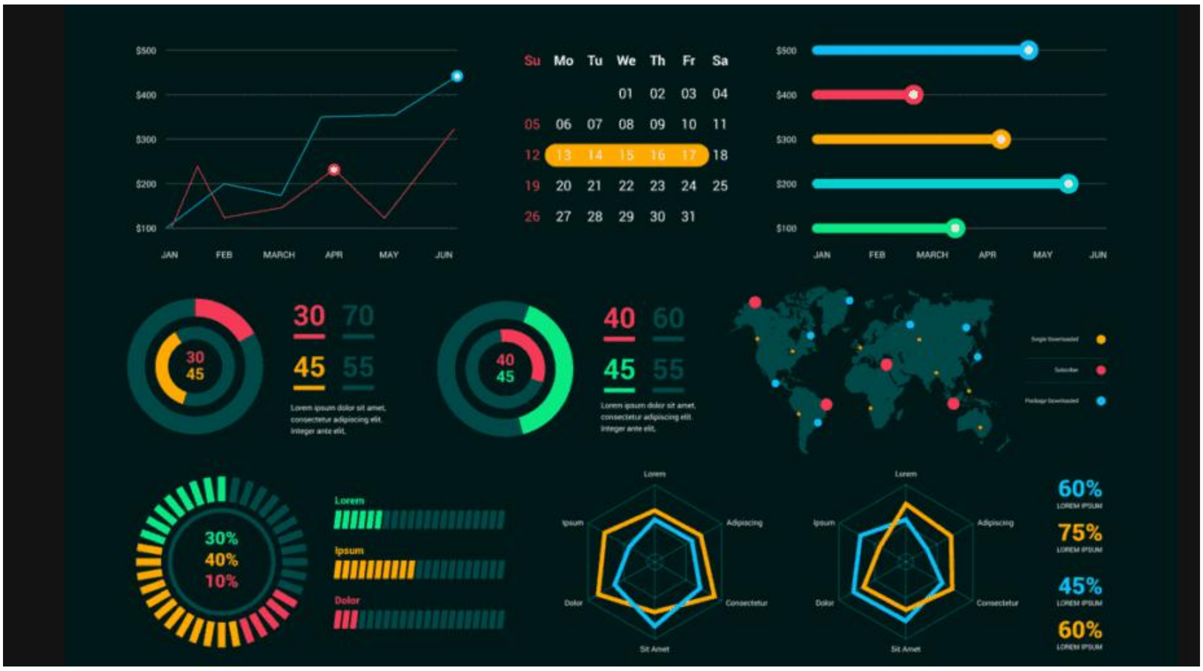
Data Visualization (CSE3020)
Slot: D1
Review 3

Electric Vehicles

A thorough analysis into the world of electric vehicles

Submitted to
Dr. Lydia Jane G
Vellore Institute of Technology, Vellore



Brought to you by :-

Shivam Raj, 20BCE0819

Ananya Singh, 20BCE0785

1. Our Data set names are:
   -Electric Vehicle Analysis (ElectricCarData_Clean.csv)
   -black carbon concentration dataset (Dataset.csv)

2. Our dataset was downloaded from the following sites:
   - https://www.kaggle.com/tyrawls/electric-vehicle-analysis
   - https://github.com/environmentalstatistics/Shiny-1/blob/master/Dec-2017.csv

Introduction:

As we know our environment is degrading day by day due to pollution and the major contribution to this is done by fuel vehicles. So to sustain our environment and to save our mother earth from depleting further, we need to do something to decrease the pollution levels.

Objective:

To understand the increasing levels of carbon emission in environment on daily monthly and yearly basis and see the trends over various time periods thereby focusing on the immediate need to switch to Electric vehicles.
So one of the major initiatives we as youth can take is by reducing the use of fuel vehicles. People say to avoid cars and travel by either a public transport bus or use a cycle instead isn't a permanent solution to this. We all have developed a kind of lifestyle that we can't live without vehicles.
So we have analysed data in such a way that people go for electric cars instead of fuel vehicles. We have tried to cover each and every aspect that will lead the user in choosing a perfect car for themselves!

General Overview :

Conventional cars use internal combustion engines (ICEs) that generally run on fossil fuels like gas or diesel. EVs use one or more electric motors powered by rechargeable lithium-ion batteries, the same kinds of batteries that power smartphones and laptops. And like electronic devices, EVs plug into external power sources for charging. Other types of batteries rely on regenerative braking for charging, or generating electricity from the vehicle's frictional energy. In addition to being less polluting than fuel engines, lithium-ion batteries often work more efficiently. Many have a guaranteed life span of 8-10 years.
Like stopping at a gas station, a network of charging stations gives EV batteries access to power on the road. An EV's driving range between stations is dependent on its battery life. Extreme driving conditions or weather can also affect an EV's range as they use more energy to compensate.
Because EVs do not rely on fossil fuels for power, they may not have certain components that ICE vehicles do. For instance, parts such as fuel lines, fuel tanks, and tailpipes. This means that most EVs do not emit carbon dioxide emissions (CO_2), which helps reduce air pollution.
An electric car can be a great way for you, as a consumer, to save a lot of money on gas. However, there are so many different reasons why you should invest in an electric car in the modern-day of technology.
1. No Gas Required
2. More Convenient

3. Savings
4. No Emissions
5. Popularity
6. Safe to Drive
7. Cost-Effective
8. Low Maintenance
9. Reduced Noise Pollution
10. Battery Life & Cost
11. Easy Driving

The major benefit of electric cars is the contribution that they can make towards improving air quality in towns and cities. With no tailpipe, pure electric cars produce no carbon dioxide emissions when driving. This reduces air pollution considerably.
Put simply, electric cars give us cleaner streets making our towns and cities a better place to be for pedestrians and cyclists. In over a year, just one electric car on the roads can save an average 1.5 million grams of CO_2. That's the equivalent of four return flights from London to Barcelona.

3. CODE:

```
 ## app.R ##
library(shinydashboard)
library(shiny)
library(plotly)
library(plyr)
library(dplyr)
library(ggplot2)
library(datasets)
library(tidyverse)
library(leaflet)

evs <-
read.csv('C:/Users/Administrator/Desktop/ElectricCarData_Clean.csv')
price <- evs[, c(14)]
acc <- evs[,c(3)]

Rapidcharge_price <- evs[, c(8, 14)]
Rapidcharge_price <-
Rapidcharge_price[order(Rapidcharge_price$PriceEuro), ]
avg_price <- aggregate(Rapidcharge_price$PriceEuro ~
Rapidcharge_price$RapidCharge, FUN = "mean")
avg_price <- avg_price %>% rename(RapidCharge =
`Rapidcharge_price$RapidCharge`, Avg_Price =
`Rapidcharge_price$PriceEuro`)
less_than_avg <- Rapidcharge_price %>% filter(PriceEuro <
avg_price$Avg_Price[2])
yes_percentage <- round((nrow(less_than_avg %>% filter(RapidCharge ==
"Yes")) /nrow(less_than_avg)) * 100, 2)
no_percentage <- round(100 - yes_percentage, 2)
```

```
percentages <- data.frame(rapidcharge = c("Yes", "No"), percentage =
rbind("1" = yes_percentage,"2" = no_percentage))
less_than_25 <- Rapidcharge_price %>% filter(PriceEuro < 25000)
yes_percentage1 <- round((nrow(less_than_25 %>% filter(RapidCharge ==
"Yes")) /nrow(less_than_25)) * 100, 2)
no_percentage1 <- round(100 - yes_percentage1, 2)
percentages_2 <- data.frame(rapidcharge = c("Yes", "No"), percentage =
rbind("1" = yes_percentage1,"2" = no_percentage1))

head(percentages_2)

colors <- c('rgb(5, 193, 230)', 'rgb(255, 48, 124)')

data <- read.csv("C:/Users/Administrator/Desktop/Dataset.csv")


data$Date <-
strptime(as.character(data$Date.yyyy.MM.dd.),format="%m/%d/%Y")
data$Date <- as.POSIXct(data$Date)

data$DateTime <-
strptime(as.character(data$DateTime),format="%m/%d/%Y %H:%M")
data$DateTime <- as.POSIXct(data$DateTime)

data$Day <- as.numeric(as.character(strftime(data$DateTime,format="%d")))
data$Hour <- as.numeric(as.character(strftime(data$DateTime,format="%H")))

data <- data %>% filter(BC6!=0)

ui <- dashboardPage(skin="purple",
                    dashboardHeader(title = "Review 3",

tags$li(class="dropdown",tags$a(href="https://github.com/ananyasingh13",i
con("github"),"Ananya's Github", target="_blank")),

tags$li(class="dropdown",tags$a(href="https://github.com/Shivamraj-
hub",icon("github"),"Shivam's Github", target="_blank"))),
                    dashboardSidebar(sidebarMenu(
                       menuItem("Dashboard", tabName = "dashboard", icon
= icon("dashboard")),
                       menuItem("Data 1", tabName = "data1", icon =
icon("database")),
                       menuItem("Interactivity", tabName =
"Interactivity", icon = icon("bar-chart")),
                       menuItem("Data 2", tabName = "data2", icon =
icon("database")),
                       menuItem("Aggregates", tabName = "aggregate", icon
= icon("box")),
```

```
                      menuItem("Charts",
                               icon = icon("line-chart"),
                               menuSubItem("graph1",tabName = "chart1",
icon = icon('line-chart')),
                               menuSubItem("graph2",tabName = "chart2",
icon = icon('line-chart')),
                               menuSubItem("graph2a",tabName = "chart2a",
icon = icon('line-chart')),

                               menuSubItem("graph3",tabName = "chart3",
icon = icon('line-chart')),
                               menuSubItem("graph4",tabName = "chart4",
icon = icon('line-chart')),
                               menuSubItem("graph5",tabName = "chart5",
icon = icon('line-chart'))),
                      menuItem("Link to code files", href =
"https://github.com/ananyasingh13/Data-visualization-Project", icon =
icon("code"))
                   )),
                   dashboardBody(

                      tabItems(

                         tabItem(tabName =
"Interactivity",selectInput(inputId="color1",label="Choose Color",choices
= c("Red"="Red","Blue"="Blue","Green"="Green"),

selected = "Blue",multiple = F),
                                 radioButtons(inputId = "border1",label
= "Select Border",choices = c("Black"="#000000","White"="#ffffff")),


selectInput(inputId="channel1",label="Choose Channel",choices =
c("BC1"="BC1",

"BC2"="BC2",

"BC3"="BC3",

"BC4"="BC4",

"BC5"="BC5",

"BC6"="BC6",

"BC7"="BC7"),
                                          selected = "BC6",multiple =
F),
```

```
                                  sliderInput(inputId = "bins1xz",
                                              label = "Number of bins:",
                                              min = 1,
                                              max = 50,
                                              value = 30),

                                  sliderInput(inputId = "range1",
                                              label = "Data Range",
                                              min = 1,
                                              max = 31,
                                              value = c(1,31)),
                                  fluidRow(box( width=12,
                                                plotOutput(outputId =
"distPlot"),
                                                plotOutput(outputId =
"distPlot1"),
                                                plotOutput(outputId =
"distPlot2")))),

                          tabItem(tabName =
"dashboard",fluidRow(box(title='Map of VIT Vellore
Campus',status="success",background = "green")),leafletOutput("map"),
                                  fluidRow(box(title='Dashboard Creation
using R ','Made by Shivam Raj and Ananya
Singh',status="success",background = "green"))),

                          tabItem(tabName =
"data2",dataTableOutput("mydatatable")),
                          tabItem(tabName =
"data1",dataTableOutput("datatable")),

                          tabItem(tabName = "aggregate",
                                  fluidRow(infoBoxOutput("max", width=6),
infoBoxOutput("min",width=6)),
                                  fluidRow(infoBoxOutput("max1", width=6),
infoBoxOutput("min1",width=6))),


                          tabItem(tabName =
"chart1",fluidRow(box(title='Relationship between Power Train and Range
of the EV',

plotlyOutput("one"), width=12, height=500,status = "primary")),

fluidRow(box(title="interpretation",status="success",width=10, background
= "black", "The above graph shows the relation between Range of the cars
in Km vs the Power train. This graph is an interactive scatterplot where
we can hover around our cursor to see a particular entry or we can zoom
```

```
into the plot or select a particular region and also take snapshots. It
gives us an idea of how the range of Different EMVs vary with model."))),

                          tabItem(tabName =
"chart2",fluidRow(box(title='Relation between price and rapid charging',

plotlyOutput("two"), width=12,status = "primary")),

fluidRow(box(title="interpretation",status="success",width=10, background
= "black","This plot shows the relation between Price of the car and
weather it has rapid charging or not. It also indicates the average for
both the choices. This interactive plot helps user choose wisely as there
is an arrow pointer which they can use to select particular price ranged
cars. User can also zoom in to the graph for better clarity of number of
options and also they can snip or capture the plot
 "))),



                          tabItem(tabName =
"chart3",fluidRow(box(title='Quickest 0-100 km/h in an EV',

plotlyOutput("three"), width=12, status = "primary")),

fluidRow(box(title="interpretation",status="success",width=10, background
= "black","The above interactive bar plot is mainly for speed enthusiasts
as people have this misconception of weather EVs can deliver the power
and speed that normal fuel vehicles do. So by this interactive graph
users can easily go for cars which have faster accelerations with all
interactive functions in it."))),

                          tabItem(tabName =
"chart4",fluidRow(box(title='Efficiencies of EVs',

plotlyOutput("four"), width=12, status = "primary")),

fluidRow(box(title="interpretation",status="success",width=10, background
= "black","The above interactive bar plot is mainly for people who look
for an economic drive as people have this misconception of whether EVs
can deliver the same kind of fuel economy fuelled cars do. So by this
interactive graph users can easily go for cars which have good economy
with all interactive functions in it.
 "))),

                          tabItem(tabName =
"chart5",fluidRow(box(title='Relationship between Power Train and Top
Speed of the EV',

plotlyOutput("five"), width=12, status = "primary")),
```

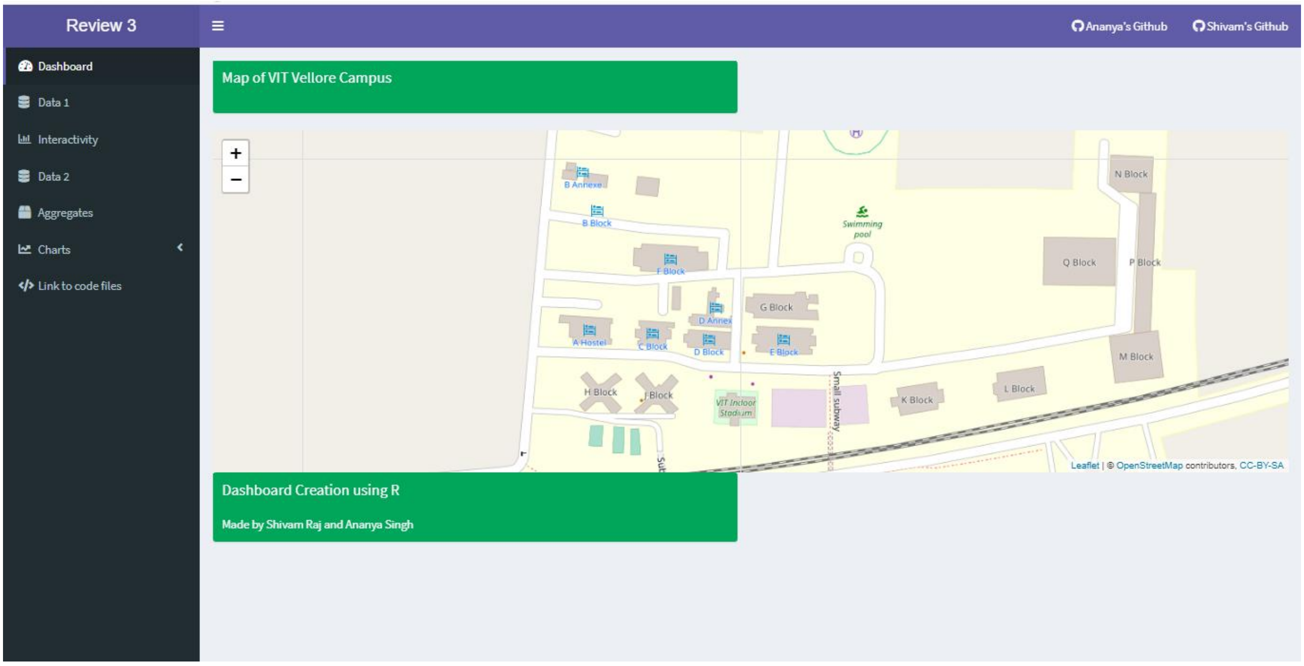
```
fluidRow(box(title="interpretation",status="success",width=10, background
= "black","The above plot shows the top speed of an EV in accordance to
its Power Train Technology. We can see that AWD type have the max top
speeds. Thus they are better choices. User can hover their cursor to the
spots to get the clear idea of which power train will will go upto
highest speeds.
 ")))



                       )
                    )
)

server <- function(input, output) {
   set.seed(122)
   output$disPlot <- renderPlot({plot(rnorm(input$n), rnorm(input$n))})
   output$map<- renderLeaflet({leaflet() %>% addTiles() %>%
setView(79.1594 , 12.9717, zoom=16)})
   output$one <- renderPlotly({plot_ly(data=evs, x=~Range_Km, y=
~PowerTrain, type="scatter",color=~PowerTrain, mode="markers")})
   output$two <- renderPlotly({plot_ly(data=evs, x= ~PriceEuro, y=
~RapidCharge, color=~RapidCharge ,type="scatter")})



   output$three <- renderPlotly({plot_ly(evs, x = ~Model, y = ~AccelSec,
type ='bar')})
   output$four <- renderPlotly({plot_ly(evs,  x = ~Model, y =
~Efficiency_WhKm, type ='bar')})
   output$five <- renderPlotly({plot_ly(evs, x = ~TopSpeed_KmH, y =
~PowerTrain, type="scatter",color=~PowerTrain, mode="markers")})

   output$min <- renderInfoBox({infoBox(title = "Minimum
Price",value=min(price),subtitle = "Minimum Price of EMV in Euro",
fill=TRUE)})
   output$max <- renderInfoBox({infoBox(title = "Maximum
Price",value=max(price),subtitle = "Maximum Price of EMV in Euro",
fill=TRUE)})
   output$min1 <- renderInfoBox({infoBox(title = "Minimum
Acc/sec",value=min(acc),subtitle = "Minimum acceleration/sec from all EMV
in m/s", fill=TRUE, color="yellow")})
   output$max1 <- renderInfoBox({infoBox(title = "Maximum
Acc/sec",value=max(acc),subtitle = "Maximum acceleration/sec from all EMV
in m/s", fill=TRUE, color="yellow")})

   output$mydatatable <- renderDataTable({evs})

   output$datatable <- renderDataTable({data})
```

```
   output$distPlot <- renderPlot({

      if(input$color1=="Red"){
         sColor = "#ff3300"
      }else if(input$color1=="Blue"){
         sColor = "#3399ff"
      }else if(input$color1=="Green"){
         sColor = "#66ff33"
      }

      p2 <- data %>%  filter(Day >= input$range1[1] & Day <=
input$range1[2]) %>% ggplot()
      if(input$channel1 == "BC1"){
         p2 <- p2 + geom_histogram(aes(x=BC1),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC2"){
         p2 <- p2 + geom_histogram(aes(x=BC2),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC3"){
         p2 <- p2 + geom_histogram(aes(x=BC3),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC4"){
         p2 <- p2 + geom_histogram(aes(x=BC4),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC5"){
         p2 <- p2 + geom_histogram(aes(x=BC5),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC6"){
         p2 <- p2 + geom_histogram(aes(x=BC6),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }else if(input$channel1 == "BC7"){
         p2 <- p2 + geom_histogram(aes(x=BC7),bins =
input$bins1xz,col=input$border1,fill=sColor)
      }
      p2 <- p2 +  theme_bw()+
         theme(axis.title =
element_text(size=12,color="BLACK",face="bold"),
               axis.text =
element_text(size=14,color="BLACK",face="bold"))+
         labs(x="Black Carbon (ng/m3)",y="Count",title=paste(" Histogram
showing Black Carbon Concentration",input$channel1,sep = " "))

      p2
      #hist(x, breaks = bins, col = sColor, border = input$border1,
      #     xlab = "Waiting time to next eruption (in mins)",
      #     main = "Histogram of waiting times")
   })

   output$distPlot1 <- renderPlot({
```

```
      p1 <- data  %>%  filter(Day >= input$range1[1] & Day <=
input$range1[2]) %>% ggplot(aes(x=DateTime))
      if(input$channel1 == "BC1"){
         p1 <- p1 + geom_line(aes(y=BC1,col="BC1"),size=0.5)
      }else
         if(input$channel1 == "BC2"){
            p1 <- p1 + geom_line(aes(y=BC2,col="BC2"),size=0.5)
         }else
            if(input$channel1 == "BC3"){
               p1 <- p1 + geom_line(aes(y=BC3,col="BC3"),size=0.5)
            }else
               if(input$channel1 == "BC4"){
                  p1 <- p1 + geom_line(aes(y=BC4,col="BC4"),size=0.5)
               }else
                  if(input$channel1 == "BC5"){
                     p1 <- p1 + geom_line(aes(y=BC5,col="BC5"),size=0.5)
                  }else
                     if(input$channel1 == "BC6"){
                        p1 <- p1 +
geom_line(aes(y=BC6,col="BC6"),size=0.5)
                     }else
                        if(input$channel1 == "BC7"){
                           p1 <- p1 +
geom_line(aes(y=BC7,col="BC7"),size=0.5)
                        }
      p1 <- p1 +  theme_bw()+
         theme(axis.title =
element_text(size=12,color="BLACK",face="bold"),
               axis.text =
element_text(size=14,color="BLACK",face="bold"))+
         labs(x="Time",y="Black Carbon (ng/m3)",title="Variation in Black
Carbon Concentration in Air for year 2017",colour="Channel")

      p1

   })

   output$distPlot2 <- renderPlot({
      d <- data  %>%  filter(Day >= input$range1[1] & Day <=
input$range1[2])

      d <- ddply(d, .variables = c("Hour"),function(x){

         BC1avg <- mean(x$BC1,na.rm = T)
         BC2avg <- mean(x$BC2,na.rm = T)
         BC3avg <- mean(x$BC3,na.rm = T)
         BC4avg <- mean(x$BC4,na.rm = T)
         BC5avg <- mean(x$BC5,na.rm = T)
```

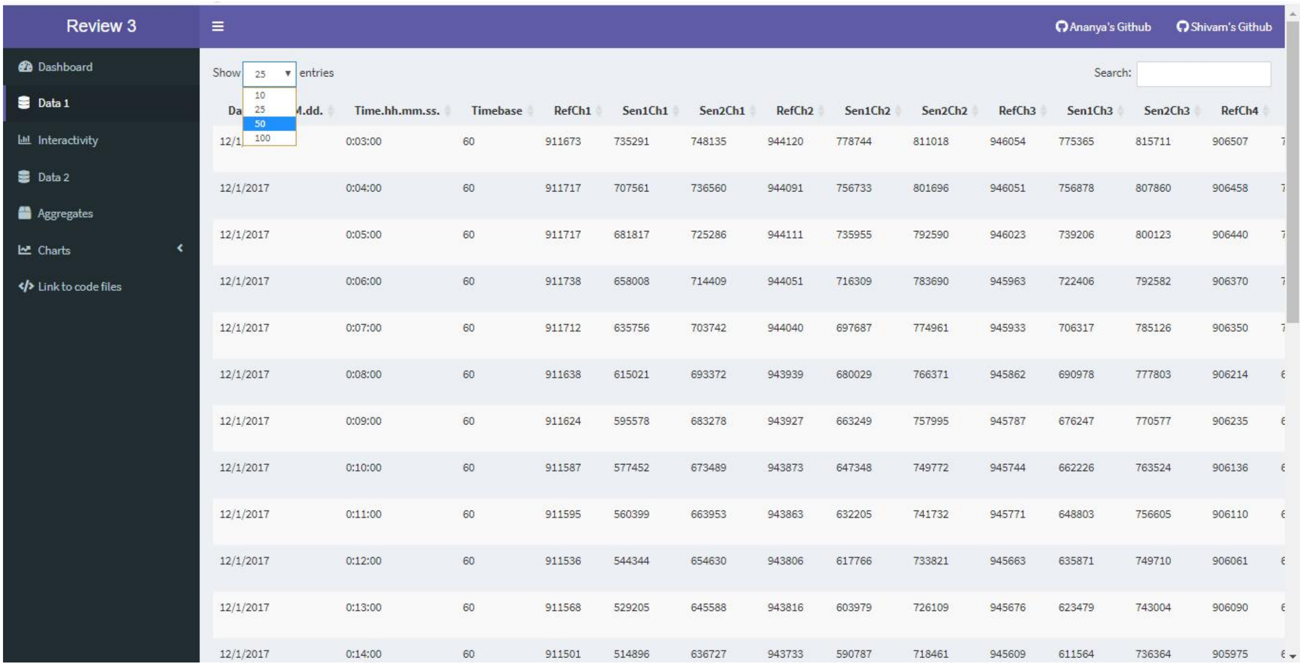
```
         BC6avg <- mean(x$BC6,na.rm = T)
         BC7avg <- mean(x$BC7,na.rm = T)

         data.frame(BC1avg,BC2avg,BC3avg,BC4avg,BC5avg,BC6avg,BC7avg)
      })

      p1 <- d %>% ggplot(aes(x=Hour))
      if(input$channel1 == "BC1"){
         p1 <- p1 + geom_line(aes(y=BC1avg,col="BC1"),size=1)
         p1 <- p1 + geom_point(aes(y=BC1avg))
      }else if(input$channel1 == "BC2"){
         p1 <- p1 + geom_line(aes(y=BC2avg,col="BC2"),size=1)
         p1 <- p1 + geom_point(aes(y=BC2avg))
      }else if(input$channel1 == "BC3"){
         p1 <- p1 + geom_line(aes(y=BC3avg,col="BC3"),size=1)
         p1 <- p1 + geom_point(aes(y=BC3avg))
      }else if(input$channel1 == "BC4"){
         p1 <- p1 + geom_line(aes(y=BC4avg,col="BC4"),size=1)
         p1 <- p1 + geom_point(aes(y=BC4avg))
      }else if(input$channel1 == "BC5"){
         p1 <- p1 + geom_line(aes(y=BC5avg,col="BC5"),size=1)
         p1 <- p1 + geom_point(aes(y=BC5avg))
      }else if(input$channel1 == "BC6"){
         p1 <- p1 + geom_line(aes(y=BC6avg,col="BC6"),size=1)
         p1 <- p1 + geom_point(aes(y=BC6avg))
      }else if(input$channel1 == "BC7"){
         p1 <- p1 + geom_line(aes(y=BC7avg,col="BC7"),size=1)
         p1 <- p1 + geom_point(aes(y=BC7avg))
      }
      p1 <- p1 +  theme_bw()+
         theme(axis.title =
element_text(size=12,color="BLACK",face="bold"),
               axis.text =
element_text(size=14,color="BLACK",face="bold"))+
         labs(x="Time",y="Black Carbon (ng/m3)",title="Mean of Black
Carbon Concentration in Air - Average Diurnal Variation-
2017",colour="Channel")

      p1

   })


}

shinyApp(ui, server)
```

4. Screenshots from our dashboard along with explanation:

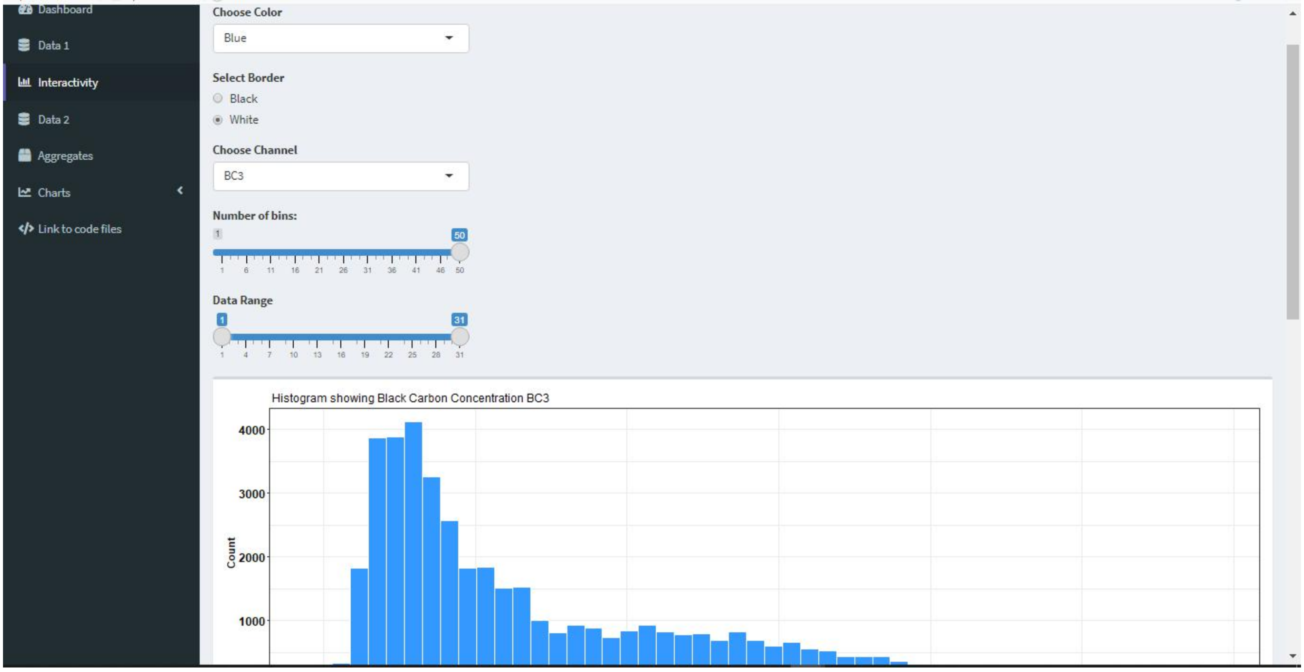


This is the page we land to when we first launch our dashboard. It has a live map of Vellore and we can zoom in and out of the live map.

The three lines on the top pane slide the sidebar in and out of screen.



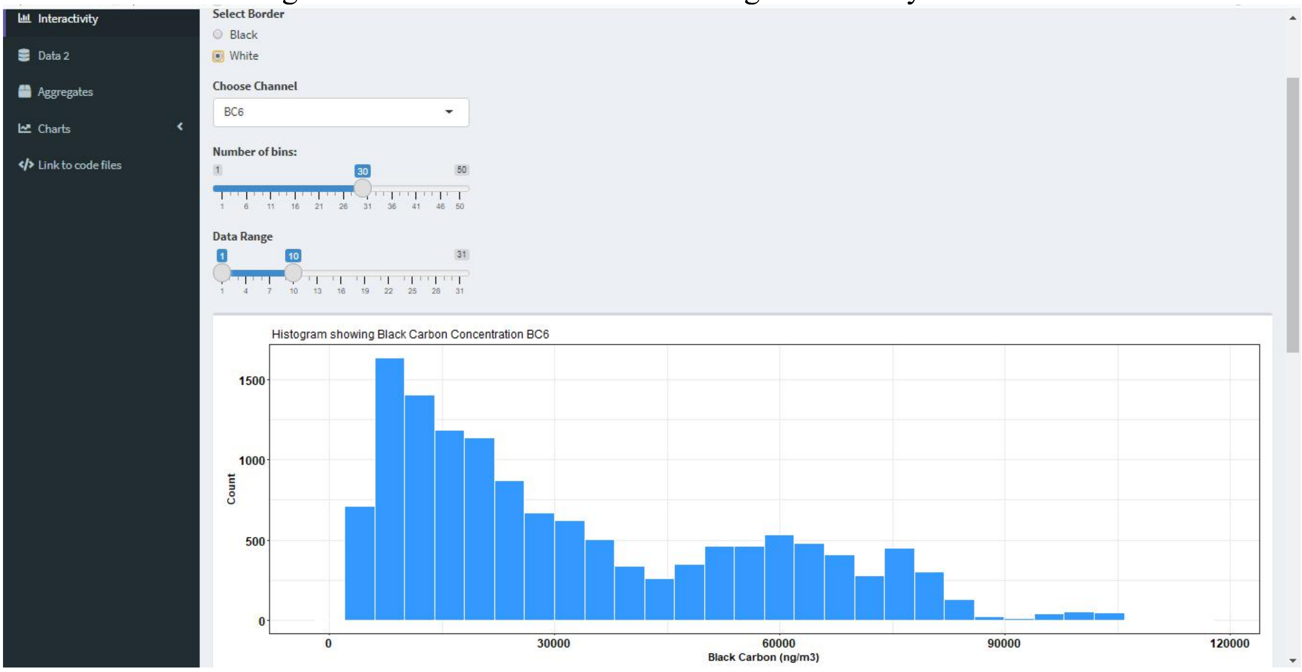
This is the second page of our Dashboard, it contains Annual data for black carbon concentration in air of every minute for an entire year. The number of entries can be adjusted accordingly. Also any particular value can be searched either globally from top-right search bar or column wise through the search bar at the end of every column. We have used this dataset to emphasize on the rising levels of black carbon concentration in air, thereby concluding the urgent need to use electromotives on a larger scale.

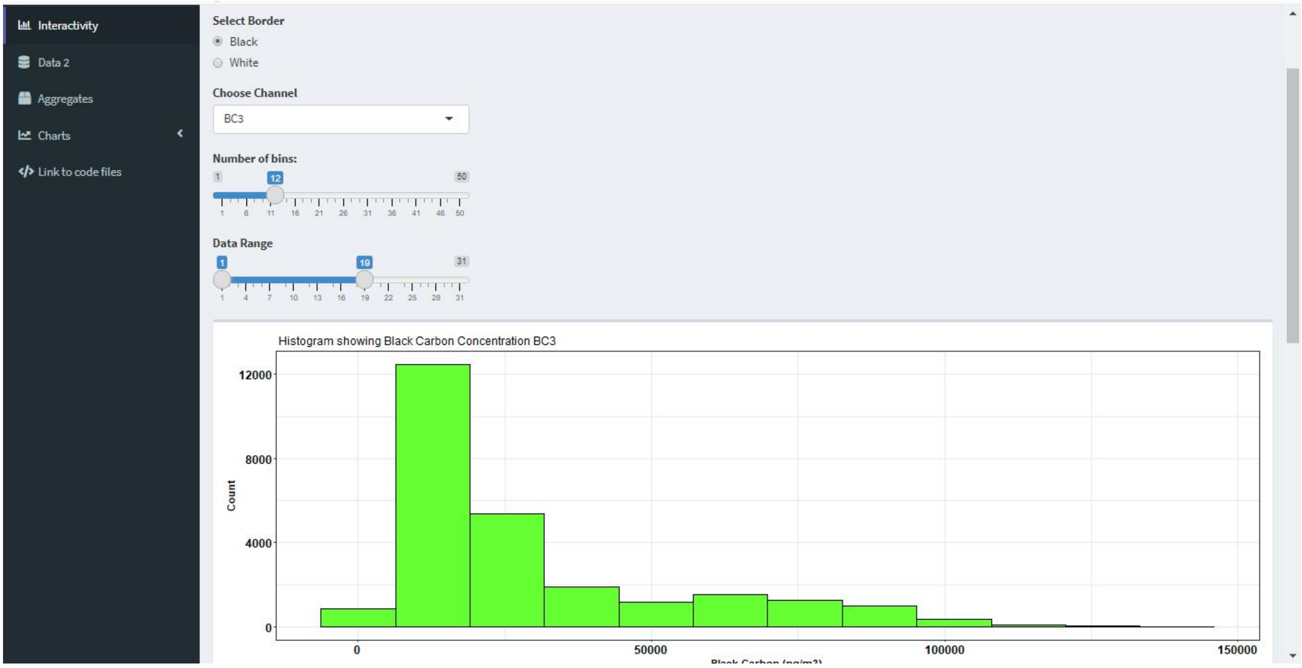


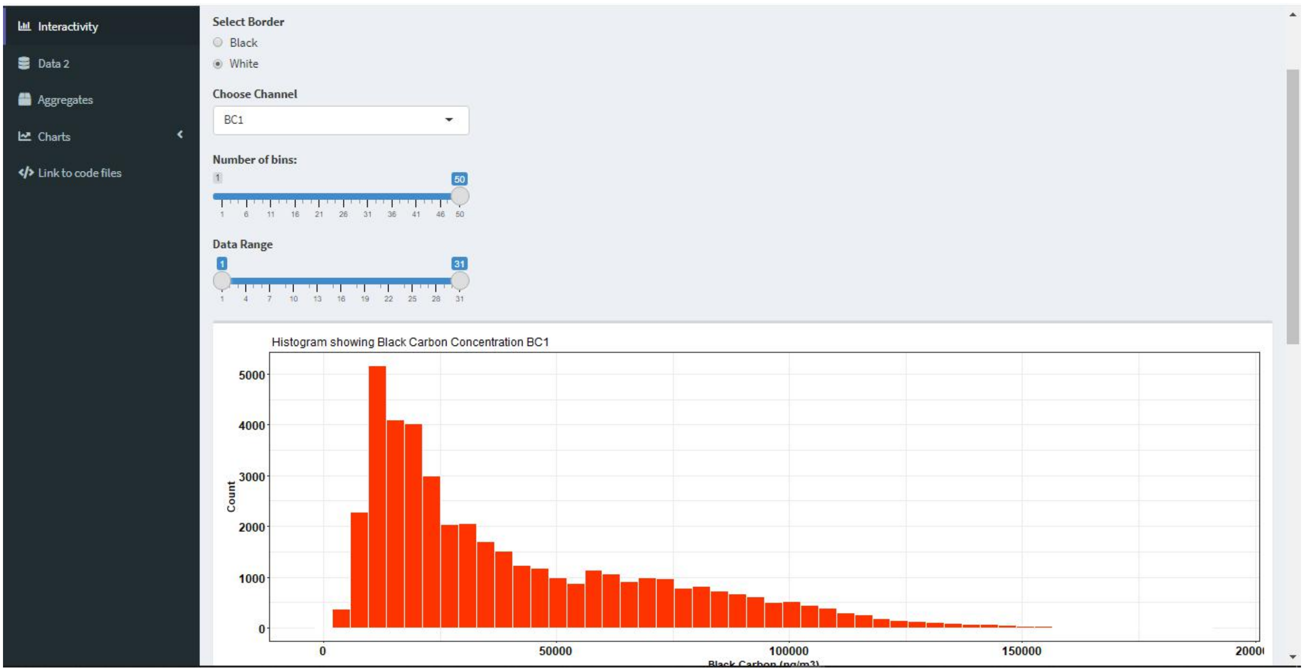
This is the interactivity pane, here we have visualized the carbon concentration dataset, showing the dangerous lavels of black carbon in air, the variation of black carbon concentration throughout the day for a month and the average dirunia variation of black carbon concentration . This gives us a fair idea of at what time of day the concentration is at its peak and low, on which days the emmisions are high and on which day they are low and the trends of its variation.

In the above screenshot, we can choose different colours for histogram and its border, choose a channel for black carbon emmision, and vary the number of bins and data range.

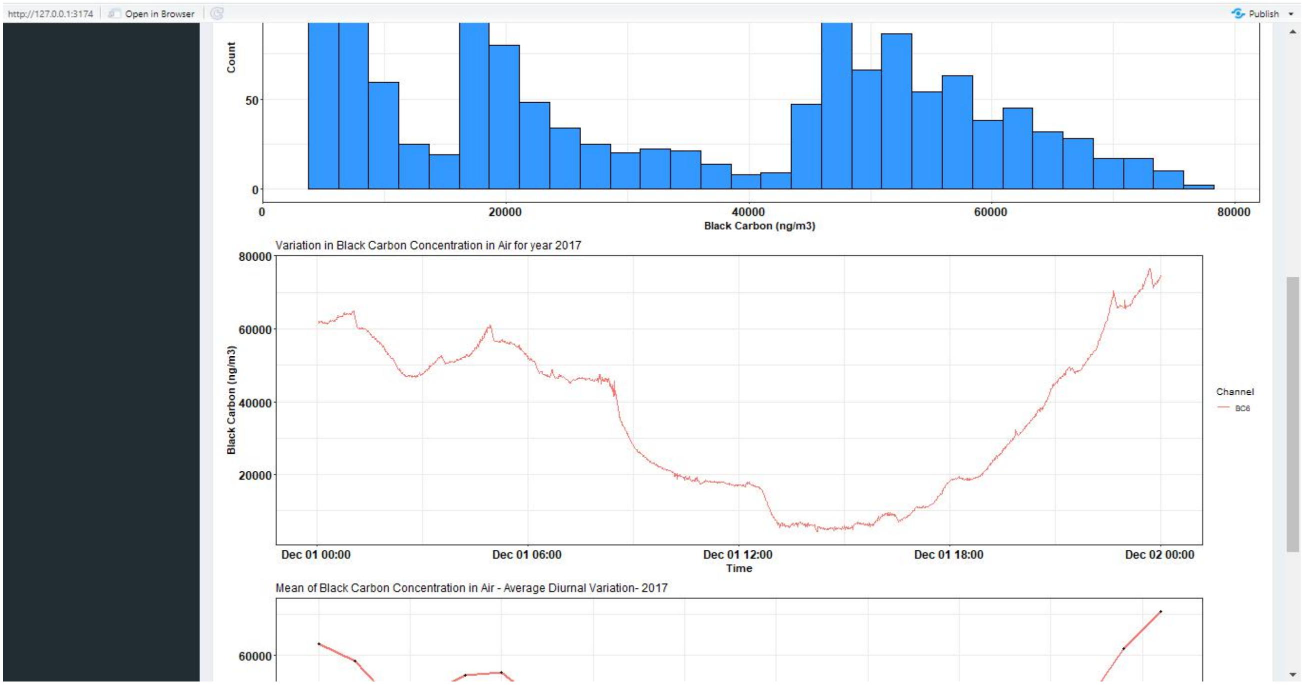
Below are a few histograms at different variations showing interactivity:



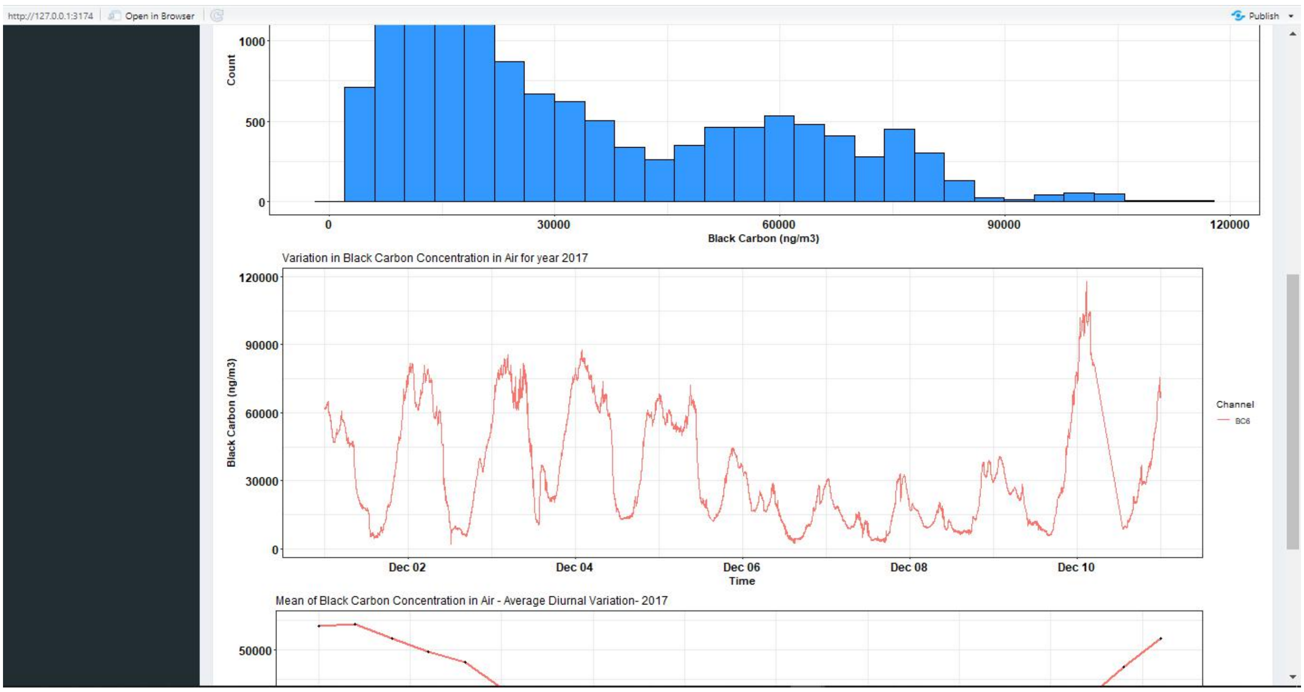

Interactivity
Data 2
Aggregates
Charts
Link to code files
Select Border
Black
White
Choose Channel
BC3
Number of bins:
Data Range
Histogram showing Black Carbon Concentration BC3
Count
12000
8000
4000
0
0
50000
100000
150000


Interactivity
Data 2
Aggregates
Charts
Link to code files
Select Border
Black
White
Choose Channel
BC1
Number of bins:
Data Range
Histogram showing Black Carbon Concentration BC1
Count
5000
4000
3000
2000
1000
0
0
50000
100000
150000

Similarly by varying the datarange we see the trends of black carbon emmision on hourly basis and its aggregate over a day:

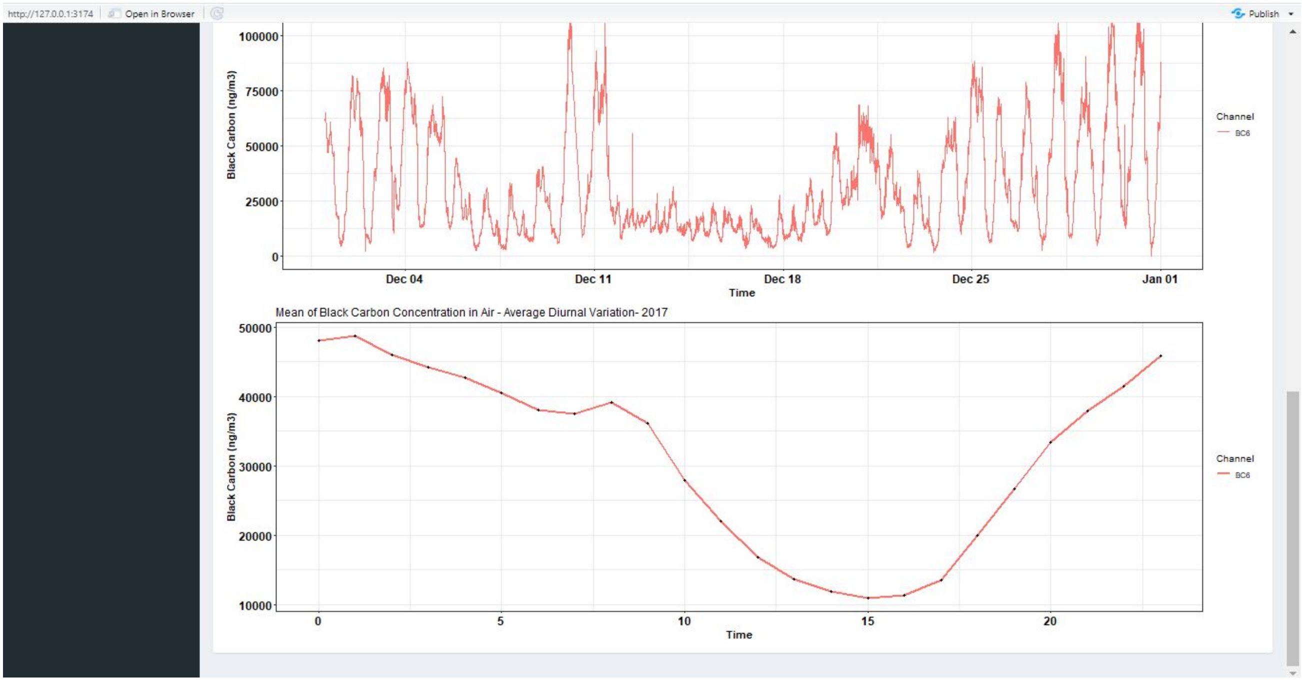


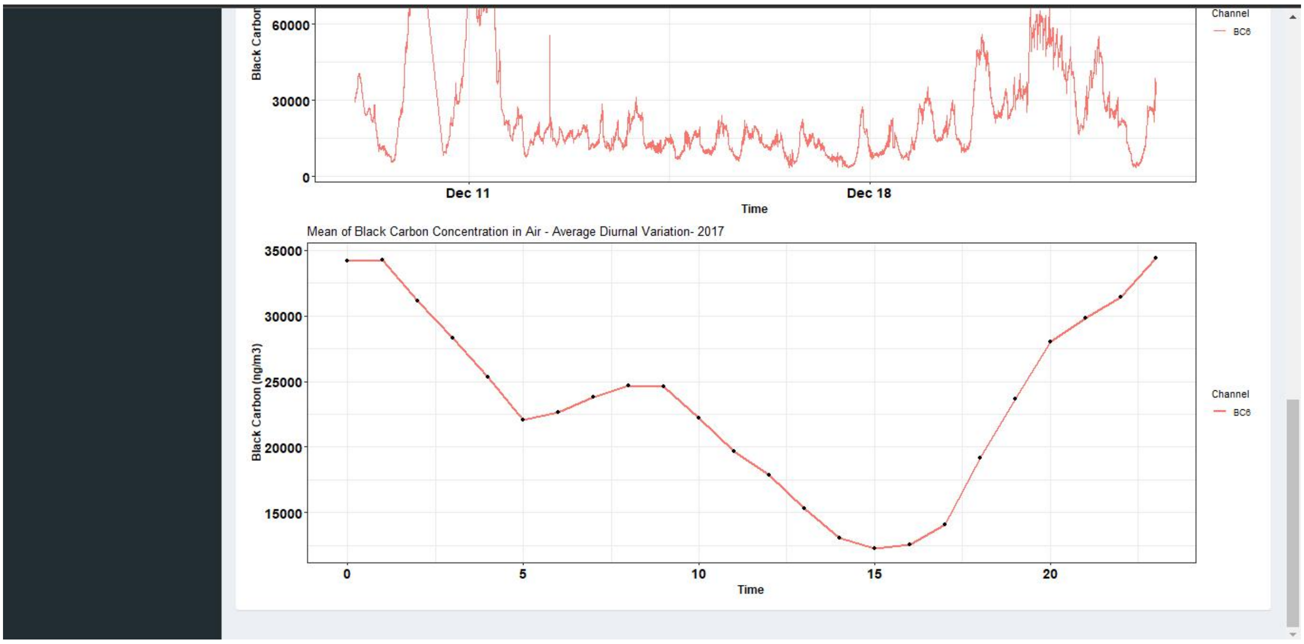
We see that the carbon emmisions are very high early in the morning and late in the night, peak time for any normal working person to be out on the streets, thus susceptible serious health hazards .



Similarly we see that the trend remains same even for a span of ten days.

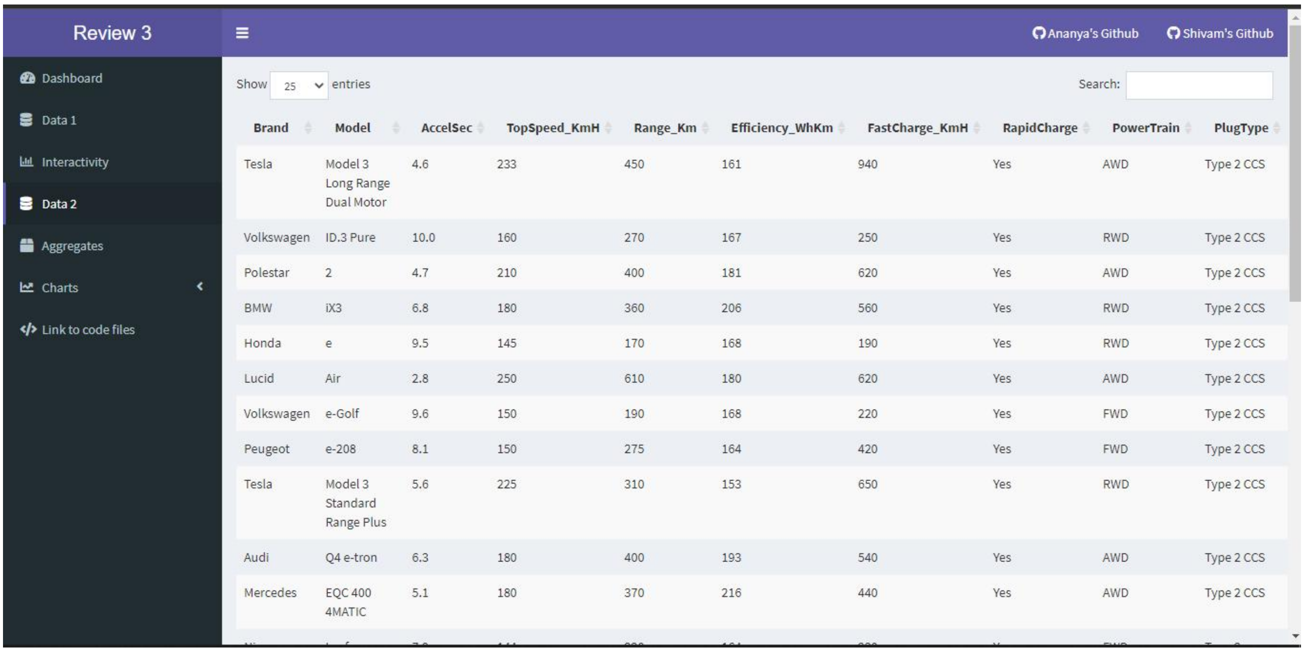
Lastly we observe the trend of average black carbon emission in air over a month for a year.





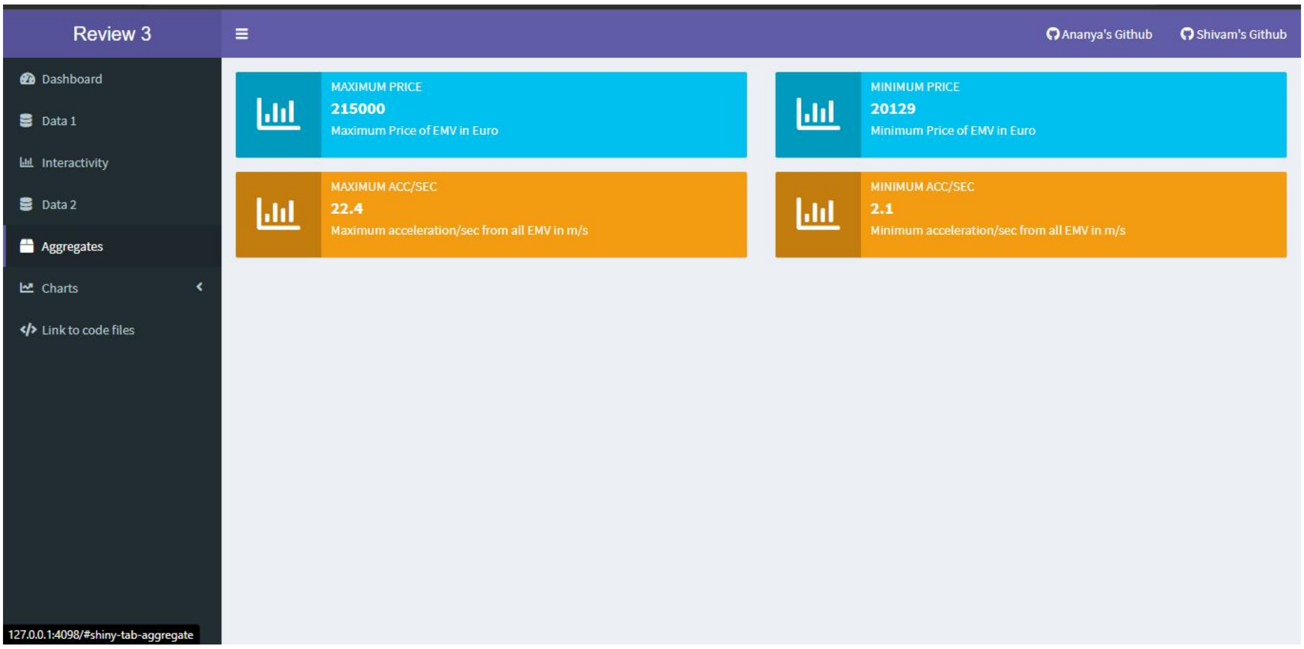
We also notice the trends in period of month when the corbon level is very high, we see that during month beginnings and month ends the carbon leverls are vey high.

The fourth menu of our dash board consists of our main data set that is Electric Vehicle Analysis Dataset.



Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType
Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	Yes	AWD	Type 2 CCS
Volkswagen	ID.3 Pure	10.0	160	270	167	250	Yes	RWD	Type 2 CCS
Polestar	2	4.7	210	400	181	620	Yes	AWD	Type 2 CCS
BMW	iX3	6.8	180	360	206	560	Yes	RWD	Type 2 CCS
Honda	e	9.5	145	170	168	190	Yes	RWD	Type 2 CCS
Lucid	Air	2.8	250	610	180	620	Yes	AWD	Type 2 CCS
Volkswagen	e-Golf	9.6	150	190	168	220	Yes	FWD	Type 2 CCS
Peugeot	e-208	8.1	150	275	164	420	Yes	FWD	Type 2 CCS
Tesla	Model 3 Standard Range Plus	5.6	225	310	153	650	Yes	RWD	Type 2 CCS
Audi	Q4 e-tron	6.3	180	400	193	540	Yes	AWD	Type 2 CCS
Mercedes	EQC 400 4MATIC	5.1	180	370	216	440	Yes	AWD	Type 2 CCS

Same as before The number of entries can be adjusted accordingly. Also any particular value can be searched either globally from top-right search bar or column wise through the search bar at the end of every column. From this Dataset we will find out the various pros and cons of every model of EV present. We will compare various attributes that are essential while choosing the most efficient and economical EV.

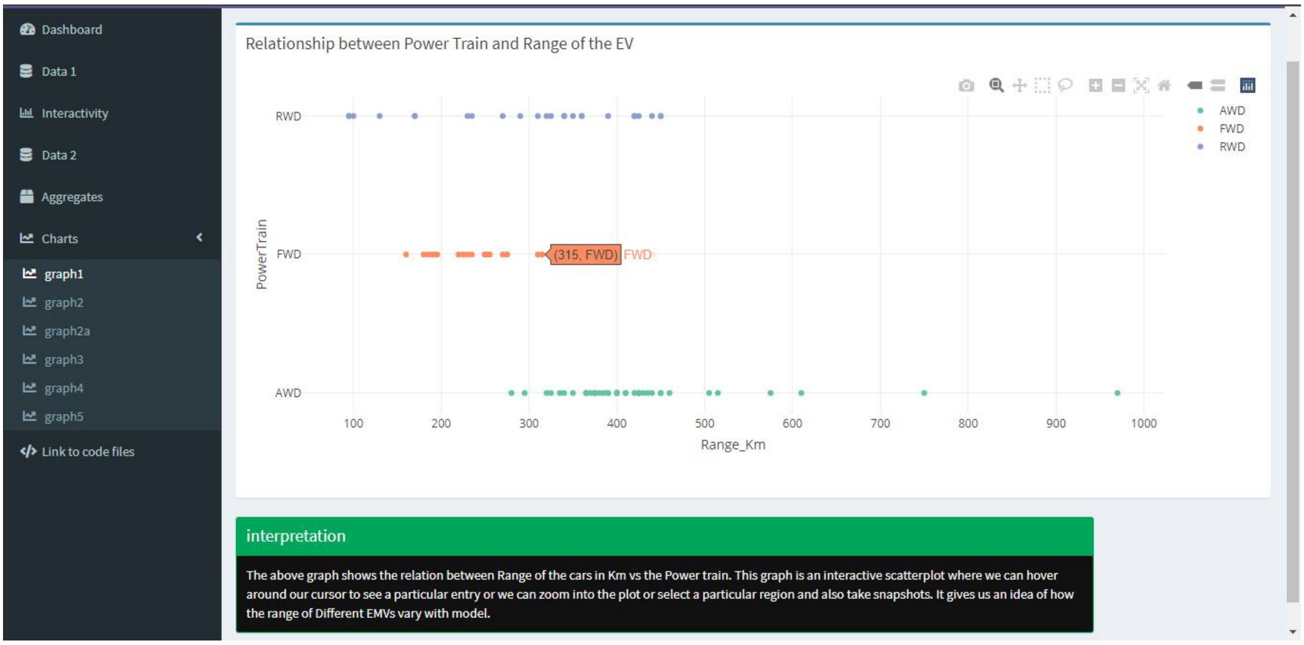


In the next menu we have listed some aggregates like minimum price , maximum price , minimum acc/sec, max acc/sec to give a basic idea of range of cost and acceleration of EVs.

In this next menu we have compiled the various interactive graphs used in previous reviews along with their interpretations. The graphs are made using plotly and are highly interactive.

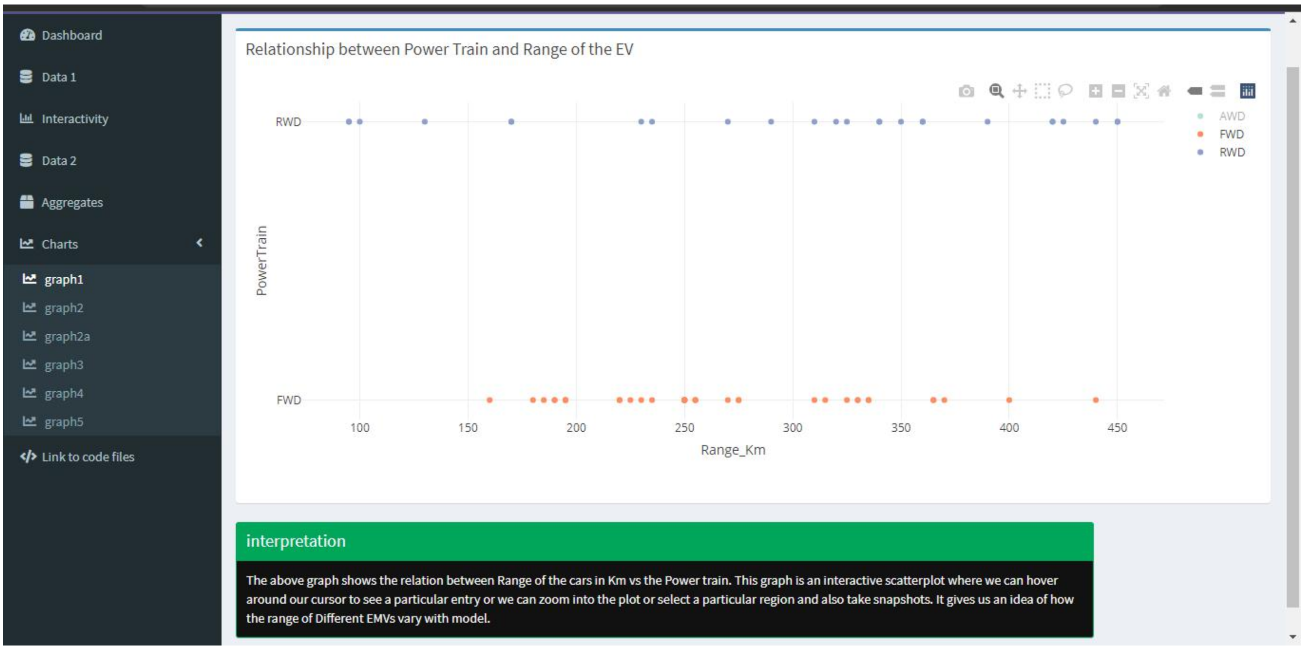
The above graph shows the relation between Range of the cars in Km vs the Power train. This graph is an interactive scatterplot where we can hover around our cursor to see a particular entry or we can zoom into the plot or select a particular region and also take snapshots. It gives us an idea of how the range of Different EMVs vary with model.

We can view the individual data by hovering over the point.



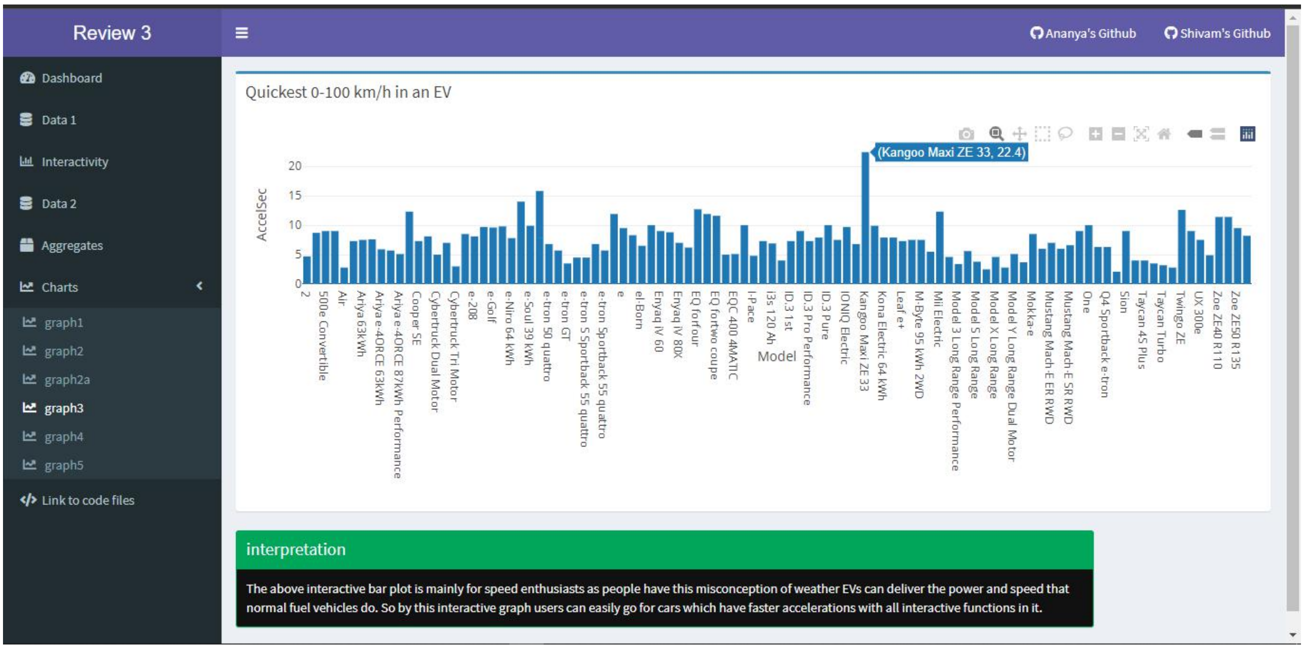
We can magnify and see the trends of specific parts:

We can choose the datas for which we want to see the visualization by enabling and diabling then through legends:

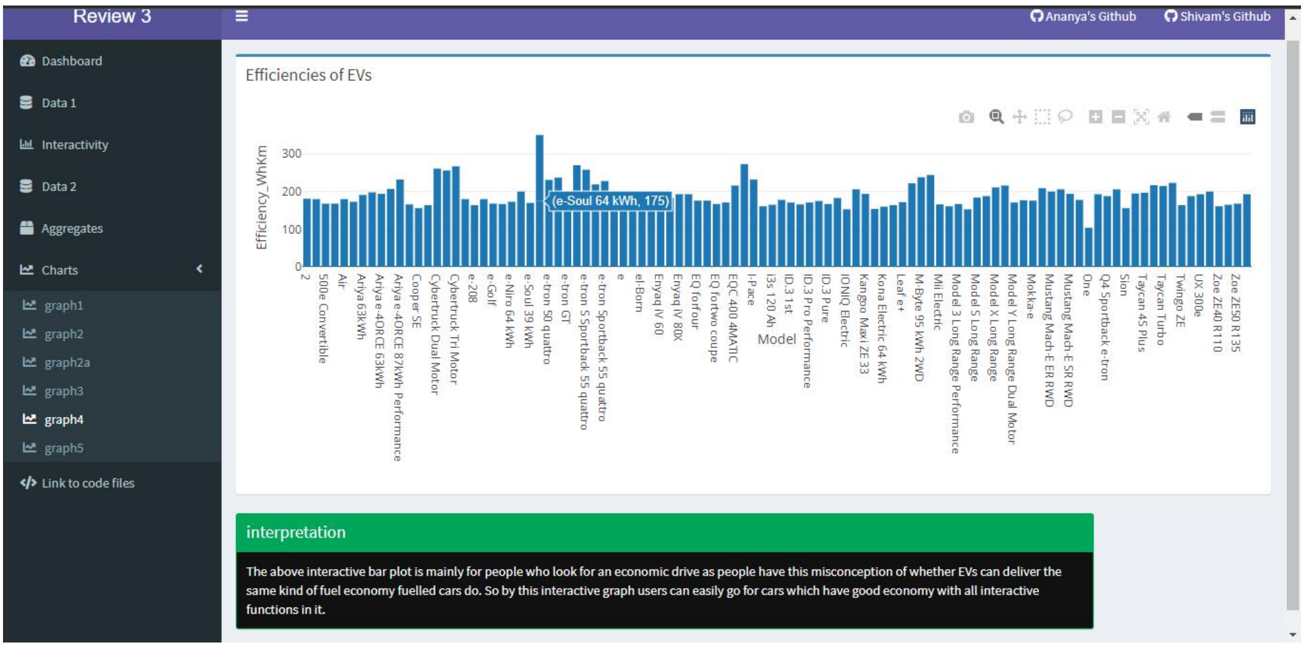


These are the various graphs in our dashboard along with their interpretation.

This plot shows the relation between Price of the car and weather it has rapid charging or not. It also indicates the average for both the choices. This interactive plot helps user choose wisely as there is an arrow pointer which they can use to select particular price ranged cars. User can also zoom in to the graph for better clarity of number of options and also they can snip or capture the plot



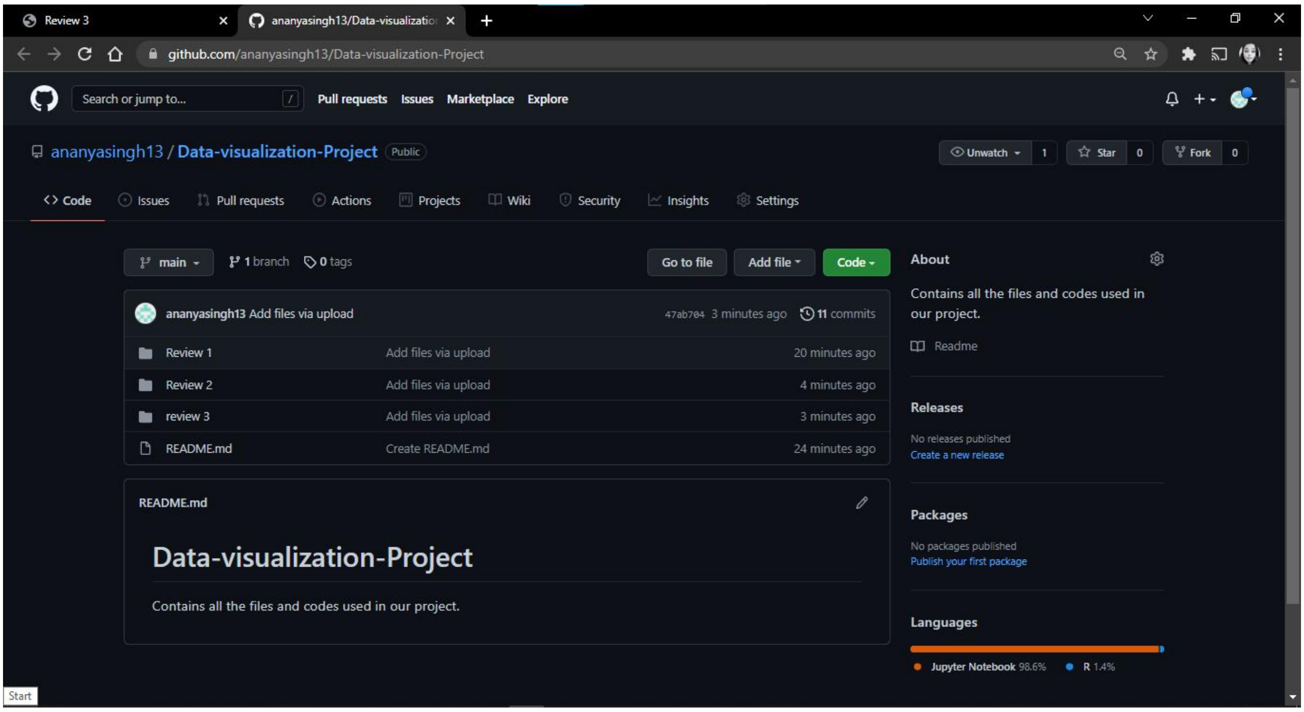
The above interactive bar plot is mainly for speed enthusiasts as people have this misconception of weather EVs can deliver the power and speed that normal fuel vehicles do. So by this interactive graph users can easily go for cars which have faster accelerations with all interactive functions in it.

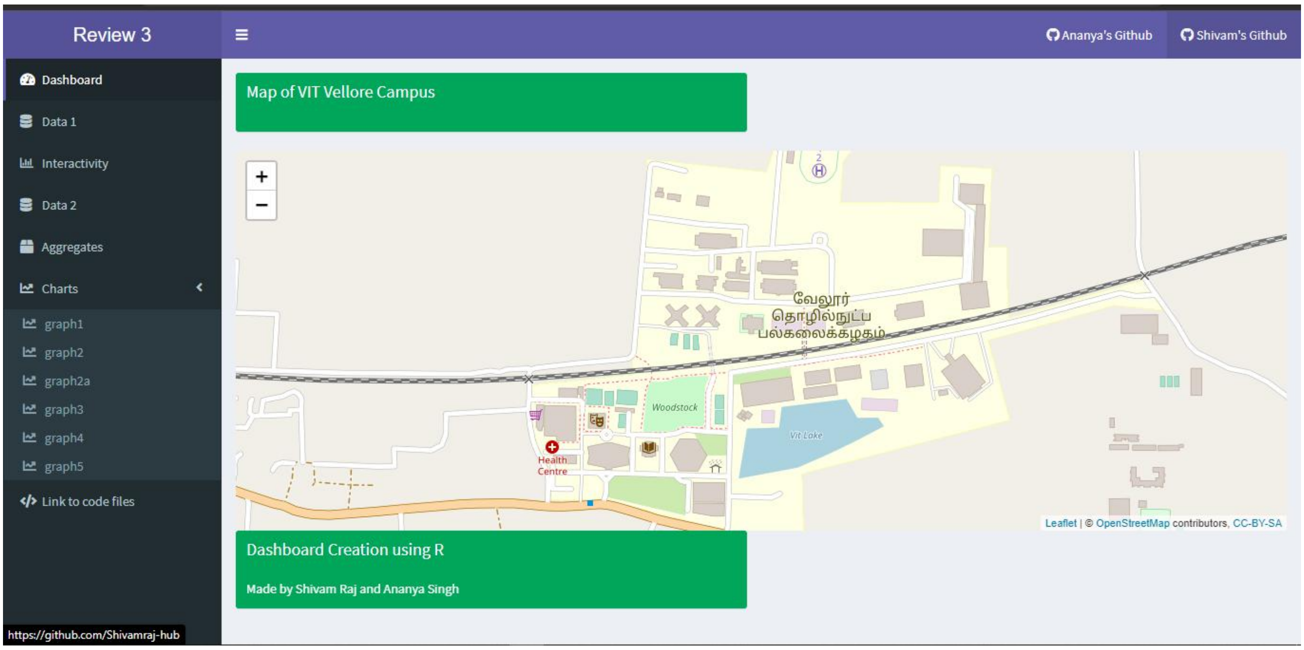


The above interactive bar plot is mainly for people who look for an economic drive as people have this misconception of whether EVs can deliver the same kind of fuel economy fuelled cars do. So by this interactive graph users can easily go for cars which have good economy with all interactive functions in it.

The above plot shows the top speed of an EV in accordance to its Power Train Technology. We can see that AWD type have the max top speeds. Thus they are better choices. User can hover their cursor to the spots to get the clear idea of which power train will will go upto highest speeds.

The last menu option takes the user to the github link where every code, dataset and files related to our project are updated.





The two menu options in top-right take the user to the respective githubs of the teammates.