. After obtaining this data, we needed to eliminate inconsistent data that is, filter it. This was done by removing data in which age of students is wrongly entered that is, outside 5 and 20.

What was used

SQL Server : was used to make the connection to the database provided to us

SSMS : was used as an interface to the SQL Server and was used to obtain the required data in the form of a relation

MS Excel : was used to convert the data to a Comma Separated Value file

R Studio : was used to clean the data by removing ages beyond 5 and 20 years of age. It was used to calculate and store the sum and average percentile of students of a particular age. And lastly, it was used to generate a bar plot between age and average percentile of that age.

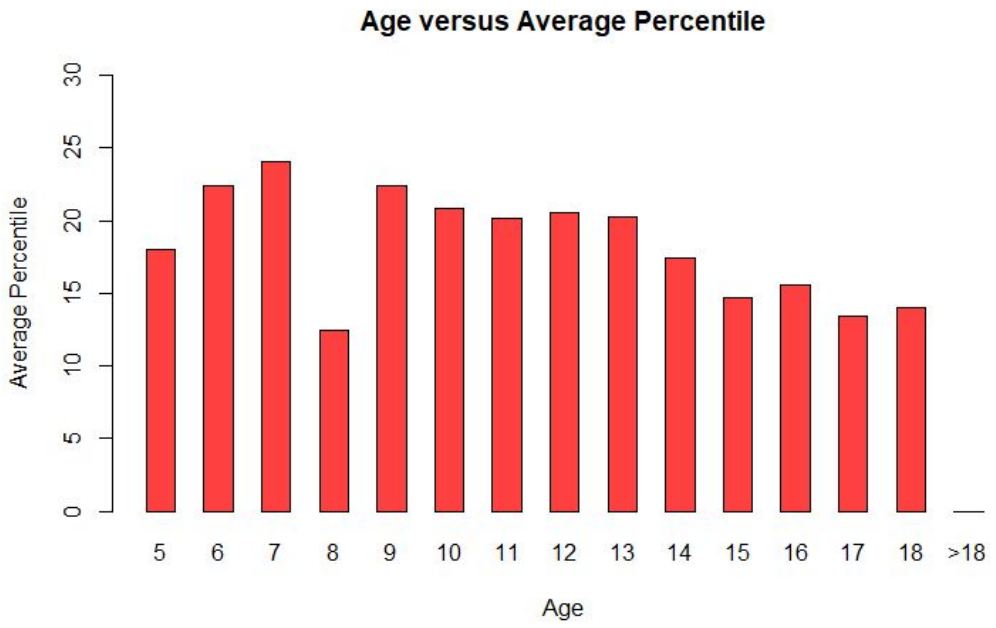
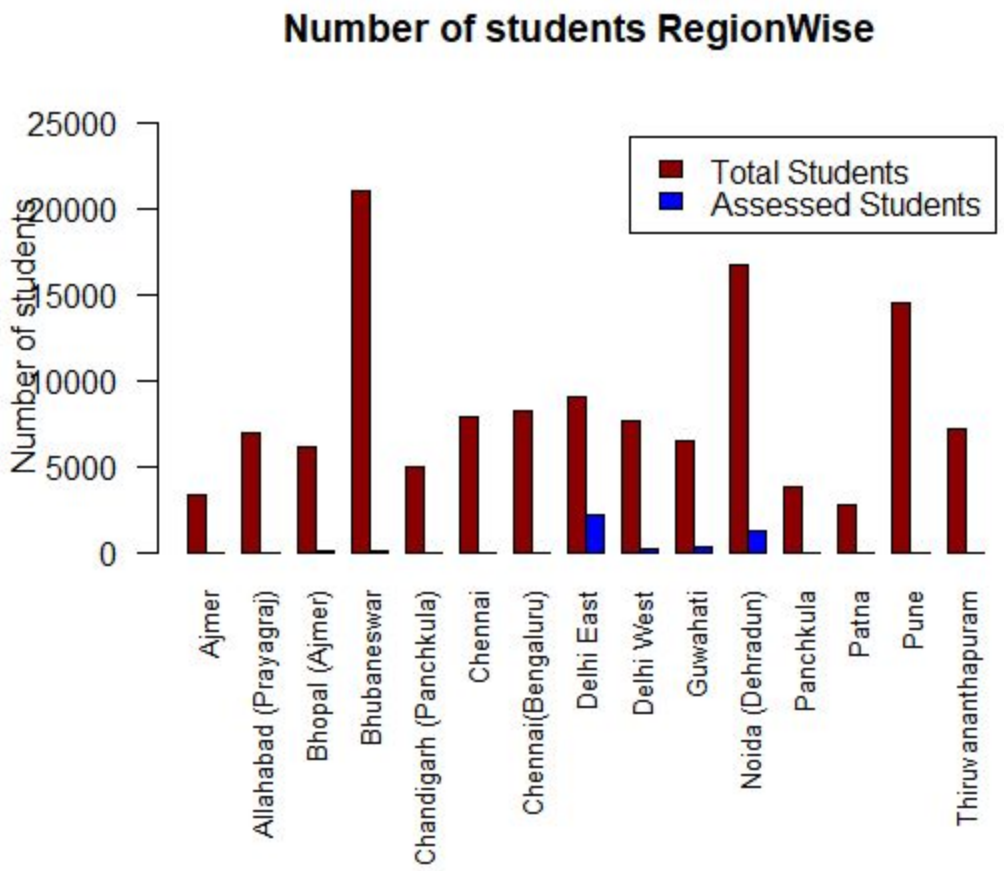


Fig: Age versus average percentile

Task 3: Assessed and Total students Region-Wise.

Grouped Bar PLot representing Assessed and Total students Region-Wise.



Flow Line

Task was to write a modularised code to depict the above graph

1. First from the proper tables of the GoForFit Database the appropriate and required attributes were extracted and merged in a dataframe.

2. Data was cleaned and filtered to meet the requirements.

3. Total number of students and number of assessed students was calculated and stored in different vectors.

4. With the use of barplot the above graph was generated.

What was used

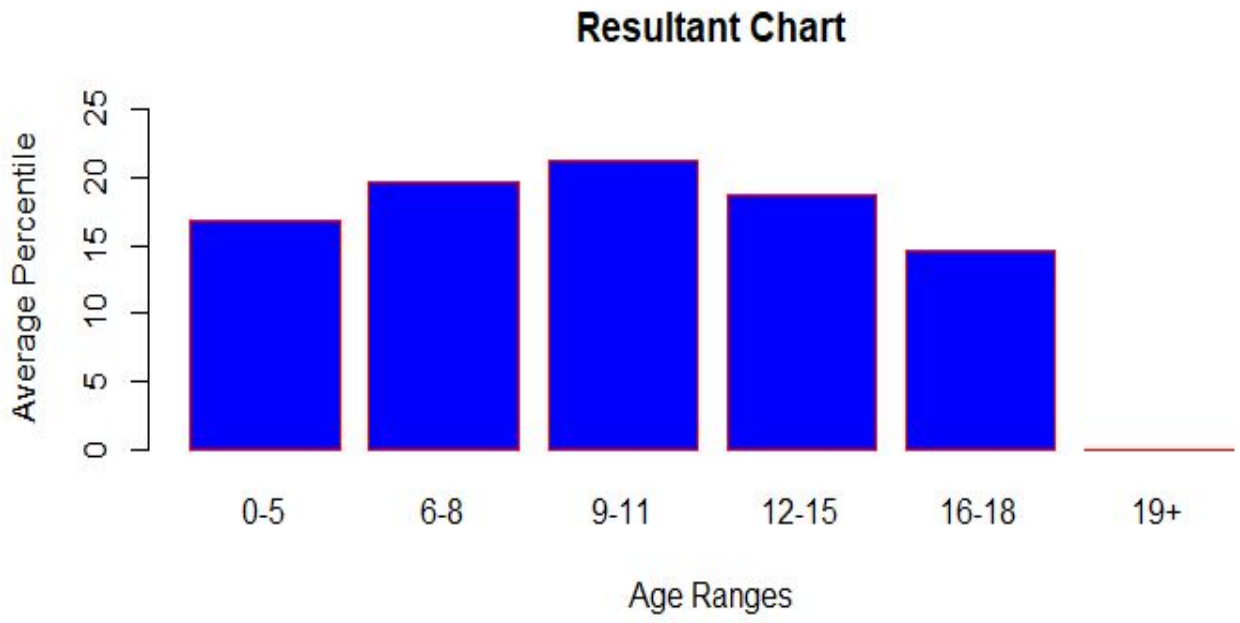
SQL Server : was used to make the connection to the database provided to us

SSMS : was used as an interface to the SQL Server and was used to obtain the required data in the form of a relation

MS Excel : was used to convert the data to a Comma Separated Value file

R Studio : was used to clean the data by removing ages beyond 5 and 20 years of age. It was used to calculate and store the sum and average percentile of students of a particular age. And lastly, it was used to generate a bar plot between age and average percentile of that age.

Task 4 : Bar Plot representing the average percentile of the students of different age groups



The flow line to plot the above graph was similar to the flow line of the task 3.

Purpose:

1)Knowing how students of different age groups are performing

2)Comparing the performance of the students of different age groups

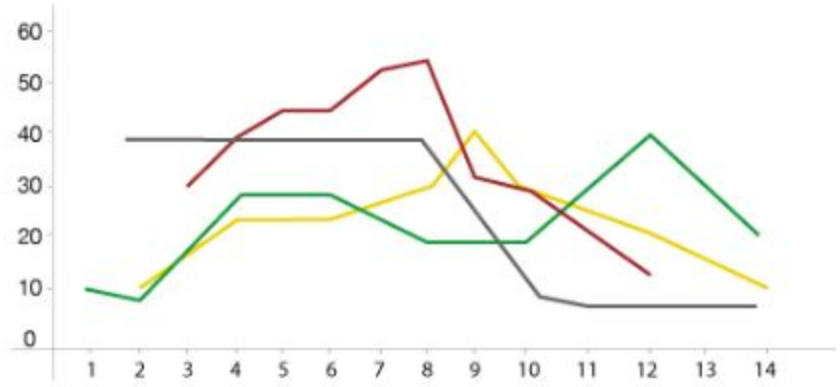
3)A student can compare his performance with his age group and also with performance other age groups and can improve on his performance.

Health Indicator Graphs

Using GoForFit database Health Indicator Graphs were made region wise and age wise.For plotting these graphs we need to calculate BMI of every student using the information in the database.For this we need to merge various tables like Student Master , City Master , Senior Test Result , Region Master ,Board Chain Mapping,Board Master,Zone Master.

Before starting this task knowledge of various packages like ggplot,dplyr,reshape was acquired.

The graphs required was a multiple line plot.-



(example of multiple line plot)

Where on the y-axis we need to have percentage and on the x-axis we need to have region or age and lines in line plot should correspond to the percentage of students underweight,normal,overweight or obese in a given region (or a given age).

Health Indicator Graph (Region-Wise)

To get region wise health indicator multiple line plot a program was made which takes following as input -

1.Board or Chain .

2.Board Id or Chain Id.

3.Zone Id.

If use chooses Board in first option he/she will have to provide Board Id in second option.Else the user have to provide Chain Id in second option.Zone Id is optional part.If user provides Zone Id then we need to consider only those schools which are present in that zone else if Zone Id is not provided program will consider all Zones in consideration.

Following procedure was taken to plot the graph -

1.Installed packages RODBC , ggplot2 , reshape2 ,dplyr so that we can use some of the functions which are present in those packages.

RODBC package was installed so that our program can connect to the server directly and fetch a table or can run a query.(it saves a lot of time from making excel sheet and then making CSV file).

ggplot is used to plot several graphs and has more options in terms of visualisation compared to qplot or mat plot.

Reshape is used to melt the data before we are going to pass it to ggplot function for plotting the

graph.

Dplyr package is used in Selecting, filtering, and aggregating the data.

2.Tables like Student Master , City Master , Senior Test Result , Region Master ,Board Chain Mapping,Board Master,Zone Master were fetched from database and they were joined with each other using merge function.

Because duplicate rows are created in this process and we need to remove those rows according to student ID. Also we can remove those columns which we don't require so that less memory space is taken.

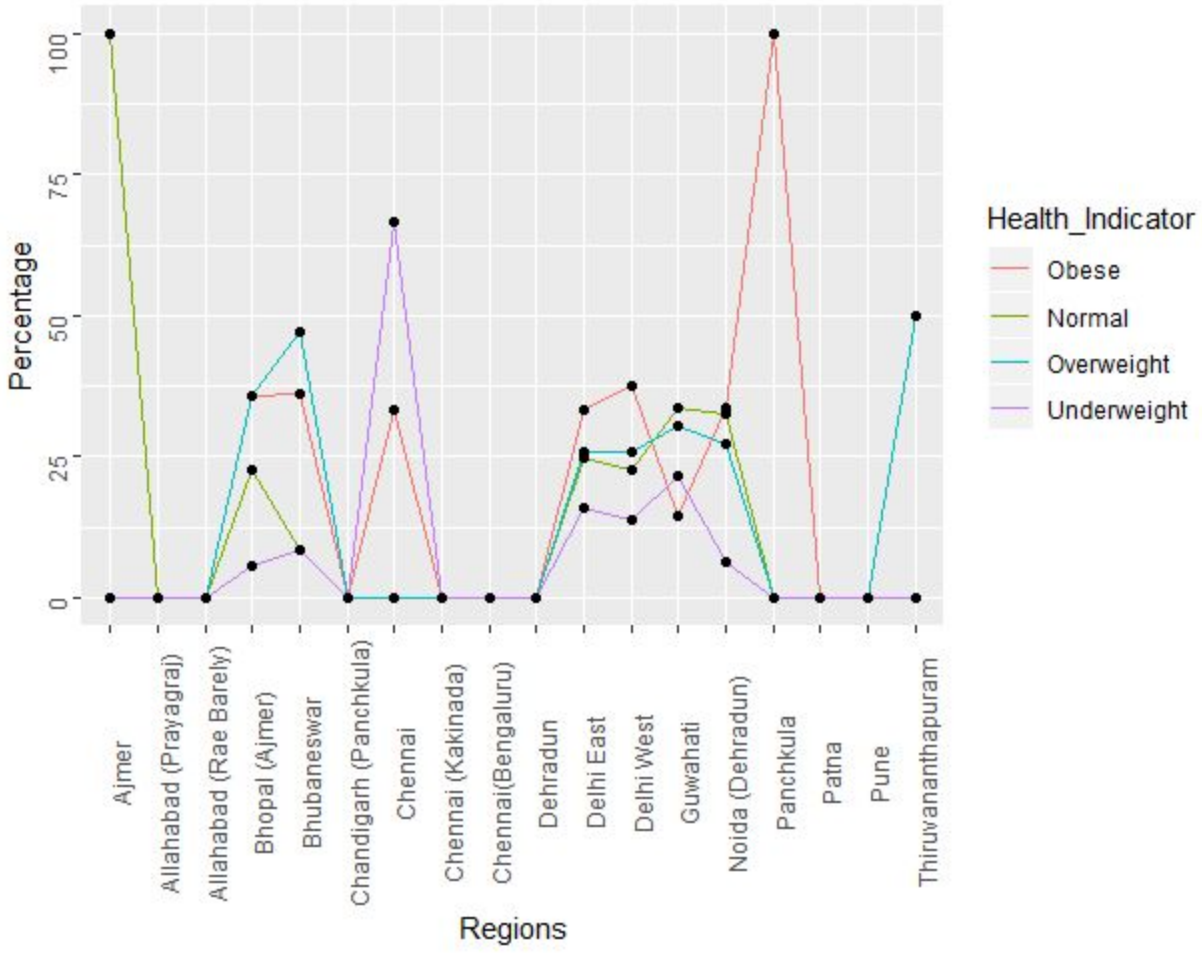
3.The column containing score of height and weight is fetched from the dataframe formed in step 2.BMI is calculated from both the columns.This column (containing the BMI) is finally attached to the dataframe formed in step 2.

4.Using Age ,Gender ,BMI columns and benchmark table a column is made which tells if a student is underweight,normal,overweight or obese.Percentage of students who are underweight ,normal,overweight or obese is calculated .While calculating percentage sometimes we get nan value and were removed.

5.Finally the dataframe is further filtered by the Board(or Chain) , Board Name (or Chain Name) and Zone Name given by the user.

6.Column made in 4 and region column are separated from the dataframe and are plotted using ggplot function.

Following is one of the outputs obtained for CBSE board students (for all zones) -



Web App for Region-Wise Health Indicator Program

We used shiny package for making web pages.Before Shiny, data analysts used to write algorithms in R and then work with web developers to make data visualizations for websites.Shiny let's R developers to make interactive data visualizations for the web pages .

Following procedure was followed for developing web app for Region-Wise health indicator program -

1.Two page ui.R (for developing user interface),server.R(for doing work server side work) are made.

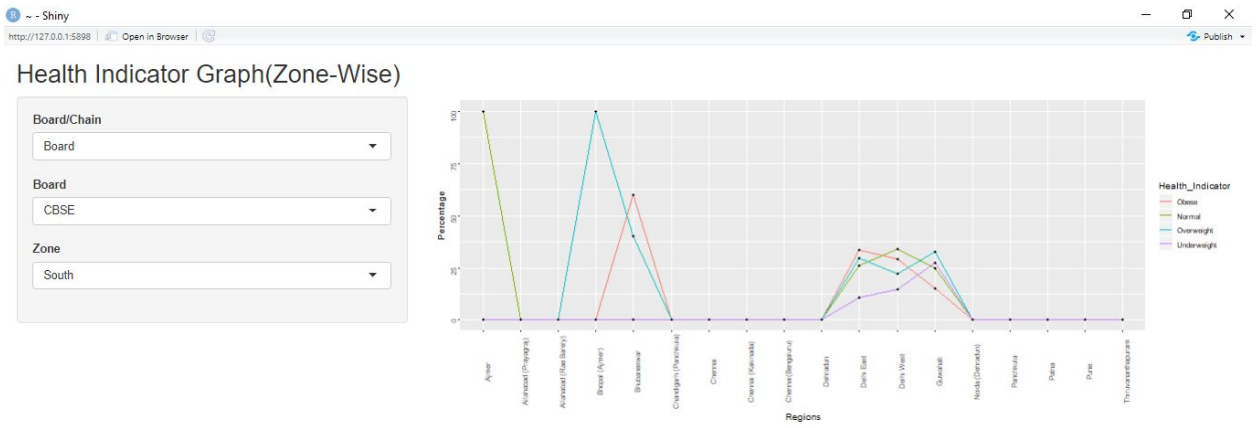
2.In ui.R we need to write code for making Title Panel .After that we need to write code for sidebar panel (in which we need to pass function for making drop down menu for Board and Zone).

3.In ui.R , the code for drop down menu for Board and Chain is written in the form of conditional panel .Conditional Panel is used when we want to display drop down menu according to choice selected in the drop down menu above it.In this case first drop down menu has two choices - Board,Chain .Now if Board is selected , due to conditional panel second drop down menu will appear which will have name of different boards.If Chain is selected then second drop down menu will appear which will have name of different Zones.

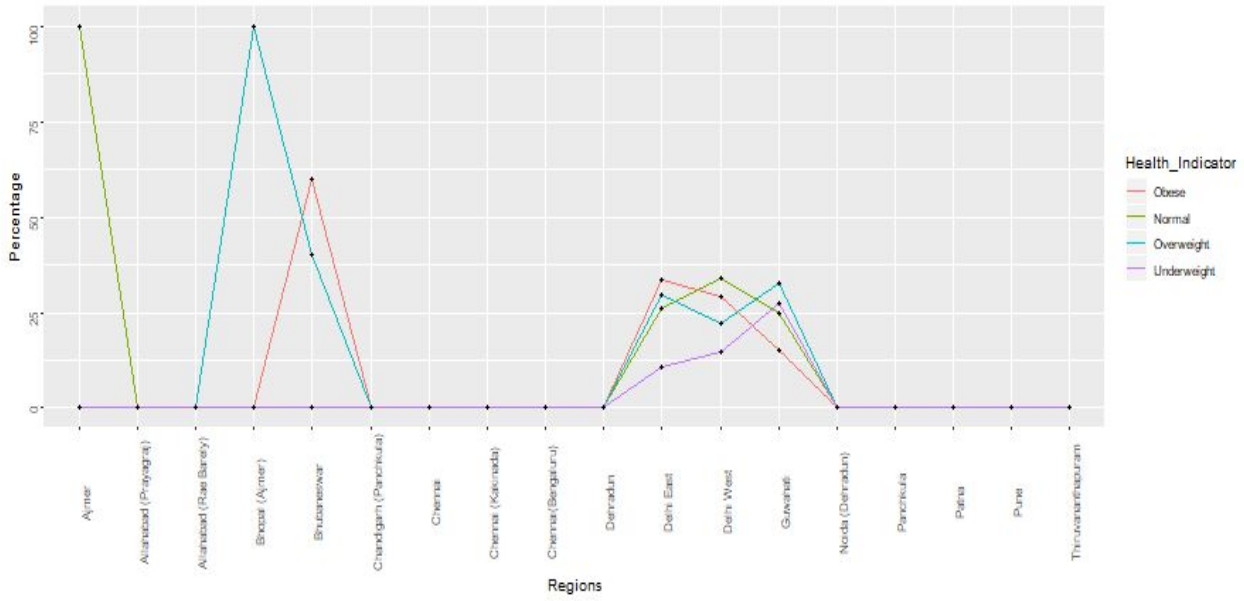
4.In ui.R code for mainPanel is written which displays visualisation on main panel of web page.

5.In server.R all the data selected by the user is fetched and passed to a function which makes the visuation and stores it into a variable .That variable is used by ui.R to display the visualisation.

Following is one of the outputs obtained -



Following is the graph in above output -



Web App for Age-Wise Health Indicator Program

Following is the procedure for making age-wise health indicator graph (similar to region-wise health indicator graph)-

1.Two page ui.R (for developing user interface),server.R(for doing work server side work) are made.

2.In ui.R we need to write code for making Title Panel .After that we need to write code for sidebar panel (in which we need to pass function for making drop down menu for Board, Zone,Region,State,City,School Name,Type(i.e Line Plot,Bar Plot,Pie Chart)).

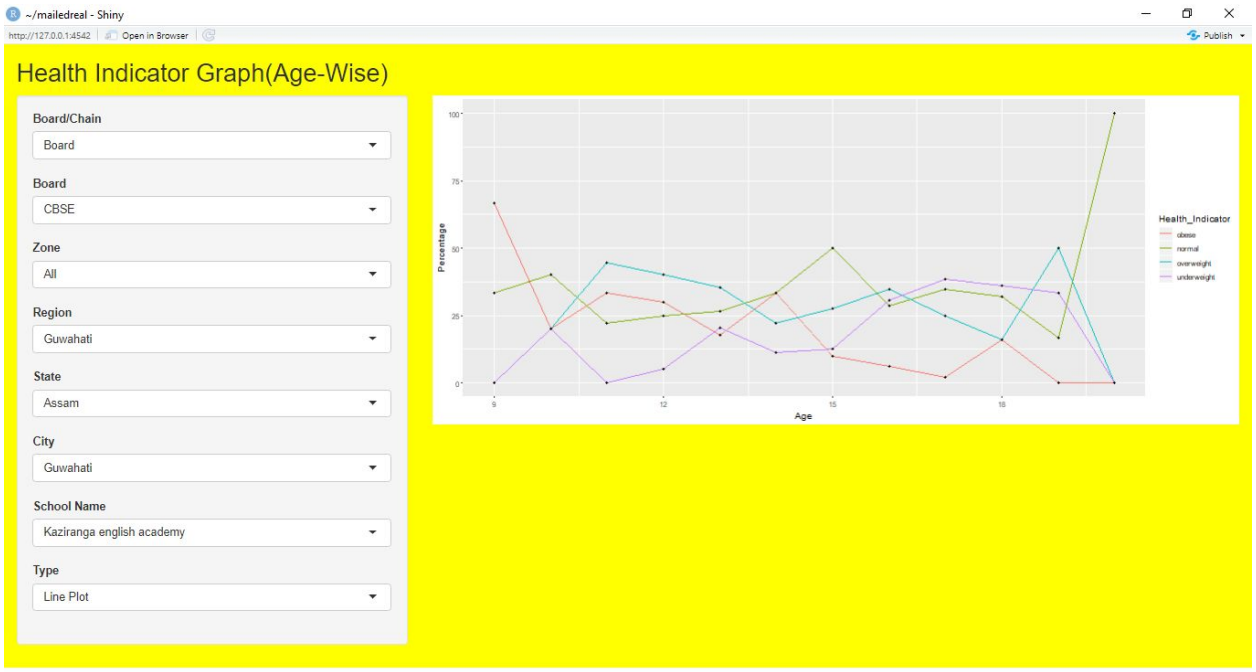
3.In ui.R , the code for drop down menu for Board and Chain is written in the form of conditional panel .Conditional Panel is used when we want to display drop down menu according to choice selected in the drop down menu above it.In this case first drop down menu has two choices - Board,Chain .Now if Board is selected , due to conditional panel second drop down menu will appear which will have name of different boards.If Chain is selected then second drop down menu will appear which will have name of different Zones.

4.In ui.R code for mainPanel is written which displays visualisation on main panel of web page.

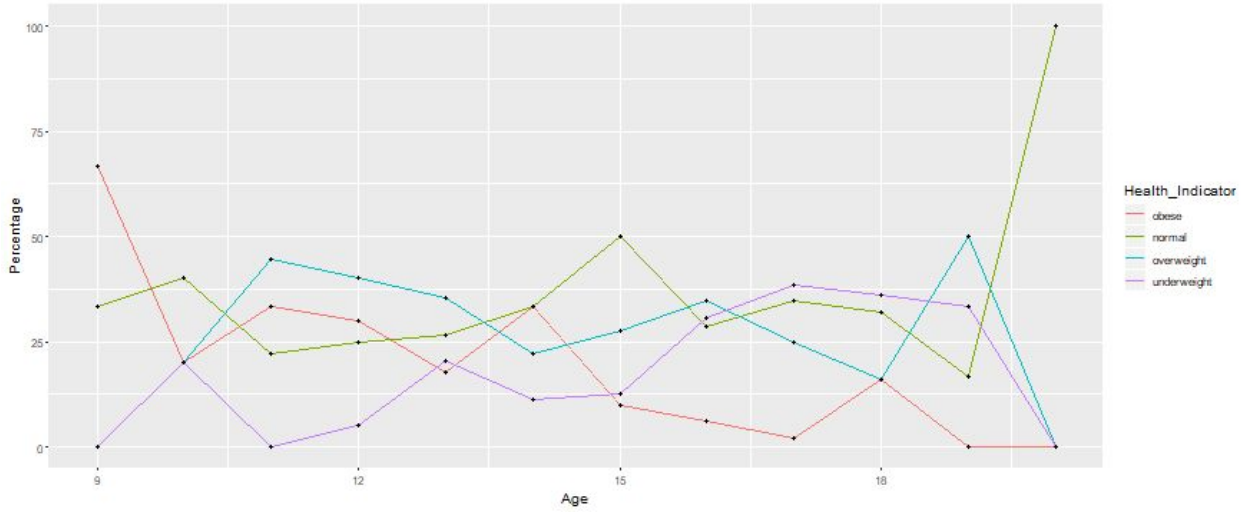
5.Since we want drop down menus to follow a hierarchy i.e Options available in a drop down menu will depend on option chosen in previous drop down menu we need to use reactive and

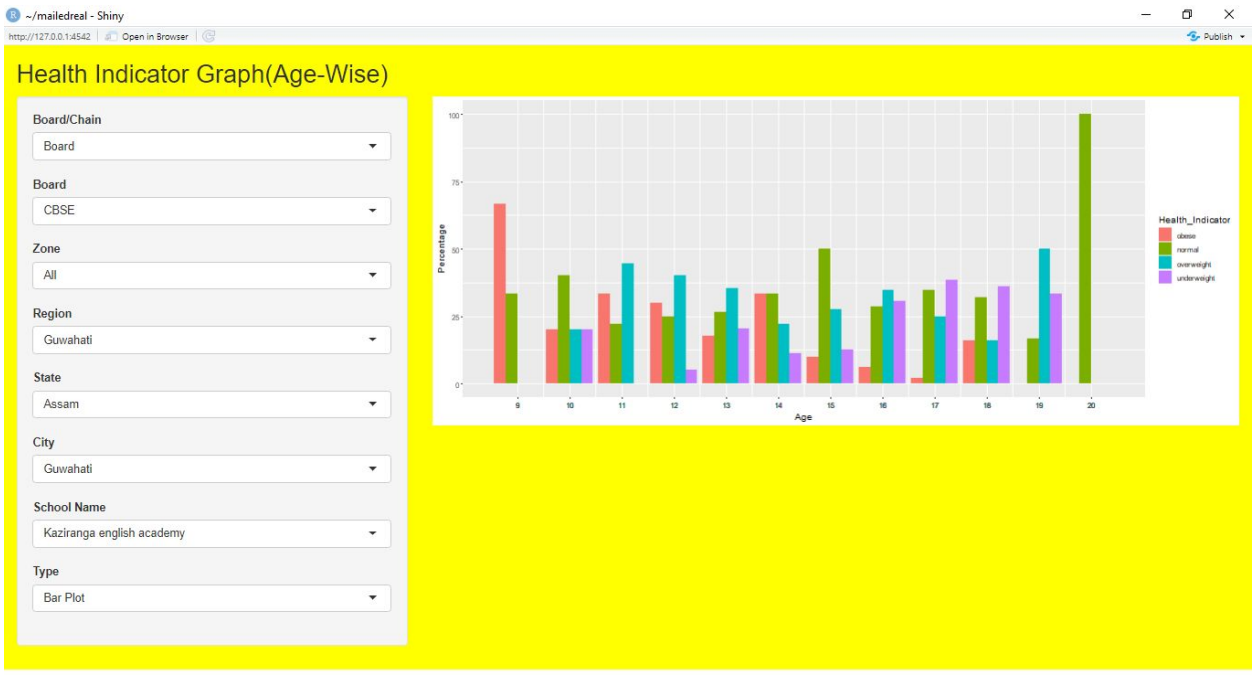
renderUI function.For example - Suppose we choose North in Zone option then only those regions will appear in Region option which are present in north zone.

6.In server.R all the data selected by the user is fetched and passed to a function which makes the visuation and stores it into a variable .That variable is used by ui.R to display the visualisation.

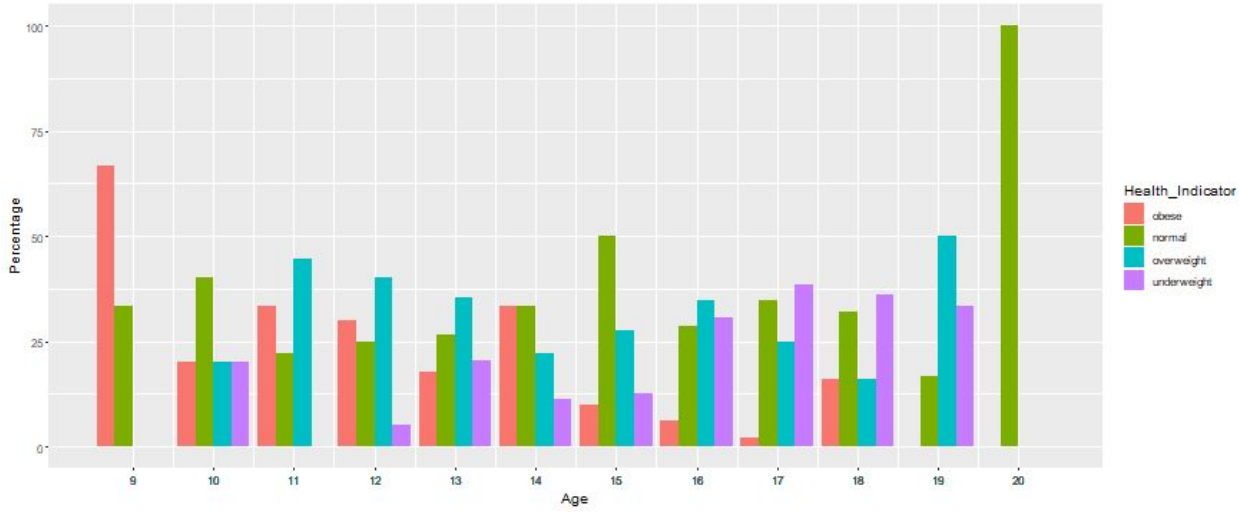


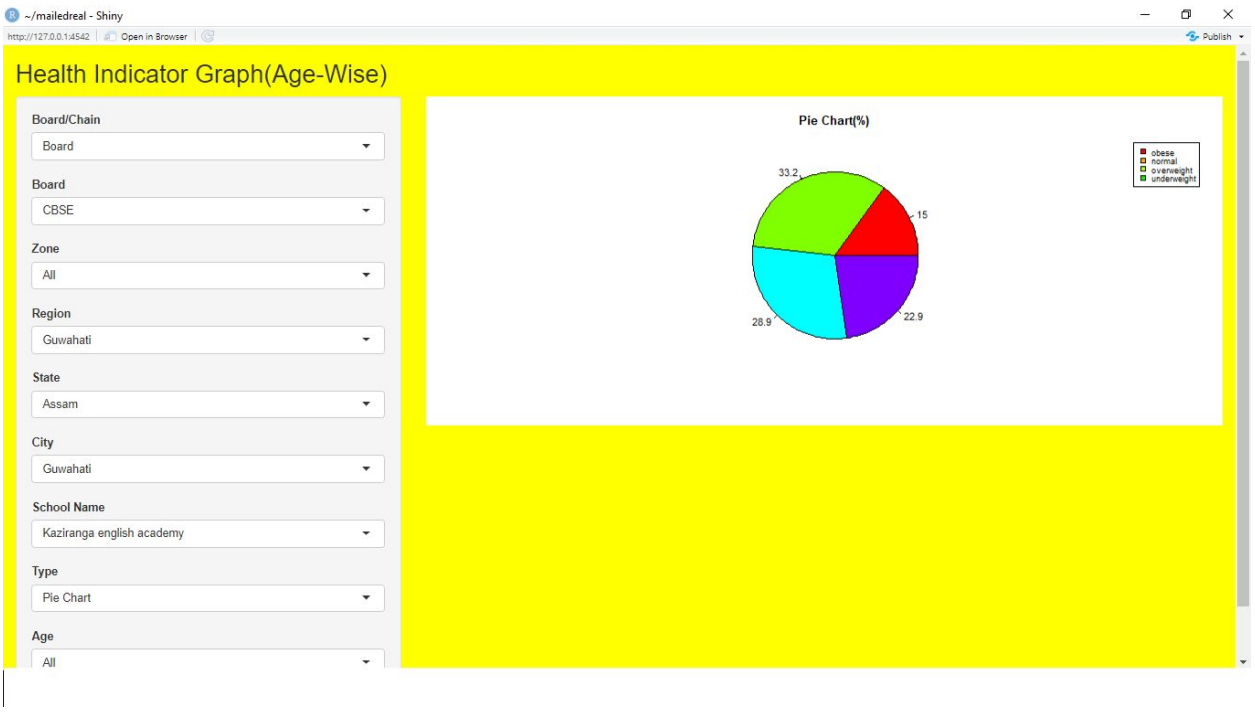
Line plot in above output -



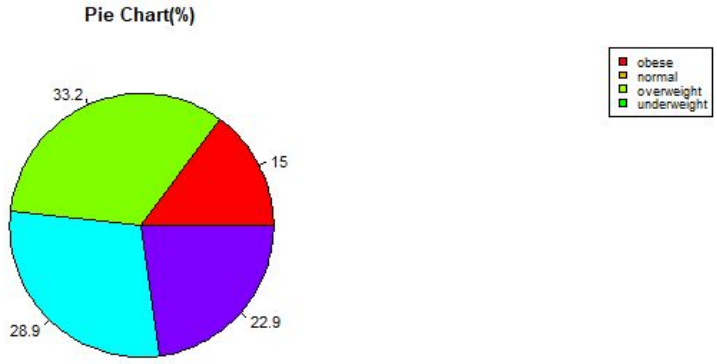


Bar plot in above output -





Pie Chart in above output -


Pie Chart(%)
obese
normal
overweight
underweight
33.2
15
28.9
22.9

Health Indicator Graph(Age-Wise)

Following procedure was taken to plot the health indicator graph (Age-Wise) -

1.For plotting age-wise health indicator graph we need to take the following as input from user -

a.Region Id,

b.State Id

c.City Id

d.School Id

The packages (RODBC , ggplot2 , reshape2 ,dplyr) required for plotting region-wise health indicator graph are also used for plotting age-wise health indicator graph.

2.Tables like Student Master , City Master , Senior Test Result , Region Master ,Board Chain Mapping,Board Master,Zone Master were fetched from database and they were joined with each other using merge function.
Because duplicate rows are created in this process and we need to remove those rows according to student ID.

Also we can remove those columns which we don't require so that less memory space is taken.

3.The column containing score of height and weight is fetched from the dataframe formed in

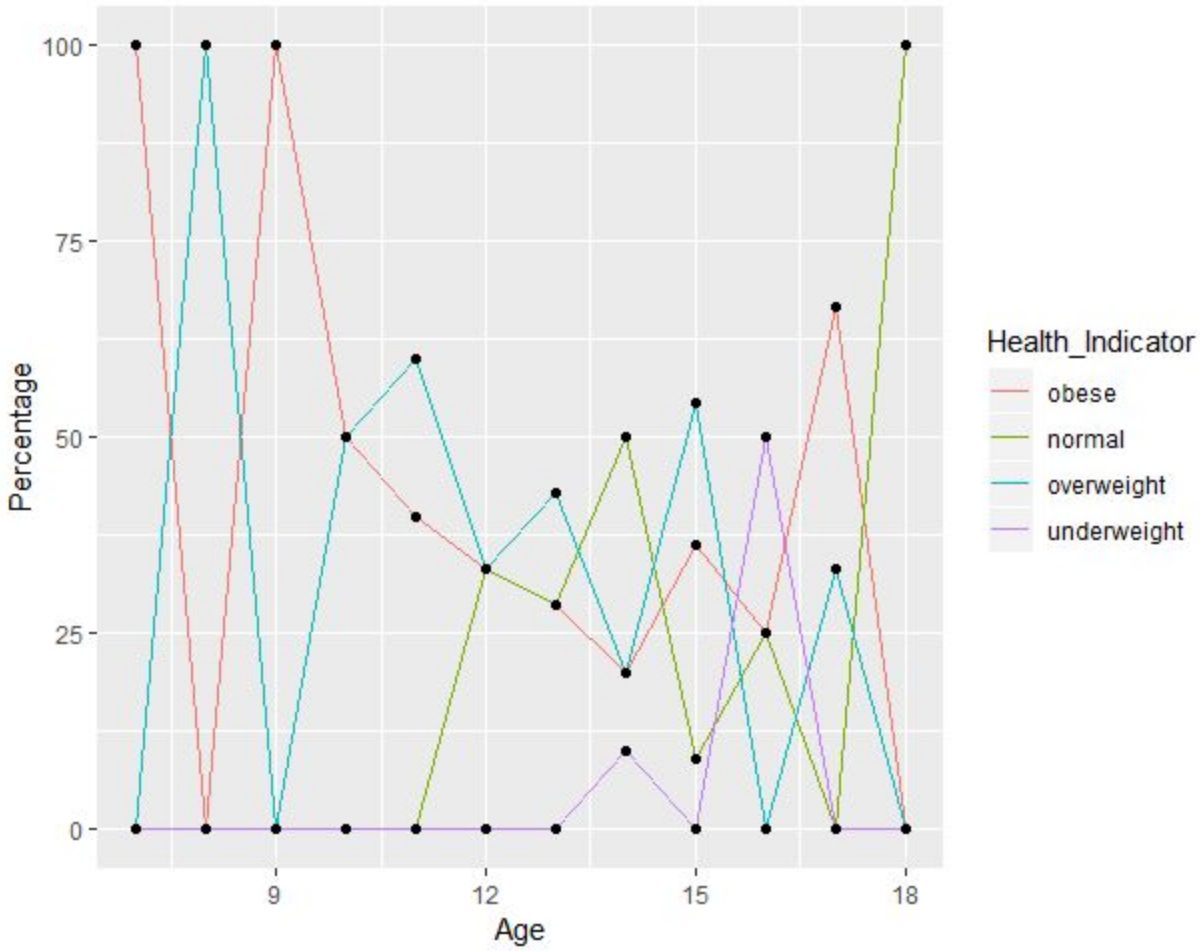
step 2.BMI is calculated from both the columns.This column (containing the BMI) is finally attached to the dataframe formed in step 2.

4.Using Age ,Gender ,BMI columns and benchmark table a column is made which tells if a student is underweight,normal,overweight or obese.Percentage of students who are underweight ,normal,overweight or obese is calculated .While calculating percentage sometimes we get nan value and were removed.

5.Finally the dataframe is further filtered according to Region Id,State Id,City Id,School Id given by the user.

6.Column made in 4 and Age column are separated from the dataframe and are plotted using ggplot function.

Following is one of the outputs obtained for region Id = 11,state Id = 21,city Id = 361,school Id = 62


100
75
50
25
0
Percentage
9
12
15
18
Age
Health_Indicator
obese
normal
overweight
underweight

The Performance Indicator (based on Average Score)

Purpose of this graph

To analyse the relationship between the age of a student and the average performance as demonstrated by students that age. This will help us compare the performance of students of different ages and display which set of students lack in performance and hence, need extra training.

Requirements

We want to demonstrate the relationship between age and the average score of students by plotting the average score of students which meet the required specifications as mentioned by the user.

The user can filter students on the basis of the following fields:

Ii. Joined Through(Board or Chain)

ii.Board Name or Chain Name

iii.Zone Name

iv.Region Name

v.State Name

vi.City Name

vii.School Name

viii. Term

ix. Age Group

Students belonging to the particular categories as mentioned by the user should be analysed for performance. If a given field is unspecified, it is presumed that the user wants to analyse all students regardless of that field value.

A brief on the followed procedure

1. Install odbc and DBI packages
2. Connect to the database
3. Access the appropriate tables in the database to retrieve the required data. If input is wrongly entered by the user, display an error message
4. Clean data to remove any inconsistency
5. Manipulate data to find the average score of each student and hence, for each age
6. Install ggplot2 and plotly packages
7. Plot the bar graph with age on the x-axis and average score on the y-axis

A Detailed Procedure

Connecting to the database:

Connect to the database using the method: dbConnect() by filling in the required fields: Driver, Server, Database, Username, Password and Port

Relations accessed:

SeniorTestResults: for StudentID, Age, TestTypeID Score

StudentMaster: for StudentID, StudentName, Gender, SchoolID

SchoolMaster: for SchoolID, SchoolName, BoardID/SchoolChainID, ZoneID, RegionID, StateID, CityID

BoardMaster: for BoardID, BoardName

ZoneMaster: for ZoneID, ZoneName

RegionMaster: for RegionID, RegionName

StateMaster: for StateID, StateName

CityMaster: for CityID, CityName

Error Conditions:

1. JoinedThrough field is empty
2. Given the order:

    i.Board Name or Chain Name

    ii.Zone Name

    iii.Region Name

    iv.State Name

    v.City Name

    vi.School Name

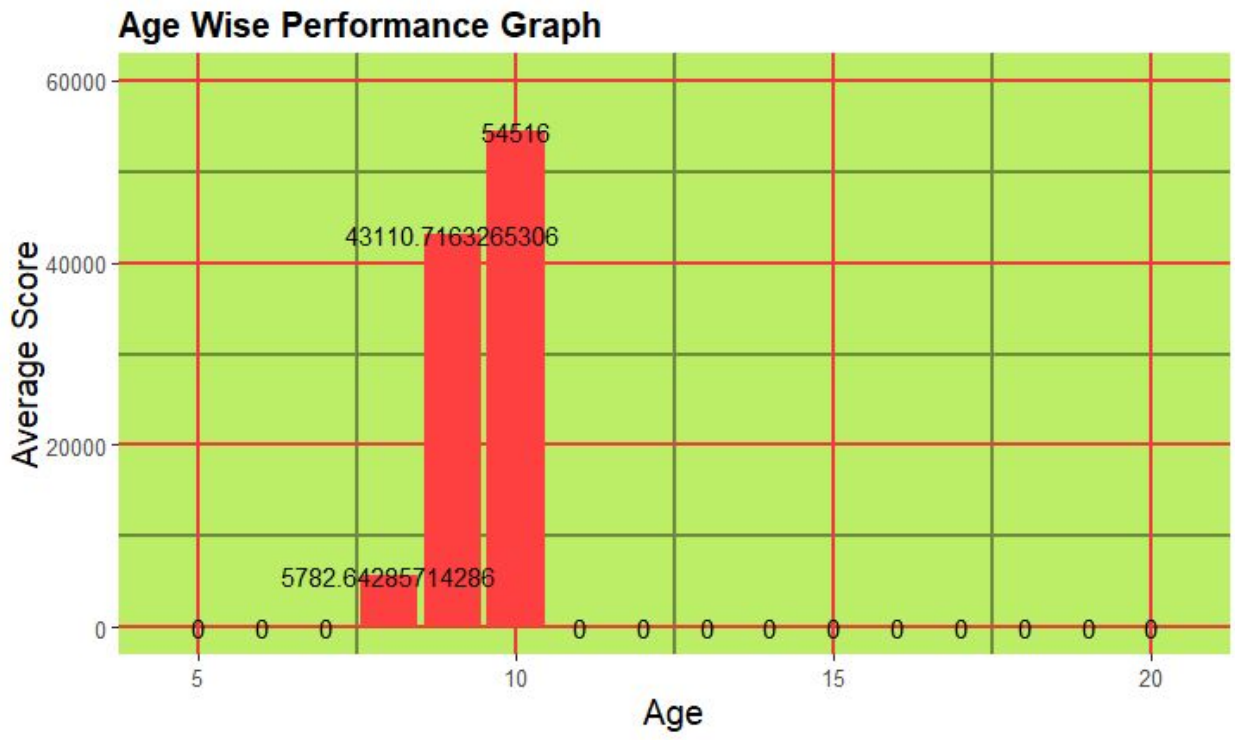
If any of the fields is empty and one or more fields below it, are non-empty.

Cleaning data:

1. Removing all entries with age of students outside [5,20]

2. Removing all entries in which the age and test type do not match according to the given guidelines:
    a. Ages 5-8: Flamingo Balance Test, Plate Tapping Test
    b. Ages 9-20: Sit and Reach Test, 600m Run, Push Ups, 50m Dash,
3. Removing all entries in which the score of the student is invalid that is, outside the specified minimum and maximum limits for a student of that age and gender, taking the same test
4. Manipulating data to obtain the average score for each student
5. Plotting the bar graph with age on the x-axis and average score on the y-axis

Output:



Output: Bar Plot. Age versus average score obtained by students of all ages in the II term belonging to all schools in North West Delhi affiliated to the CBSE board. (Board : CBSE,

Zone:North, Region: Delhi West, State: Delhi (West), City: North West Delhi - A)

Rectifications to this edition of the graph

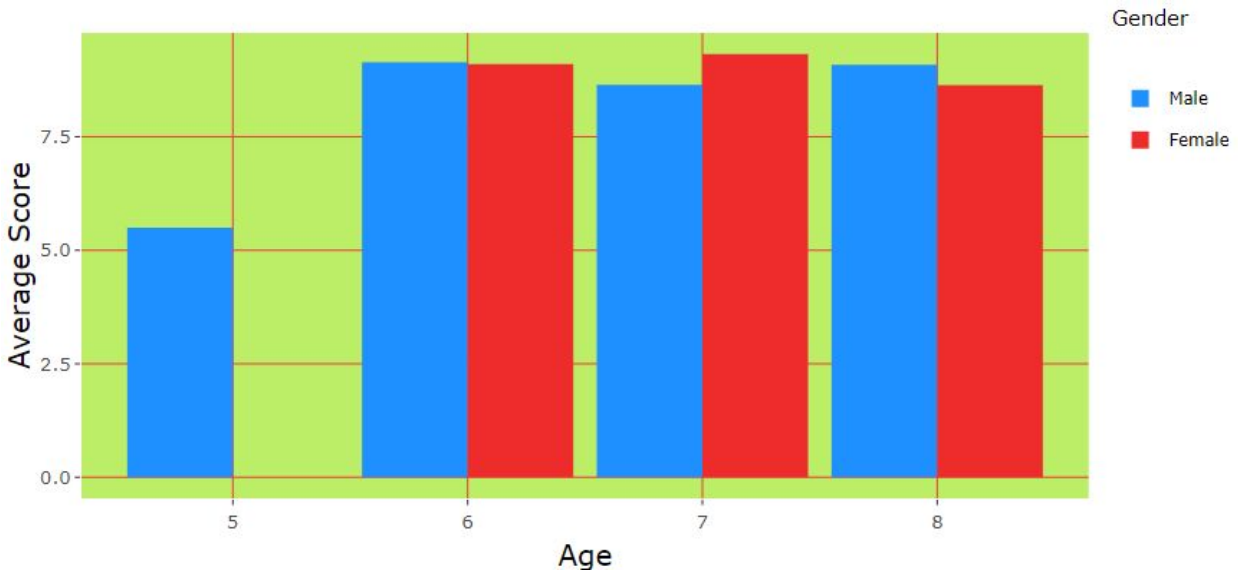
As such, the graph computes the average score by averaging the scores of all test types. However, the scores differ in their units based on the kind of test. Hence, it is incorrect to add the scores as such.

Two approaches were used to combat this error.

First, one graph each, was created for different test types on a single display using the facets method of ggplot2. This output was obtained.

Subsequently, it was determined that this is not as per the user requirements. User requirements included that test type should be passed as an input parameter by the user and within each age, the average score should be displayed separately for boys and girls. Owing to this, the written method was modified to include another input variable for the test type. Now, data would be extracted for only the particular test type and single graph for that test type would be displayed.

This output was obtained.



A web application for the Performance Indicator

The Detailed Procedure

Designing the appearance of the page

This includes setting the background colour, text font, colour and size. Using the setBackgroundColour() of shinyWidgets, the background colour was changed. Text formatting was done using HTML tags such as h1.

Creating the sidebar panel

This includes arranging input widgets on the sidebar panel so that the user can give input using them. Two main features which need to be present are that if the options displayed in a drop down are dependant on a previous input parameter, the options should be customized accordingly. Additionally, if an input parameter is in its default value of ‘All’, then any inputs dependant on it should not have any options other than ‘All’ or ‘Select’.

The following input parameters are listed in their order of hierarchy:

Joined Through > Select Board or School Chain

Zone > Region > State > City > School

Resolution of error conditions

There are three error conditions which need to be taken care of. The input parameters 'Joined Through', 'Age Group' and 'Test Type' are necessary to be chosen by the user to generate the graph. If one or more of them is not selected, an error notification pops up when the user attempts to generate the graph.

Speeding up the system

The graph displayed is an output which depends on the inputs: JoinedThrough, Board or School Chain Name, Zone, Region, State, City, School, Age Group, Test Type and Term. If the user changes the selected value of any of these, the entire computation for the graph occurs.

If the user wants to change the selected values of multiple inputs, then after each input value is changed, the graph will be regenerated and the user will be required to wait till the new graph appears to be able to change the next input parameter value.

This wastes a lot of time, while producing intermediate graph outputs which are not useful.

Hence, an action button was added and the graph was isolated from all inputs other than the action button. Therefore, the graph is now regenerated only when the action button is clicked.

Adjusting the y-axis range

An additional feature which was added was an adjustable y-axis which changes its range according to the range of values to be plotted on the y-axis that is, the maximum height of the bars in the bar plot.

Adding a graph description

The graph is accompanied with a description of the graph which details the input parameter values chosen by the user for the corresponding graph.

Running the App

The system should fulfill the requirements listed under the heading ‘System Requirements to setup the Shiny App’.

Open the ui.R and server.R files in R Studio.

Click on Run App

Fitness Indicator

Fitness Indicator graphs is divided into two categories Fitness Indicator Region-Wise and Fitness Indicator Age-Wise for both of them the same dataframe as thus to increase productivity common function is written to fetch the data frame.

Data frame is composed of particular attributes of the following tables of GoForFitGoForFit Database like Student Master , City Master , Senior Test Result , Region Master ,Board Chain Mapping,Board Master,Zone Master.

Dataframe is divided into seven categories L1(0-20 percentile),L2(21-40),L3(41-60),L4(61-70),L5(71-80),L6(81-90),L7(91-100).

Fitness Indicator Region-Wise

The function takes following as the input-

1.Joined Through(Board or Chain)

2.Board Name or Chain Name

3.Zone Name

4.Age

Brief information of Implementation

Connected to the live database of GoForFit using odbc and then from the entire database appropriate dataframe was extracted. Then constructed the line plots with the help of ggplot.

Flow Line

Task was to generate multi line plot with regions on x axis and percentage of students on y.

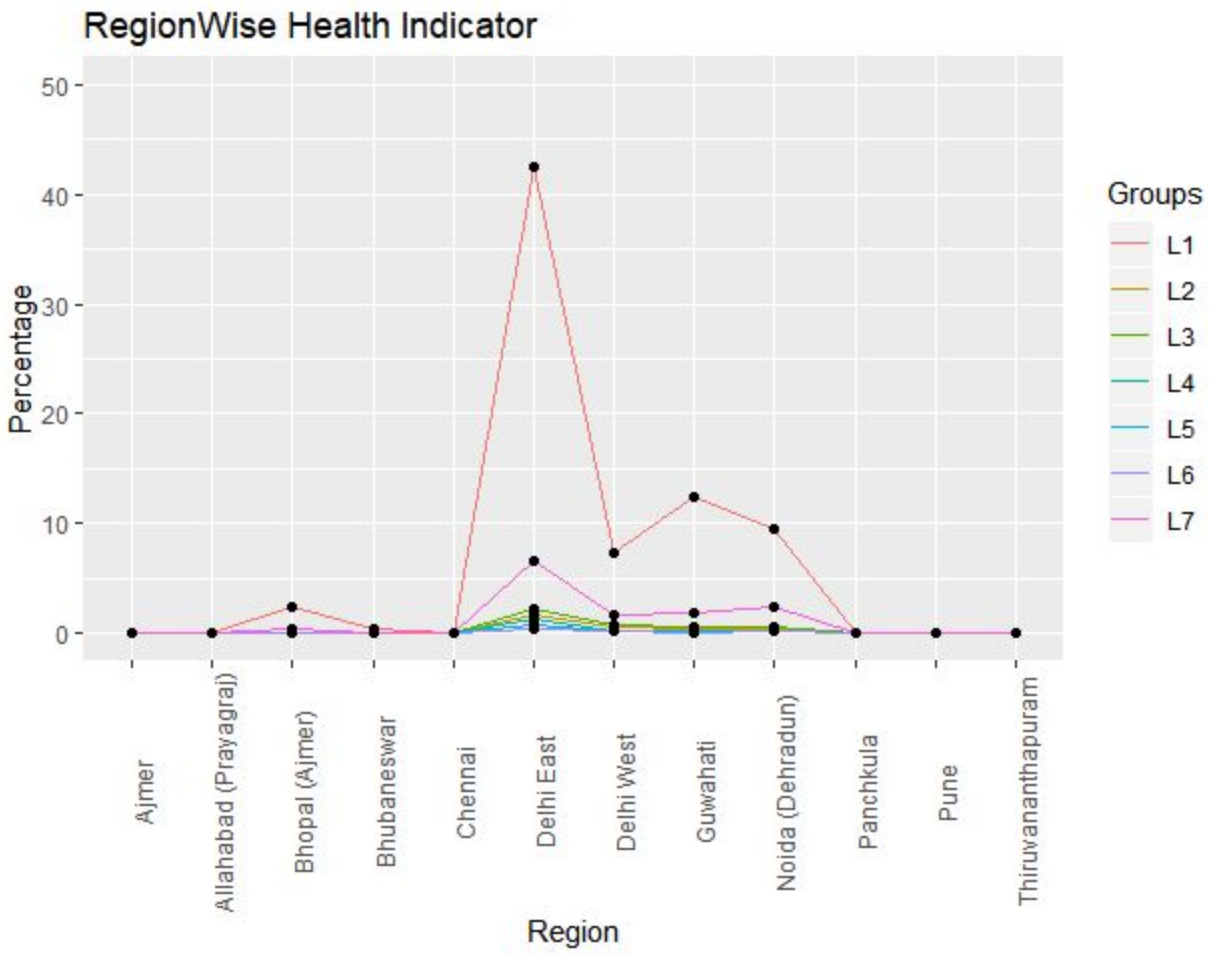
1. First from the proper tables of the GoForFit Database the appropriate and required attributes were extracted and merged in a dataframe.

2. Data was cleaned and filtered to meet the requirements.

3. Grouped the data into seven categories(L1-L7).

4. Then plotted the graph using ggplot where L1-L7 was the grouping factor and x axis contained names of regions and y axis contained the percentage of students.

Output



Fitness Indicator Age-Wise

The function takes following as the input-

1.Joined Through(Board or Chain)

2.Board Name or Chain Name

3.Zone Name

4.Region Name

5.State Name

6.City Name

7.School Name

8.Term

Brief information of Implementation

Connected to the live database of GoForFitGoForFit using odbc and then from the entire database appropriate dataframe was extracted. Then constructed the line plots with the help of ggplot.

Flow Line

Task was to generate multi line plot with percentage of students on y axis and age on rx.

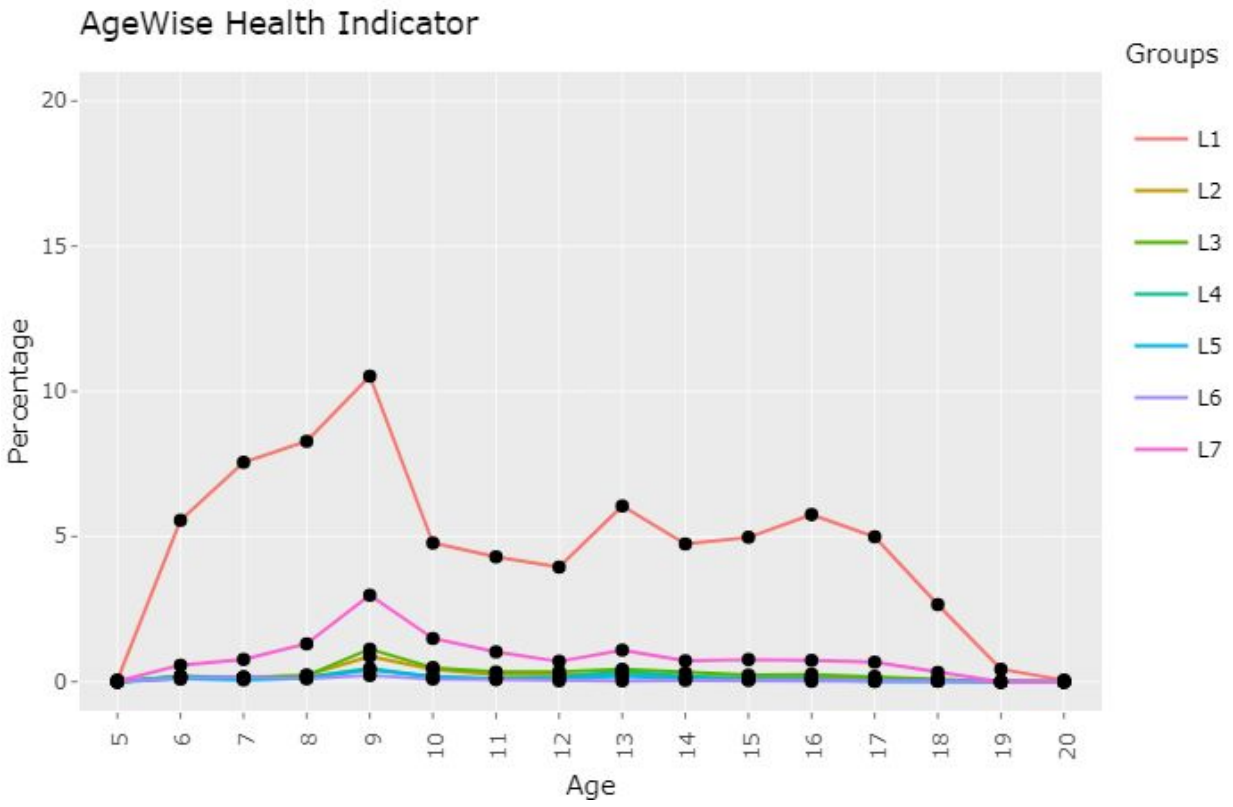
1. First from the proper tables of the GoForFit Database the appropriate and required attributes were extracted and merged in a dataframe.

2. Data was cleaned and filtered to meet the requirements.

3. Grouped the data into seven categories(L1-L7).

4. Then plotted the graph using ggplot where L1-L7 was the grouping factor and x axis contained names of regions and y axis contained the percentage of students.

Output



Fitness Indicator Web App

It is a standalone, interactive web app designed using shiny package and is hosted in RStudio itself.

Web-Apps have essentially two parts ,UI(user interface) and server, UI is basically the front-end input portion of the app which contains titles and drop-down menus/other input area for the application.

Server renders the output in this case the graphs generated.

Task was to design two web-apps Region-Wise Fitness Indicator and Age-Wise Fitness Indicator.

Region-Wise

Generates graph from the live database after going through the following four filters -

1. Joined Through
2. Board/Chain
3. Zone
4. Age(5-20)

Joined Through contains two options board and chain indicating how the student is connected to the database. It is the only portion of the code which is hard coded.

UI contains conditional panels i.e the next drop-down/input options are based on the previous selection.

In easy words if we select board the next dropdown shows the list of the boards of assessed students in the database.

Fitness Indicator

Joined Through

Board

Board

CBSE

Zone

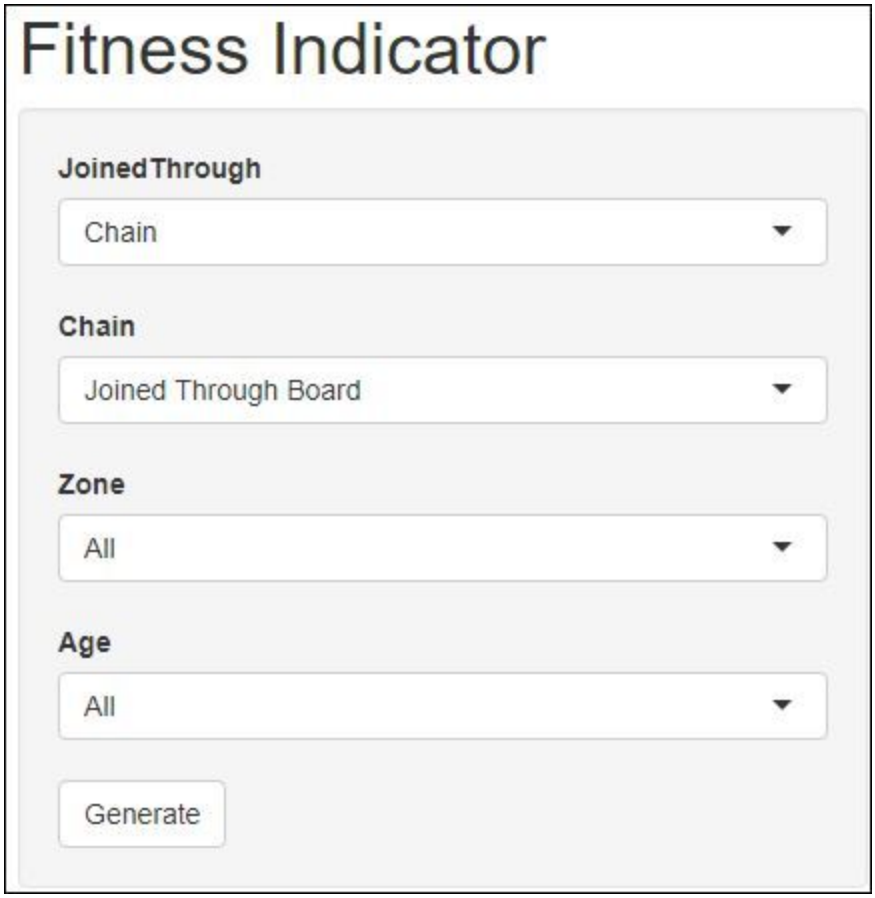
All

Age

All

Generate

If we select chain the UI changes accordingly

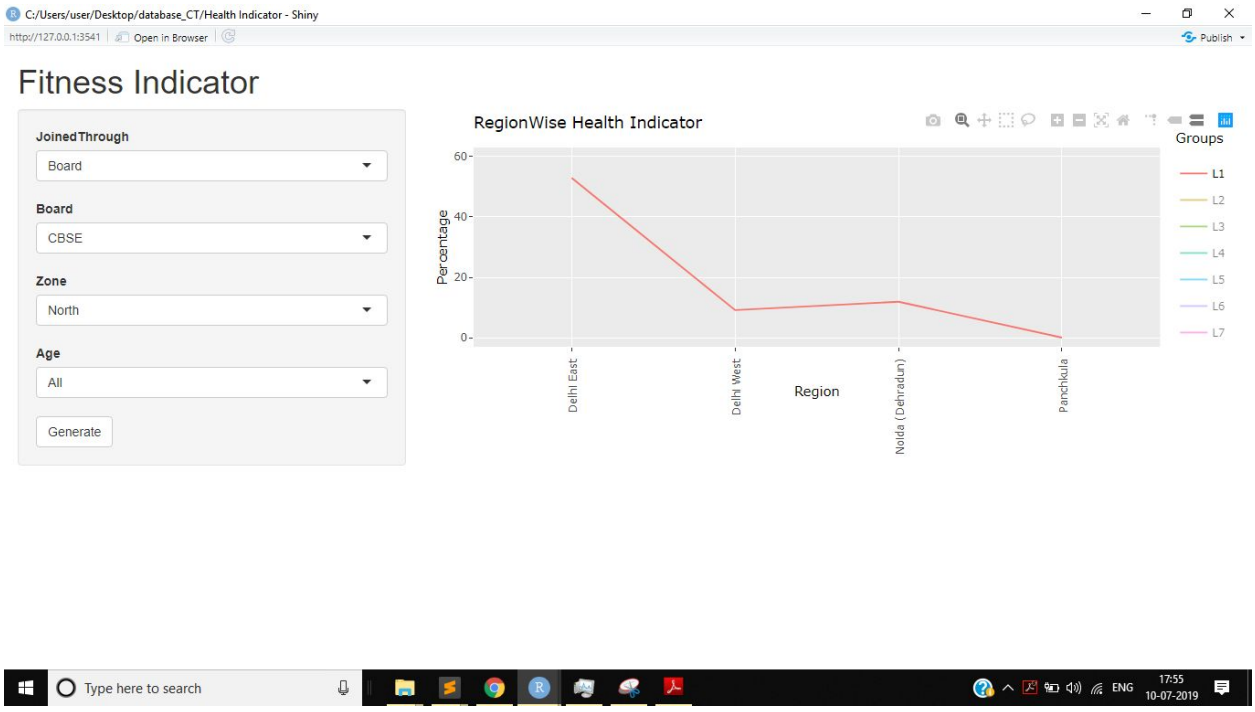


Output

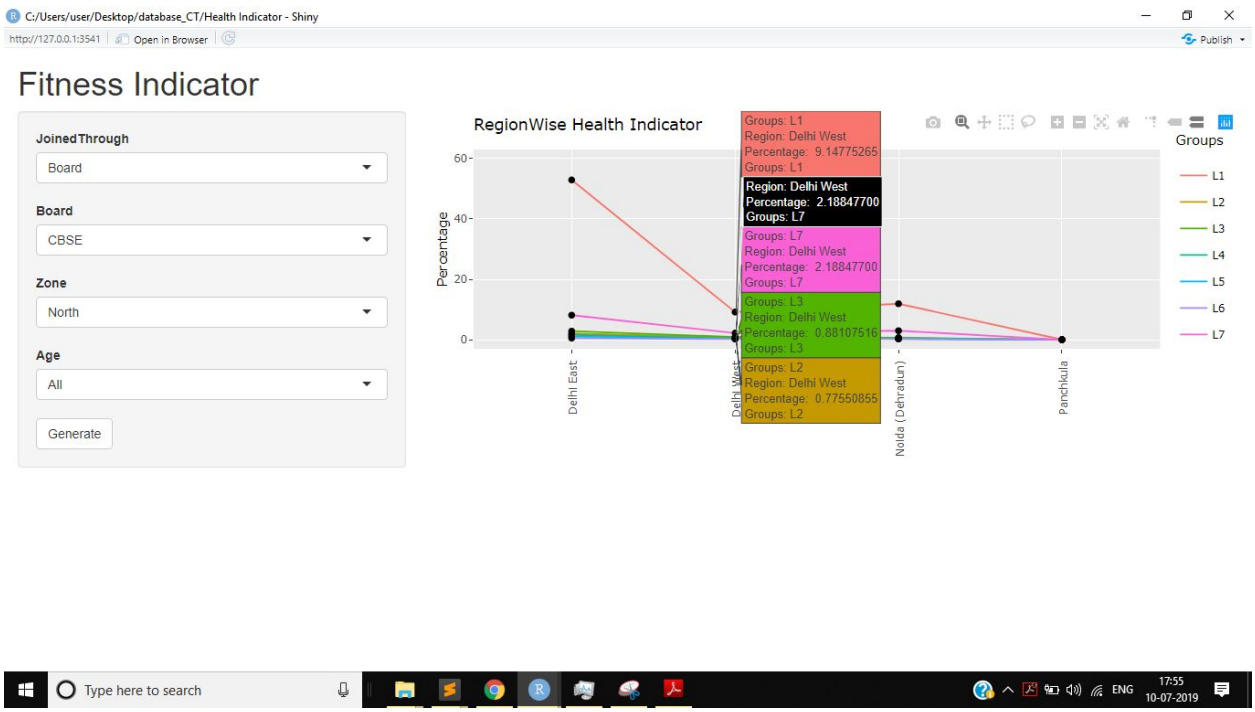
Web-app contains an action button with the name “Generate” server renders output only when we click this button.

Graph generated are interactive graphs which contains parameter isolation,tool-tip, hovering tool-tip(closest and comparative) etc.

The Graphs have adjustable y-axis which means the maximum of the graphs change according to the maximum value in that particular graph.



(Isolated Parameter)



(Hovering Tool-Tip)

Age-Wise

Contains seven filters which increases the density of data filtering. Most of the working is similar to the Region-Wise Application key difference being more filters ,and dynamic and, reactive changes in options of drop-down menu based on previous selection.

The seven filters are -

1. Joined Through
2. Board/Chain
3. Zone
4. Region
5. State
6. City
7. School

Working of the reactive dropdown menus

The dropdown options change according to the previous selections the filter is layer wise i.e Zonal level then Regional and so on. Thus if we select the ”All” option in Zone we can not select any other option in Region and the lowers layers. But if we select North Zone the regions gets filtered accordingly and only the regions in that particular zone are shown the same thing happens for lower layers.

The Performance Indicator (based on Average Percentile)

It helps in the better understanding of performance of assessed students in the GoForFit database.

Students performance graphs are divided into two categories Average Percentile Region-Wise and Average Percentile Age-Wise.

Procedure to calculate the average percentile of the students:-

If the student age is less than 8 we take 2 tests into account. These are Plate tapping test and flamingo test.We find the average percentile of the students for these tests and then average percentile for a region or age depending on the graph we chose to plot.

If student age is greater than 8 we take 5 tests into account those are sit and reach,600 meters run,50 meters dash,partial curl up and push ups.We find the average percentile of the students for these graphs and then average percentile for a region or age depending on the graph we chose to plot.

To plot above graphs we need to get the percentile of the students,their age, zone, region, state, city e.t.c they belong to from the tables of GoForFitGoForFit database like StateMaster, RegionMaster, StudentMaster, SchoolMaster,SeniorTestResults e.t.c.

Average Percentile Region-Wise

The function takes following as the input-

(All inputs are optional)

1.Joined Through(Board or Chain)

2.Board Name or Chain Name

3.Zone Name

4.starting Age

5.Ending Age

6.Term

Brief information of Implementation

Connected to the live database of GoForFit using RODBC package and then from the entire database appropriate data frames were extracted and were joined to construct bar plots with the help of ggplot2() package.

Flow Line

Task was to generate bar plot with average percentile of students on y-axis and regions on the x-axis.

1. First from the tables of the GoForFitGoForFit Database the appropriate and required attributes were extracted and merged into a dataframe.

2. Data was cleaned and filtered to meet the requirements of what the user wants that he passed through the arguments into the function.

3.Then the resulting data frame was divided into 2 dataframes one data frame consisting of all the information about the boys and the another one about the girls by dividing the sum of percentile for each student and dividing it by 2 if the age of that student is less than else by 5 because we are analysing the results for 2 and 5 tests for the students of age <8 and >=8 respectively

4.Then we merge the 2 data frames by calculating the average percentile for each region.

5.Then we plot the resulting dataframe using the package ggplot2()

Connecting to the database:

Connect to the database using the method: dbConnect() by filling in the required fields: Driver, Server, Database, Username, Password and Port

Relations accessed:

SeniorTestResults: for StudentID, Age, TestTypeID ,Percentile,CreatedOn

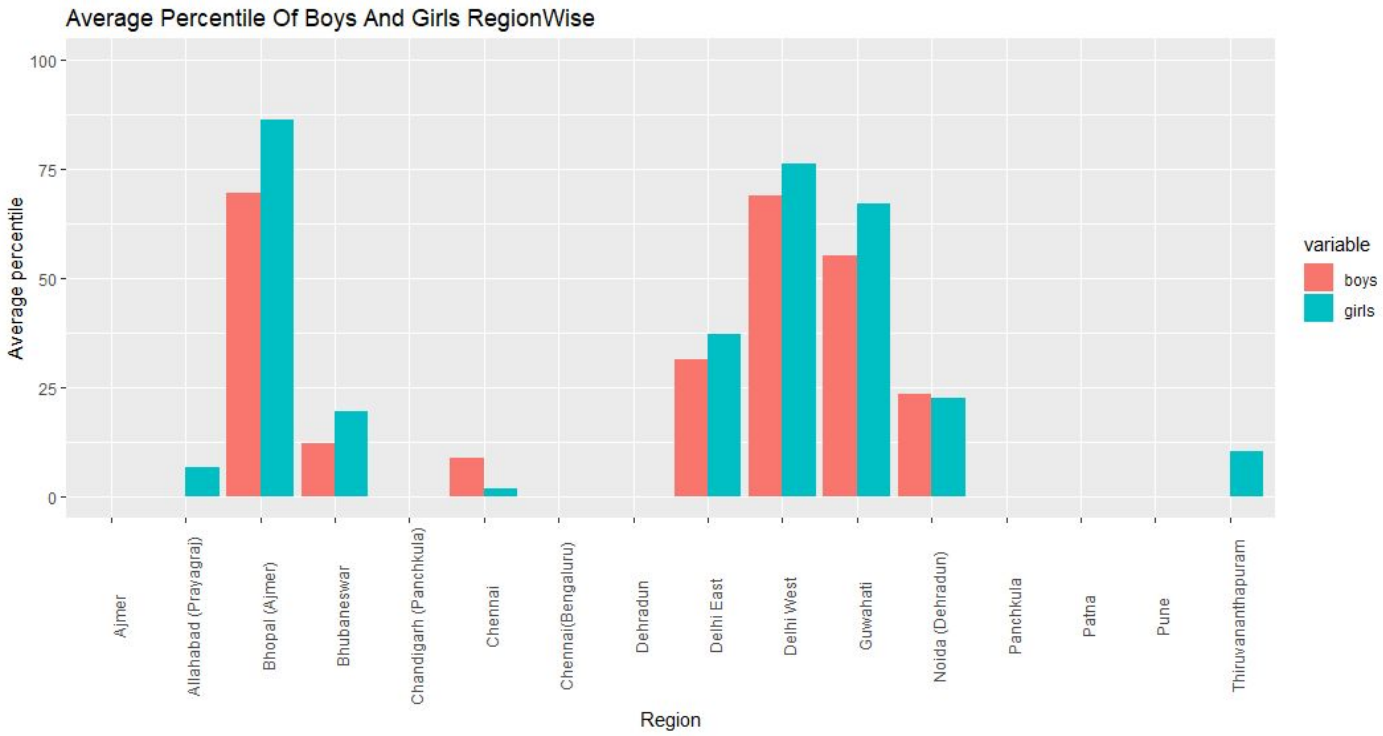
StudentMaster: for StudentID, StudentName, Gender, SchoolID

SchoolMaster: for SchoolID, SchoolName, BoardID/SchoolChainID, ZoneID

BoardMaster: for BoardID, BoardName

ZoneMaster: for ZoneID, ZoneName

Output



A web application for the Region wise Performance Indicator(Average Percentile)

The Detailed Procedure

Shiny Package was installed using install.packages() method and then I activated the package using library() method .I used fluid page which consists of 2 components those are Sidebar panel and main panel in the sidebar layout and the title panel to give the heading for the page.

Creating the sidebar panel

This includes arranging input widgets on the sidebar panel so that the user can give input using them. Two main features which need to be present are that if the options displayed in a drop down are dependant on a previous input parameter, the options should be customized accordingly. Additionally, if an input parameter is in its default value of 'All', then any inputs dependant on it should not have any options other than 'All'

The following input parameters are listed in their order of hierarchy (dependent on other input):

Joined Through > Select Board or School Chain > select zone

The other inputs are:

Select Starting Age > Select ending Age > Select Term

The default value for Select Board or School Chain,select zone, Select Term inputs is "All" and

for

Select Starting Age input is 5 and for Select ending Age input is 5.

When would UI display an output

The output graph will appear in the main panel.

The graph displayed is an output which depends on the inputs: JoinedThrough, Board or School Chain Name, Zone, Starting Age, Ending Age,Term. If the user changes the selected value of any of these, the entire computation for the graph occurs.

If the user wants to change the selected values of multiple inputs, then after each input value is changed, the graph will be regenerated and the user will be required to wait till the new graph appears to be able to change the next input parameter value.

This wastes a lot of time, while producing intermediate graph outputs which are not useful. Therefore the default value of Ending Age is 5 because the ending age cannot be 5. So whenever the user will change the ending age from 5 to other value whatever the input he has given up to that point will be taken as the input for the function discussed a few pages before.

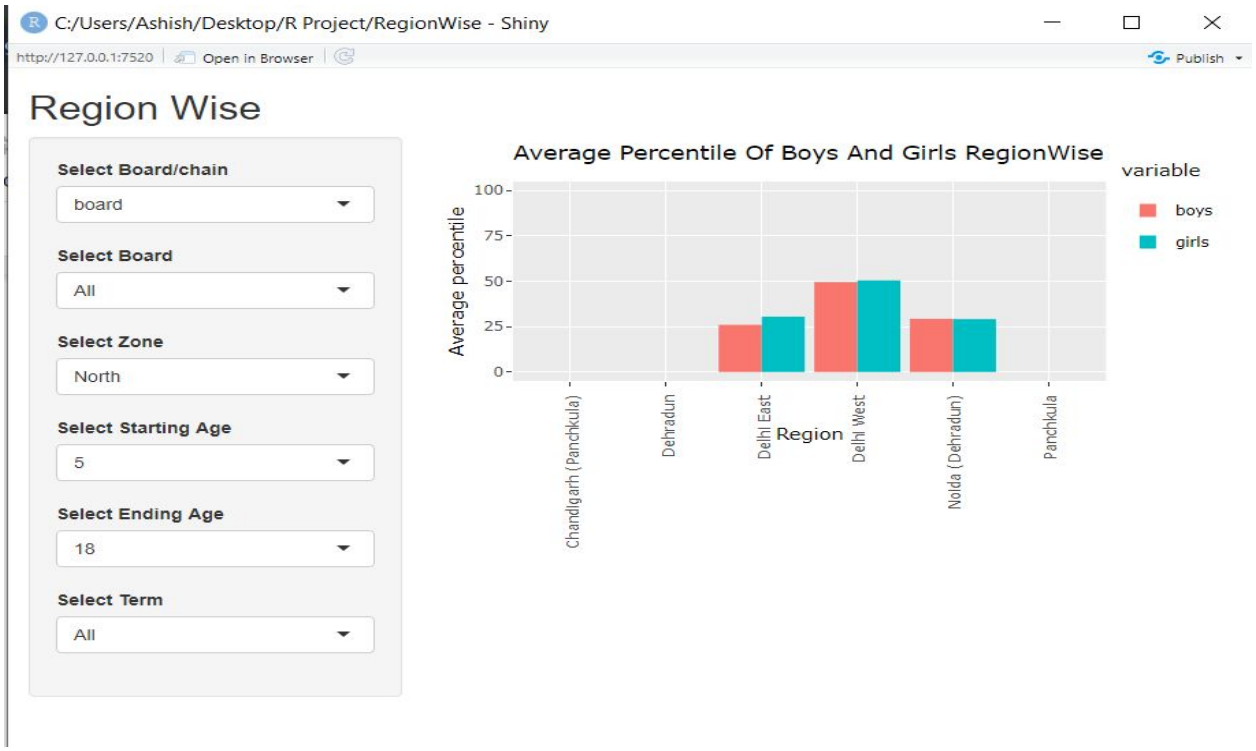
Running the App

The system should fulfill the requirements listed under the heading ‘System Requirements to setup the Shiny App’.

Open the ui.R and server.R files in R Studio.

Click on Run App

Pic for a sample input



Average Percentile Age-Wise

The function takes following as the input:-

(All inputs are optional)

1.Joined Through(Board or Chain)

2.Board Name or Chain Name

3.Zone Name

4.Region Name

5.State Name

6.City Name

7.School Name

8.Starting Age

9.Ending Age

10.Term

Brief information of Implementation

Connected to the live database of GoForFitGoForFit using RODBC package and then from the entire database appropriate data frames were extracted and were joined to construct bar plot the help of ggplot2() package

Flow Line

Task was to generate bar plot with average percentile of students on y-axis and age from 5 to 20 on the x-axis.

1. First from the tables of the GoForFit Database the appropriate and required attributes were extracted and merged into a dataframe.

2. Data was cleaned and filtered to meet the requirements of what the user wants that he passed through the arguments into the function.

3.Then the resulting data frame was divided into 2 dataframes one data frame consisting of all the information about the boys and the another one about the girls by dividing the sum of percentile for each student and dividing it by 2 if the age of that student is less than else by 5 because we are analysing the results for 2 and 5 tests for the students of age <8 and >=8 respectively

4.Then we merge the 2 data frames by calculating the average percentile for each age.

5.Then we plot the resulting dataframe using the package ggplot2()

Connecting to the database:

Connect to the database using the method: dbConnect() by filling in the required fields: Driver, Server, Database, Username, Password and Port

Relations accessed:

SeniorTestResults: for StudentID, Age, TestTypeID ,Percentile,CreatedOn

StudentMaster: for StudentID, StudentName, Gender, SchoolID

SchoolMaster: for SchoolID, SchoolName, BoardID/SchoolChainID, ZoneID, RegionID, StateID, CityID

BoardMaster: for BoardID, BoardName

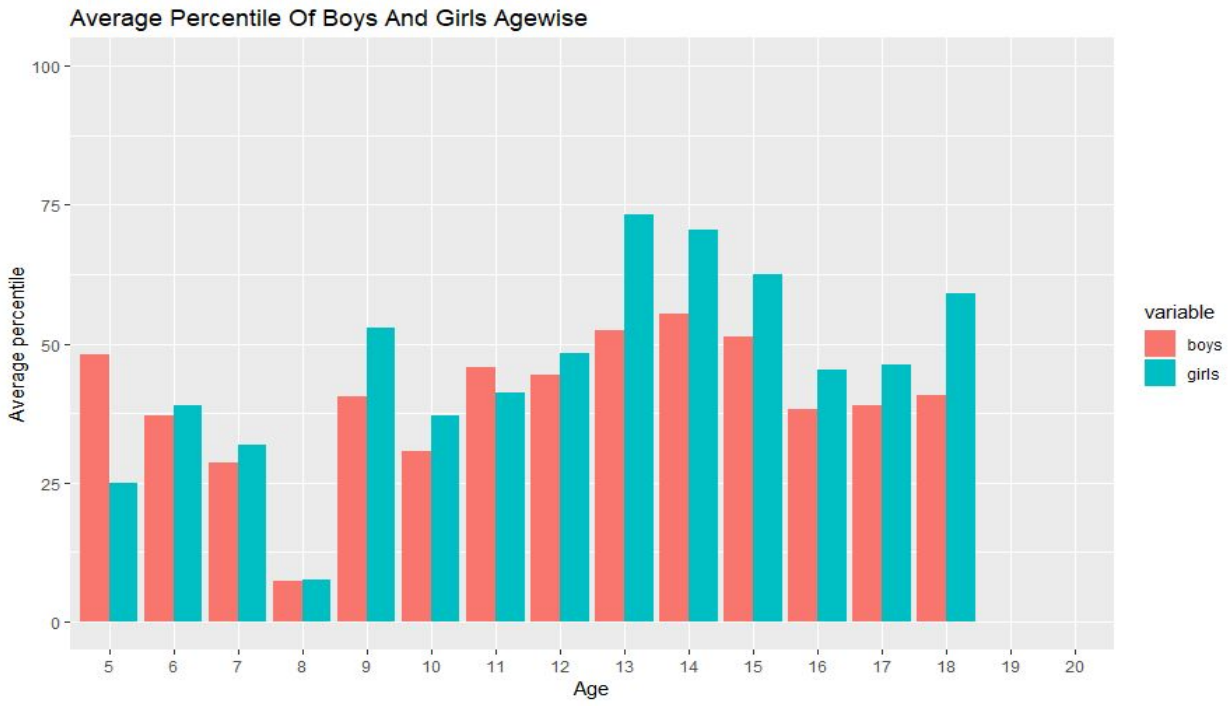
ZoneMaster: for ZoneID, ZoneName

RegionMaster: for RegionID, RegionName

StateMaster: for StateID, StateName

CityMaster: for CityID, CityName

Output



A web application for the Age wise Performance Indicator(Average Percentile)

The Detailed Procedure

Shiny Package was installed using install.packages() method and then I activated the package using library() method .I used fluid page which consists of 2 components those are Sidebar panel and main panel in the sidebar layout and the title panel to give the heading for the page.

Creating the sidebar panel

This includes arranging input widgets on the sidebar panel so that the user can give input using them. Two main features which need to be present are that if the options displayed in a drop down are dependant on a previous input parameter, the options should be customized accordingly. Additionally, if an input parameter is in its default value of 'All', then any inputs dependant on it should not have any options other than 'All'

The following input parameters are listed in their order of hierarchy:

Joined Through > Select Board or School Chain > select zone > select region >

Select state > Select city > Select School

The other inputs are:

Select Starting Age > Select ending Age > Select Term

The default value for Select Board or School Chain,select zone, Select Term, ,Select state, Select a city ,Select School inputs is "All" and for

Select Starting Age input is 5 and for Select ending Age input is 5.

When would UI display an output

The output graph will appear in the main panel.

The graph displayed is an output which depends on the inputs: JoinedThrough, Board or School Chain Name, Zone, Starting Age, Ending Age,Term,Select state, Select a city ,Select School. If the user changes the selected value of any of these, the entire computation for the graph occurs.

If the user wants to change the selected values of multiple inputs, then after each input value is changed, the graph will be regenerated and the user will be required to wait till the new graph appears to be able to change the next input parameter value.

This wastes a lot of time, while producing intermediate graph outputs which are not useful. Therefore the default value of Ending Age is 5 because the ending age cannot be 5. So whenever the user will change the ending age from 5 to other value whatever the input he has given up to that point will be taken as the input for the function discussed a few pages before.

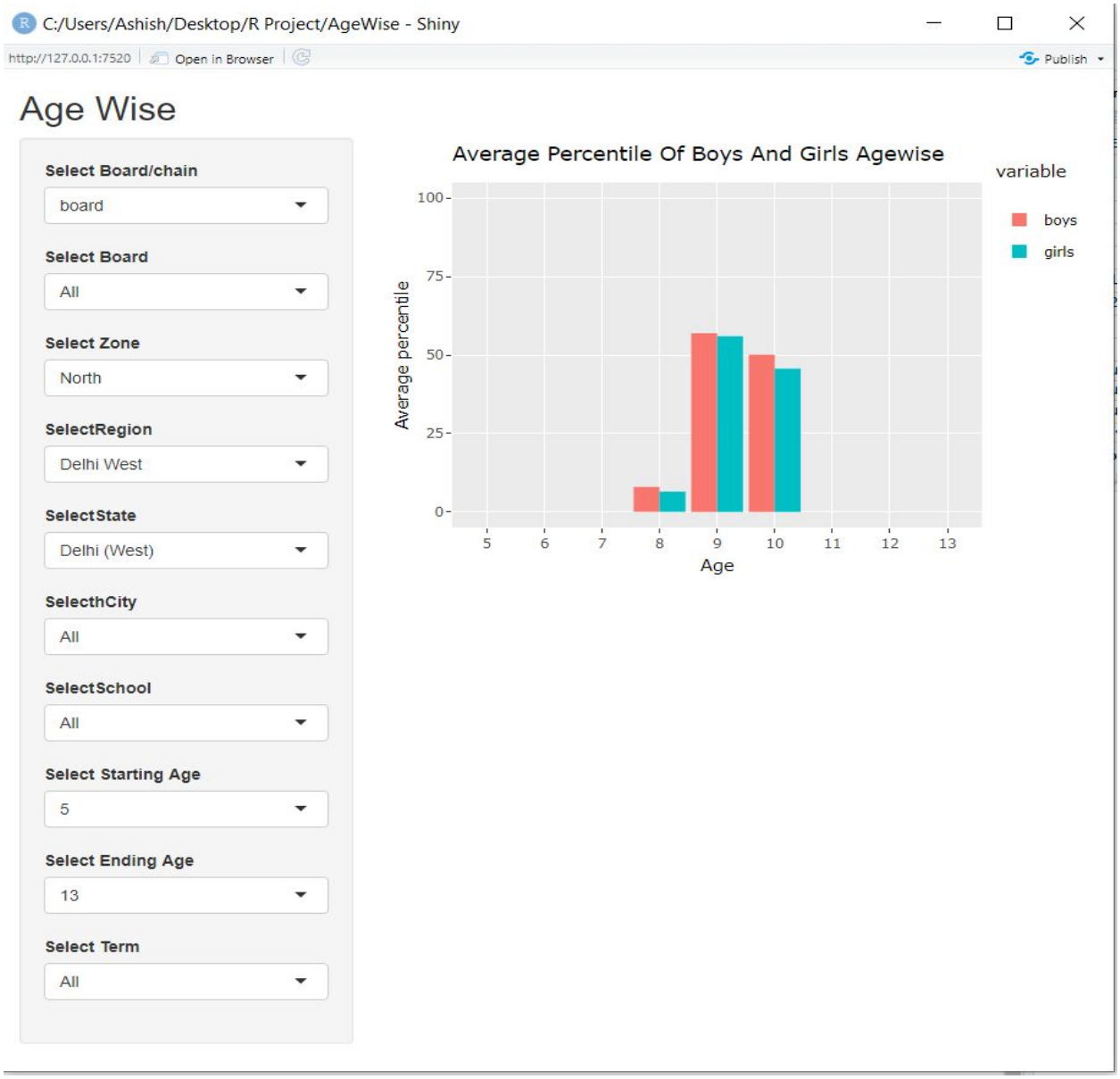
Running the App

The system should fulfill the requirements listed under the heading ‘System Requirements to setup the Shiny App’.

Open the ui.R and server.R and hola.R files in R Studio.

Click on Run App

Pic for a sample input



Conclusion

The importance of sports and fitness in one's life is invaluable. Playing sports improves team spirit skills, develops analytical thinking, leadership skills, goal setting and risk taking.It brings never giving up attitude in an individual.Fit and healthy individual leads to a strong nation.

Using the R programming language we learned how to process large data and plot graphs so that one can easily make inferences out of it.

By plotting graphs between various parameters related to fitness we can understand trends related to health of students in different schools,regions,zones in our country so that people can take steps to improve it. For example, in one of the individual tasks we made health indicator graph which tells percentage of students underweight , normal, overweight and obese both region wise and age wise .Such information can be very important for government, schools and parents.

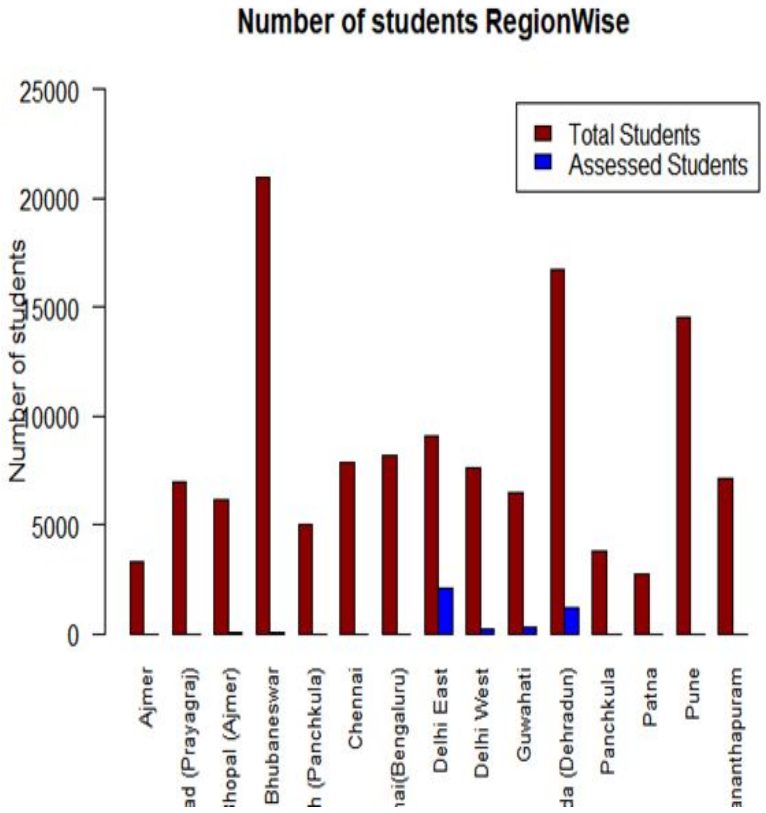
Conclusion for Task 1: Bar Graph Representing Percentage of Students vs Percentile

Most of the students (53%) are have percentile in range 0 to 20.After that 28% percent students are having percentile between 80 to 100.

Conclusion for Task 2 : Average Percentile Graph

Age 7 students have the highest percentile, followed closely by age 9 and 6. The poorest performance is that of age 8 students, and age 17 next.

Conclusion for Task 3: Total students and total assessed students.



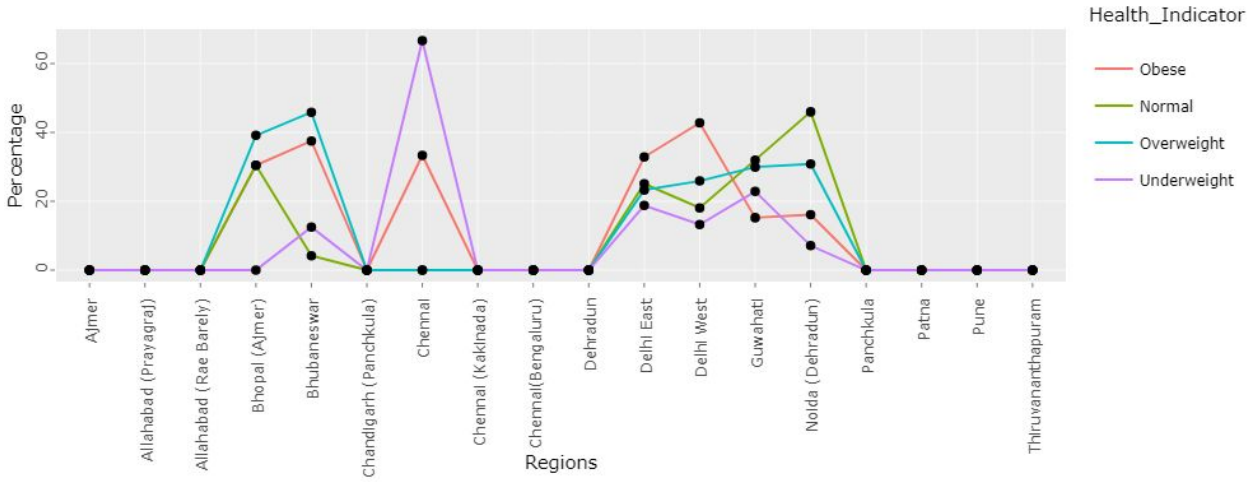
1. Delhi East has the highest number of assessed students showing that it is the best worked/tested .
2. Bhubaneswar has the lowest ratio of assessed to total students .
3. Patna has the least dense database.

Conclusion for Task 4: Average Percentile for Different Age Groups Graph

1)The age group from 9-11 has highest average percentile and hence performing better

2)The age groups of 0-5 and 16-18 must be targeted to get better performances.

A Sample Conclusion from the Health Indicator:

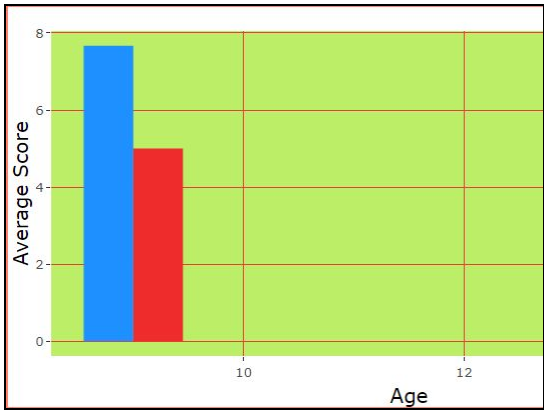


From the above graph we can see that chennai region has the highest percentage of underweight people.Noida region has the highest percentage of Normal people. Bhubaneswar region has the highest percentage of overweight people.Delhi West has the highest percentage of obese people.

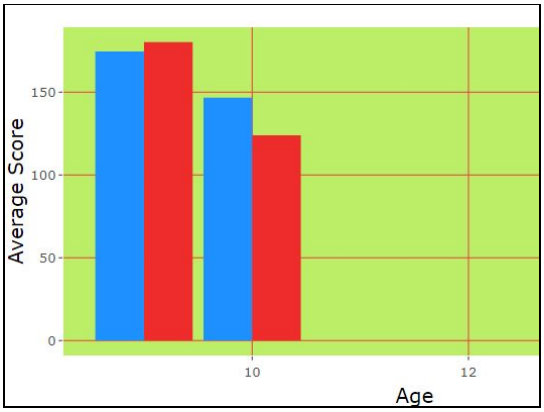
A Sample Conclusion from the Performance Indicator(Average Score):

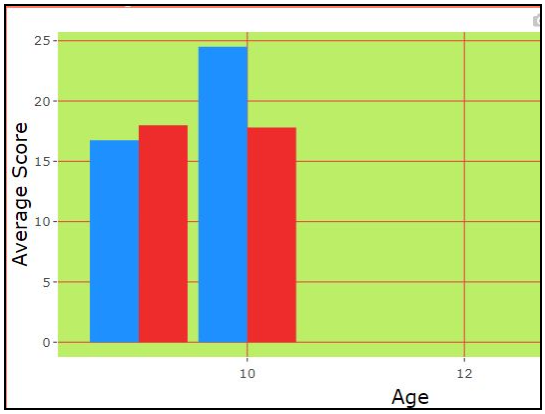
The following are graphs for the Lancer's Convent School in North West Delhi-A for students falling in the age group 9-14.

Push Up



Sit and Reach Test





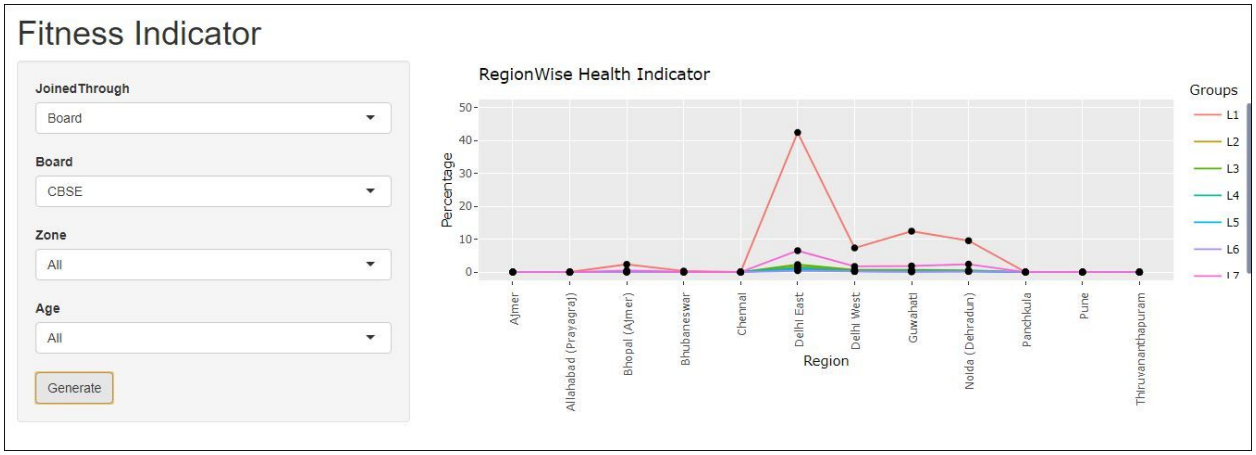
50m Dash

One can conclude from these sets of graphs that boys have a higher average score than girls in most of the tests. There are only a few instances in which girls score higher and the difference is not major.

Likewise, we can draw various conclusions from different graphs for different schools, cities, states, regions and zones, affiliated to different boards/school chains and falling in various age groups.

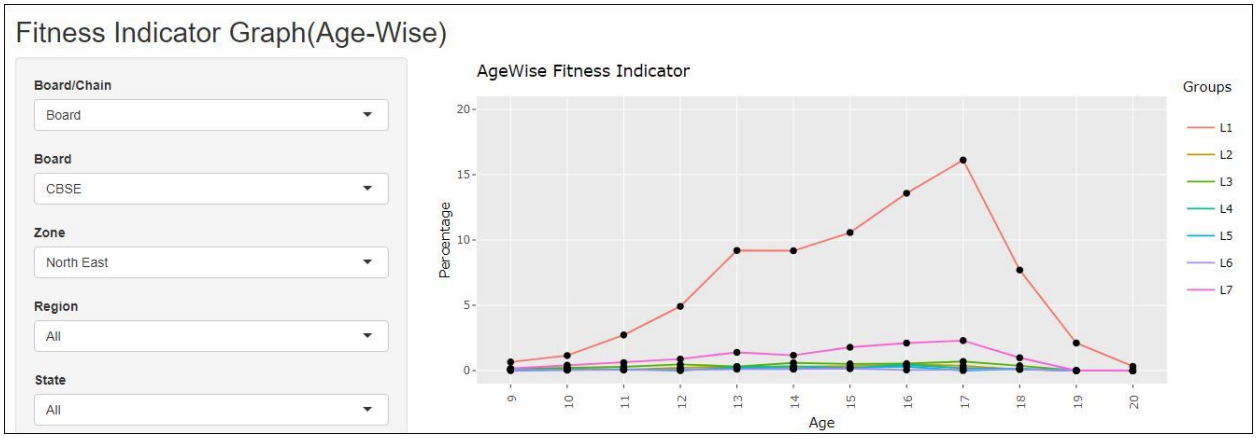
A Sample Conclusion from the Fitness Indicator:

Region-Wise(All the zones)



1. Delhi East has the highest percentage in both the least fit(L1) and the fittest(L7) categories.
2. L1 leads to be exceeding in this as well as most of other graphs generated from the web-app showing the lack of health awareness in schools.

Age-Wise(For North-East Zone)



1. In North East Region L1 line plot exceeds showing the zone has an alarming number of students in least fit category.
2. At the age of 17 the fitness level seems to be lowest after which it increases drastically.

A Sample Conclusion from the Performance Indicator(Average Percentile):

General case for Region wise:-

1)Bhopal region is the stand out performer for GoForFit having more percentile than any other region

2)Chennai region is the worst performer and should be given more focus

General case for Age wise:-

1)The students of Age 8 has a lower percentile than the other ages and must be targeted for better performances.

2)Boys of age 13 and girls of age 5 have performed better than any other ages of their category

References

1. https://www.rdocumentation.org/
2. Google Images
3. https://www.guru99.com/r-vs-python.html
4. https://www.techrepublic.com/article/r-vs-python-which-is-a-better-programming-language-for-data-science/
5. www.udemy.com

Glossary

Data Analytics - The examination of data to reach important conclusions from it

Data Cleaning- Removal of inconsistent data to 'cleanse' it so that validity and consistency are ensured.

R - programming language for statistical computing

Python - interpreted, high-level, general-purpose programming language

RStudio - a open-source integrated development environment for R for statistical computing.

SQL - a programming language for database management

Packages - Packages are collections of R functions, data, and compiled code

SQL Server : Database Engine

Microsoft SQL Server Management Studio(SSMS): GUI Administration tool for working against the SQL server engine

RODBC: An R Studio package for connecting to the database

odbcDriverConnect() : To connect to the database. An RODBC method.

sqlGetResults() : Returns the results of a query (in the form of a data frame) associated with the passed database connection handle. An RODBC method.

sqlDrop() : Removes the table SQL table (if permitted). An RODBC method.

sqlSave() : Write or update a table in the database. An RODBC method.

RODBCext: An R Studio package for accessing the database

sqlPrepare() : To pass a query (can be parameterised) via the database connection handle. An RODBCext method.

sqlExecute() : To pass parameters to a query prepared by sqlPrepare(). An RODBCext method.

ggplot2: An R Studio package for creating graphical visualisations.

ggplot() : To initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden. A ggplot2 method.

geom_bar() : To create a bar chart. A ggplot2 method.

position_dodge() : Dodging preserves the vertical position of a geom while adjusting the horizontal position. A ggplot2 method.

scale_fill_manual() : To specify our own set of mappings from levels in the data to aesthetic values. A ggplot2 method.

labs() : To label the graph. A ggplot2 method.

theme() : To format visual parameters of the graph such as background colour of the graph, text size and so on. A ggplot2 method.

plotly: An R Studio package for making graphical visualisations more interactive.

ggplotly() : To convert a ggplot2::ggplot() object to a plotly object. A plotly method.

plotlyOutput() : To output and render functions for using plotly within Shiny applications in ui.R. A plotly method.

renderPlotly(): Output and render functions for using plotly within Shiny applications in server.R. A plotly method.

shiny: An R Studio package for making GUI

shinyUI() : to register a user interface with Shiny. A shiny method.

fluidPage() : To scale components of a Shiny app in realtime to fill all available browser width.

titlePanel() : Create a panel containing an application title.A shiny method.

sidebarLayout() : Create a layout with a sidebar and main area. A shiny method.

sidebarPanel() : Create a sidebar panel containing input controls that can in turn be passed to sidebarLayout. A shiny method.

selectInput() : Create a select list that can be used to choose a single or multiple items from a list of values. A shiny method.

uiOutput() : Render a reactive output variable within an application page. A shiny method.

actionButton(): Creates an action button or link whose value is initially zero, and increments by one each time it is pressed. A shiny method.

textOutput() : Render a reactive output variable as text within an application page. A shiny method.

reactive() : Wraps a normal expression to create a reactive expression. A shiny method.

renderUI(): Renders reactive HTML using the Shiny UI library. A shiny method.

observeEvent(): Respond to "event-like" reactive inputs, values, and expressions. A shiny method.

showNotification(): To display a notification in a Shiny app. A shiny method.

renderText(): Makes a reactive version of the given function to turn its result into a single-element character vector. A shiny method.

isolate(): Executes the given expression in a scope where reactive values or expression can be read, but they cannot cause the reactive scope of the caller to be re-evaluated when they change. A shiny method.

shinyWidgets: An R studio package for adding widgets to the GUI.

setBackgroundColor() : To change the background colour of our shiny app. A shinyWidgets method

ODBC: An R Studio package for connecting to and managing the database.

odbc(): To connect to the database.

dbGetQuery(): To send query, retrieve results and then clear result set

DPLYR: An R Studio package for data manipulation

mutate(): To add new variables and preserves existing; transmute drops existing variables.