AMOD 5250-Term Project

Arham Pawle

2023-04-12

Introduction

Formula 1 as an auto sport is growing very rapidly in North America as well as in Asia[1]due to the very popular Netflix series "Drive to survive" which is packed with thrill and stories. In the context of Formula One, F1 refers to the highest level of international single-seater auto racing. The F1 World Championship is governed by the Fédération Internationale de l'Automobile (FIA) and is known for its high-speed races, advanced technology, and skilled drivers. According to Tanaka et al[2], the teams that will be competing are called "Constructors" to determine which driver is the fastest to win the driver's championship since 1950 and which constructor is fastest to win the constructor championship since 1958. The main role of constructor is to manufacture the car, engine within a set of rules to make the fastest car.

Fields et al highlight the crucial role of aerodynamics in Formula 1 racecars, which allow for improved lap times, speed, stability, and handling. F1 regulations have evolved to limit some aerodynamic technologies for fair play and competitive racing.[3]

Alten et al discuss the demands on Formula One engines and strategies used to develop them. F1 engines must be lightweight, powerful, and reliable while meeting strict regulations on fuel efficiency and emissions. The paper covers the evolution of F1 engines, development strategies, and challenges faced by engine suppliers.[4]

Tires are a crucial component of Formula One cars, responsible for movement. Pirelli's website highlights the three main types of F1 tires: slicks for dry conditions, intermediates for light rain, and wets for heavy rain. Teams must manage tire selection carefully, balancing the need for speed with tire preservation. Effective tire management often determines F1 race outcomes.[5] Aerodynamics, engines, and tires contribute to the fastest lap times in Formula One. Modoni et al note that telemetry is a critical component of F1 racing, involving real-time data transmission from the car to the team's engineers during a race. This data includes speed, engine RPM, tire temperature, fuel consumption, and other performance indicators.[6]

The data collected through telemetry is used by the team's engineers to monitor the car's performance and make adjustments in real-time to optimize performance. Telemetry also allows engineers to diagnose potential issues with the car and make informed decisions about when to make pit stops or other strategic moves during the race.In addition to its use during races, telemetry data is also used by F1 teams to improve car performance during testing and development. An app displaying real-time laptimes and telemetry data is crucial in motorsports. Existing apps often lack necessary features or have limitations, such as a limited range of data, delayed updates, confusing interfaces, or limited functionality, making them less useful for drivers and teams.

An app that can display laptimes and telemetry data in real-time would be incredibly useful for drivers and teams. It would allow them to make informed decisions about car performance, identify issues early on, and make necessary adjustments in real-time. Additionally, an app could provide a more user-friendly and customizable interface, allowing teams to view only the data that is most important to them. The 2019 Brazilian Grand Prix saw Max Verstappen's team lose telemetry data, causing difficulties in monitoring the car's performance and leading to his retirement. Similarly, a timing system issue during the 2020 Styrian Grand Prix caused confusion, making it difficult for teams to assess their cars' performance and plan race strategy accurately.

Using these limitations, I decided to create a **R shiny app** which displays both the laptimes as well as the telemetries of individual laps of the 2 drivers Max Verstappen and Lewis Hamilton which can be selected through a drop down. This app would be a prototype but **not limited** to the above mentioned functions. The reason to choose R as a language were the benefits of using R language for visualisation analysis. R is a popular programming language and open-source software environment used for statistical computing and graphics.[7] R language is ideal for creating high-quality visualizations and graphics due to its flexibility, ease of use, and availability of numerous packages. It allows easy manipulation of data and is widely used for exploratory data analysis and statistical modeling. R language also provides extensive documentation, community support, and a wide range of resources, making it highly customizable to meet specific analysis requirements. Overall, R language is an ideal choice for data analysis and visualization in fields such as statistics and data science.[8]

Advantages of using R Shiny for app development, as highlighted by Kasprza et al, include its ease of use, seamless integration with R programming language, rapid prototyping, collaboration features, interactive visualizations, and deployment flexibility. These benefits make R Shiny a suitable choice for web application development in research settings, especially for those who need to build interactive data analysis and visualization tools without extensive knowledge of web development technologies.[9]

Code

```
library(shiny)
library(ggplot2)
library(dplyr)
library(gridExtra)
```

The packages shiny, ggplot2, dplyr, and gridExtra are commonly used in data analysis and visualization tasks in R. • shiny is a web application framework that allows developers to build interactive web applications using R. It provides an easy-to-use interface that can be used to create data-driven applications without requiring extensive knowledge of web development technologies. • ggplot2 is a popular data visualization package in R that provides a flexible and powerful framework for creating a wide range of visualizations, including bar charts, line charts, scatter plots, and more.

• dplyr is a data manipulation package in R that provides a set of functions for performing common data manipulation tasks, such as filtering, sorting, grouping, and summarizing data. It is often used in conjunction with ggplot2 for data wrangling tasks. • gridExtra is a package in R that provides a set of functions for arranging and combining multiple plots into a single output. It is often used in conjunction with ggplot2 to arrange multiple plots into a grid layout for improved visualization. Together, these packages provide a powerful set of tools for data analysis and visualization in R, allowing users to perform complex data analysis tasks and create high-quality visualizations with relative ease.

```
# Load telemetry data
lap_times <- read.csv("lap_times.csv")

# create an empty list to store the data
ham_ad_2021 <- list()
ver_ad_2021 <- list()

# loop through the lap numbers and load the data
for (i in 1:58) {
  ham_ad_2021[[i]] <- read.csv(paste0("ham_data_lap_2021_ad_", i, ".csv"))
  ver_ad_2021[[i]] <- read.csv(paste0("ver_data_lap_2021_ad_", i, ".csv"))
}
```

In the above code chunk, we read the “lap_times” CSV to read the data to plot the laptimes of the drivers(Max Verstappen and Lewis Hamilton). Then we create a list data type of both the drivers so that we can read the CSV’s for all 58 laps easily with help of a for loop.

```
# Load lap time data
lap_subset <- subset(lap_times, raceId == 1073 & (driverId == 1 | driverId == 830))
# Define UI
ui <- fluidPage(

  # App title
  titlePanel("Abu Dhabi 2021 Lap Times & Telemetry"),

  # Sidebar with dropdown to select lap
  sidebarLayout(
    sidebarPanel(
      selectInput("lap_select", "Select Lap:", choices = unique(lap_subset$lap)),
      br(),
      tags$em("Note: Lap time data is only available for laps 2-58.")
    ),

    # Display lap times and telemetry data for selected lap
    mainPanel(
      tabsetPanel(
        tabPanel("Lap Times",
                 plotOutput("lap_times_plot"),
                 tags$em("This plot is a comaprison for laptimes
                 between Lewis Hamilton(in blue) and Max Verstappen(in red)
                 for Abu Dhabi Grand Prix 2021.")
        ),
        tabPanel("Telemetry Data",
                 plotOutput("telemetry_table"),
                 tags$em("This plot is a comaprison of various telemetries
                 between Lewis Hamilton(in blue) and Max Verstappen(in red)
                 for Abu Dhabi Grand Prix 2021 for the selected lap.")
        )
      )
    )
  )
)
```

In this code chunk, first we create a sidebar chunk which would contain a dropdown to select the lap number and also a text stating a “Note”. Then we create a main panel which contains 2 windows- 1st window is for displaying the plot of laptimes and 2nd window is to display the telemetry of the lap that is selected from the dropdown.

```
# Define server
server <- function(input, output) {

  # Generate lap times plot
  output$lap_times_plot <- renderPlot({

    #converts time in the format "mm:ss" to seconds.
time_sec <- unlist(lapply(strsplit(lap_times$time, split = ":", fixed = TRUE)
                          , function(xx){ as.numeric(xx[1])*60 + as.numeric(xx[2]) }))
```

```r
#Adds a new column to the "lap_times" data frame with the time in seconds.
lap_times$time_sec <- time_sec


#Converts the "driverId" column to a factor and subsets the data frame
#to include only data from two drivers in a specific race.
lap_times$driverId_fac <- as.factor(lap_times$driverId)
lap_times_subset <- subset(lap_times, raceId == 1073 & (driverId == 1 | driverId == 830))

#creates a line plot of lap times for the selected drivers
#with lap number on the x-axis and time in seconds on the y-axis.
ggplot(lap_times_subset, aes(x = lap, y = time_sec, colour = driverId_fac)) +
geom_line() +
  scale_colour_manual(values = c("steelblue", "darkred")
                      , name = "Driver", breaks = c("1", "830")
                      , labels = c("Lewis Hamilton", "Max Verstappen")) +
                        labs(y = "Lap Time (sec)", x = "Lap") +
                        theme(legend.position = "top")
  })
```

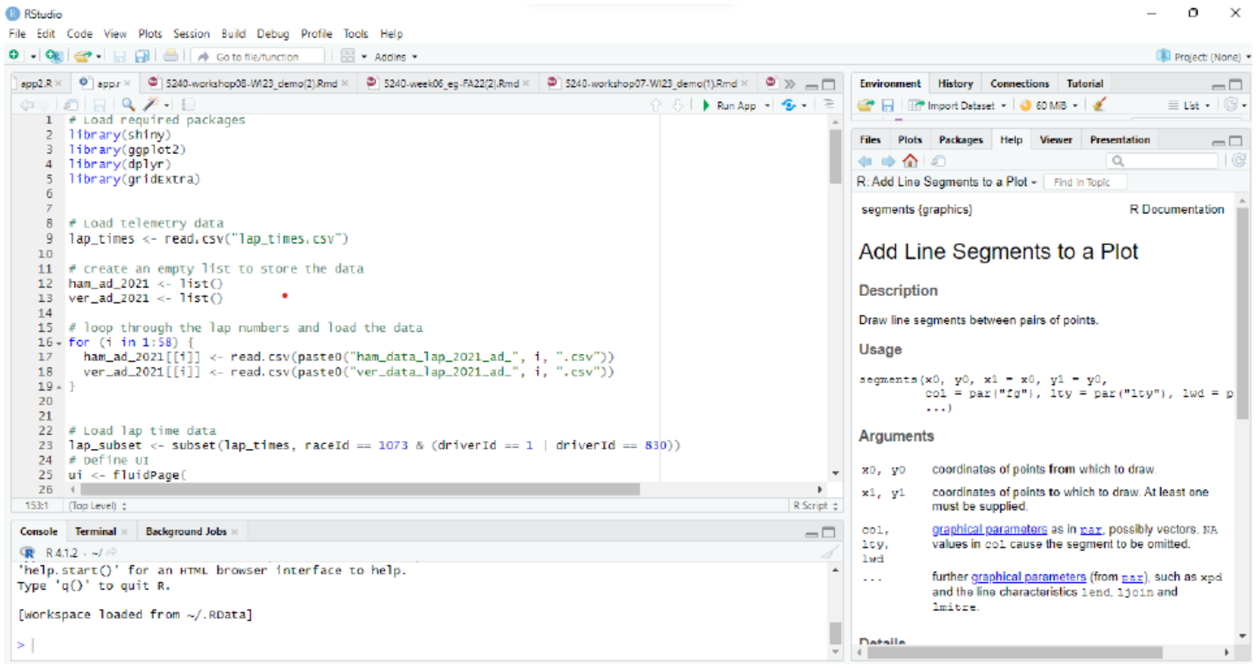
This code chunk generates a plot of lap times for two drivers (Lewis Hamilton and Max Verstappen) in a specific race (raceId == 1073). The lap times are initially stored as a character vector in the lap_times data frame. The lapply function is used to convert the lap times from minutes:seconds format to seconds format, and the resulting numeric vector is added as a new column to lap_times called time_sec. The driverId column in lap_times is then converted to a factor using as.factor, and a subset of the data frame is created to include only the laps for the two specified drivers. The ggplot function is used to create the plot, with lap number on the x-axis and lap time in seconds on the y-axis. The color of the line is determined by the driverId_fac column (which was created earlier as a factor). The scale_colour_manual function is used to specify the colors of the lines for the two drivers. The labs function is used to add axis labels, and the theme function is used to move the legend to the top of the plot. Overall, this code generates a simple but informative visualization of lap times for two drivers in a specific race.

```r
# Generate telemetry table
output$telemetry_table <- renderPlot({
#Read in the lap data for Max Verstappen and Lewis Hamilton
  for(i in 1:58)
  {
    if (input$lap_select == i)
    {
    ver_df <-ver_ad_2021[[i]]
    ham_df <-ham_ad_2021[[i]]

    #Convert the Brake column to a binary indicator variable
    ver_df$Brake <- ifelse(ver_df$Brake, 1, 0)
    ham_df$Brake <- ifelse(ham_df$Brake, 1, 0)

  #Create five plots: Speed vs Distance, Throttle vs Distance,Brake vs Distance
  #Gear number v/s Distance,RPM v/s Distance
    p1 <- ggplot(data = ver_df, aes(x = Distance, y = Speed, color = "Max Verstappen")) +
      geom_line() +
      geom_line(data = ham_df, aes(x = Distance, y = Speed, color = "Lewis Hamilton")) +
      labs(y = "Speed(km/h)", x = "Distance(in meters)") +
      scale_color_manual(values = c("darkred", "steelblue"),
                     name = "Max Verstappen vs Lewis Hamilton",
                     breaks = c("Max Verstappen", "Lewis Hamilton"),
                     labels = c("Max Verstappen", "Lewis Hamilton"))

    p2 <- ggplot(data = ver_df, aes(x = Distance, y = Throttle, color = "Max Verstappen"))+
      geom_line() +
      geom_line(data = ham_df, aes(x = Distance, y = Throttle, color = "Lewis Hamilton"))+
      labs(y = "Throttle(%)", x = "Distance(in meters)") +
      scale_color_manual(values = c("darkred", "steelblue"),
                     name = "Max Verstappen vs Lewis Hamilton",
                     breaks = c("Max Verstappen", "Lewis Hamilton"),
                     labels = c("Max Verstappen", "Lewis Hamilton")) +
      guides(color = guide_legend(override.aes = list
                            (title = "Max Verstappen vs Lewis Hamilton")))

    p3 <- ggplot(data = ver_df, aes(x = Distance, y = Brake, color = "Max Verstappen")) +
      geom_line() +
      geom_line(data = ham_df, aes(x = Distance, y = Brake, color = "Lewis Hamilton")) +
      labs(y = "Brake", x = "Distance(in meters)") +
      scale_color_manual(values = c("darkred", "steelblue"),
                     name = "Max Verstappen vs Lewis Hamilton",
                     breaks = c("Max Verstappen", "Lewis Hamilton"),
                     labels = c("Max Verstappen", "Lewis Hamilton")) +
      guides(color = guide_legend(override.aes = list(
                            title = "Max Verstappen vs Lewis Hamilton")))

    p4 <- ggplot(data = ver_df, aes(x = Distance, y = nGear, color = "Max Verstappen")) +
      geom_line() +
      geom_line(data = ham_df, aes(x = Distance, y = nGear, color = "Lewis Hamilton")) +
      labs(y = "Gear number", x = "Distance(in meters)") +
      scale_color_manual(values = c("darkred", "steelblue"),
                     name = "Max Verstappen vs Lewis Hamilton",
                     breaks = c("Max Verstappen", "Lewis Hamilton"),
                     labels = c("Max Verstappen", "Lewis Hamilton")) +
      guides(color = guide_legend(override.aes = list(title =
                                      "Max Verstappen vs Lewis Hamilton")))


    p5 <- ggplot(data = ver_df, aes(x = Distance, y = RPM, color = "Max Verstappen")) +
      geom_line() +
      geom_line(data = ham_df, aes(x = Distance, y = RPM, color = "Lewis Hamilton")) +
      labs(y = "RPM", x = "Distance(in meters)") +
      scale_color_manual(values = c("darkred", "steelblue"),
                     name = "Max Verstappen vs Lewis Hamilton",
                     breaks = c("Max Verstappen", "Lewis Hamilton"),
                     labels = c("Max Verstappen", "Lewis Hamilton")) +
      guides(color = guide_legend(override.aes = list(title =
                                      "Max Verstappen vs Lewis Hamilton")))

    #Arranges four plots (p1,p2,p3,p4 and p5) in a vertical layout.
    grid.arrange(p1, p2, p3, p4, p5 ,ncol = 1, nrow = 5)
}}})
}
```

This code chunk is a part of a Shiny app that generates telemetry plots for two Formula 1 drivers, Max Verstappen and Lewis Hamilton. The app allows the user to select a lap number using a slider input, and based on the selected lap, the code generates telemetry plots for both drivers for that lap. The code first checks which lap has been selected using the input$lap_select variable. It then retrieves the corresponding data frames for both drivers for that lap from two different lists, ver_ad_2021 and ham_ad_2021. The code then preprocesses the data by converting the binary Brake variable to 0s and 1s using the ifelse function. After preprocessing, the code generates five different telemetry plots using ggplot2, each for a different variable, including speed, throttle, brake, gear number, and RPM. The grid.arrange function is used to arrange all the generated plots in a single column using ncol = 1 and nrow = 5 parameters. Finally, the renderPlot function is used to generate the plot in the Shiny app's output section named telemetry_table.

Instructions

1.Open the app.r file via R studio. You should get a window as shown in figure below.



```
# Load required packages
library(shiny)
library(ggplot2)
library(dplyr)
library(gridExtra)


# Load telemetry data
lap_times <- read.csv("lap_times.csv")

# create an empty list to store the data
ham_ad_2021 <- list()
ver_ad_2021 <- list()

# loop through the lap numbers and load the data
for (i in 1:58) {
  ham_ad_2021[[i]] <- read.csv(paste0("ham_data_lap_2021_ad_", i, ".csv"))
  ver_ad_2021[[i]] <- read.csv(paste0("ver_data_lap_2021_ad_", i, ".csv"))
}


# Load lap time data
lap_subset <- subset(lap_times, raceId == 1073 & (driverId == 1 | driverId == 830))
# Define UI
ui <- fluidPage(
```

```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

> 
```

segments {graphics} R Documentation

Add Line Segments to a Plot

Description

Draw line segments between pairs of points.

Usage

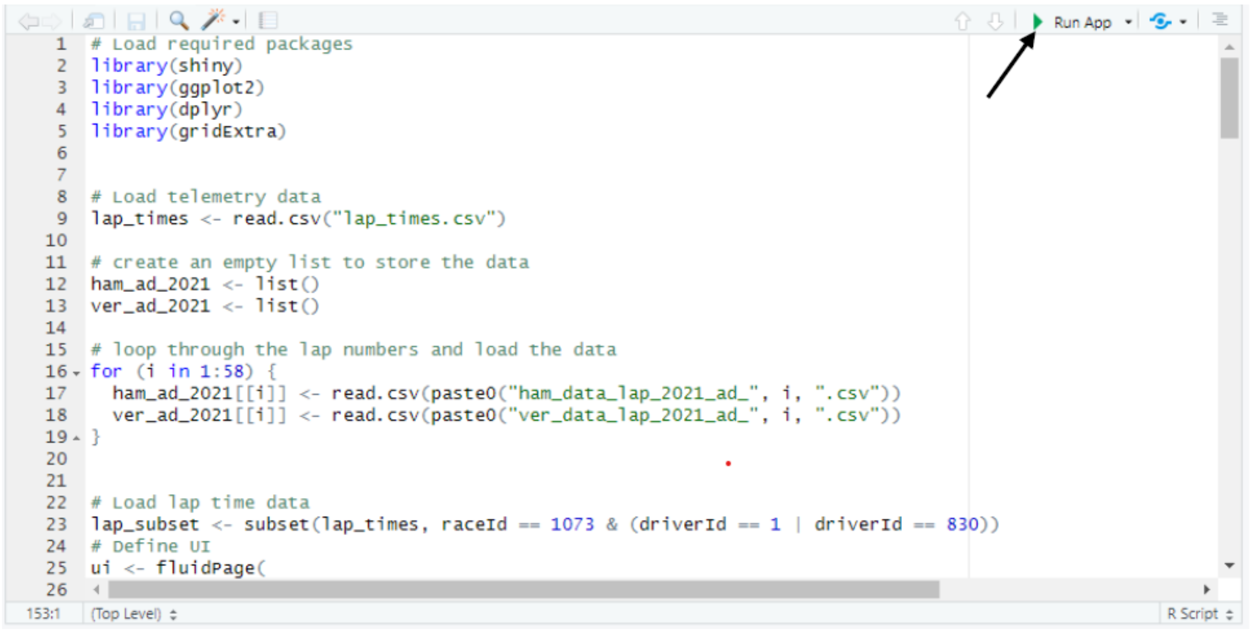
```
segments(x0, y0, x1 = x0, y1 = y0,
         col = par("fg"), lty = par("lty"), lwd = p
         ...)
```

Arguments

x0, y0	coordinates of points from which to draw.
x1, y1	coordinates of points to which to draw. At least one must be supplied.
col, lty, lwd	graphical parameters as in par, possibly vectors. NA values in col cause the segment to be omitted.
...	further graphical parameters (from par), such as xpd and the line characteristics lend, ljoin and lmitre.

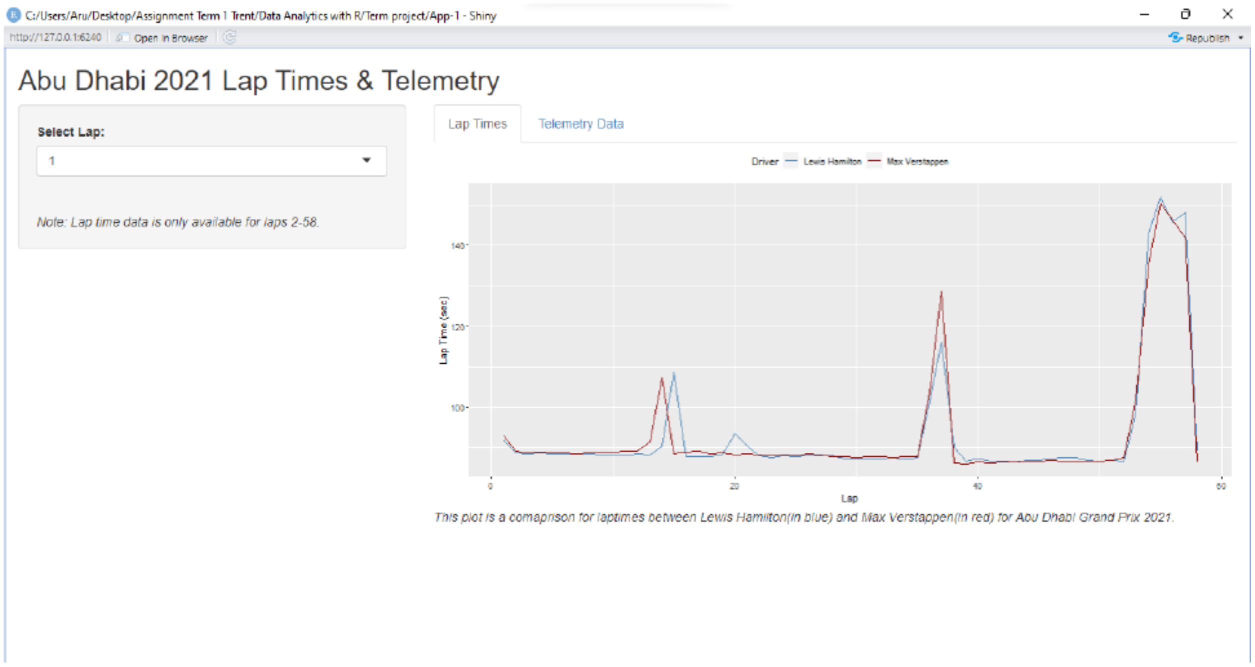
Details

2.The left top pane contains the code about the functionality and UI of the app. If you want to run the app, press the “Run App” button as which is located as shown in below image.



```
# Load required packages
library(shiny)
library(ggplot2)
library(dplyr)
library(gridExtra)


# Load telemetry data
lap_times <- read.csv("lap_times.csv")

# create an empty list to store the data
ham_ad_2021 <- list()
ver_ad_2021 <- list()

# loop through the lap numbers and load the data
for (i in 1:58) {
  ham_ad_2021[[i]] <- read.csv(paste0("ham_data_lap_2021_ad_", i, ".csv"))
  ver_ad_2021[[i]] <- read.csv(paste0("ver_data_lap_2021_ad_", i, ".csv"))
}


# Load lap time data
lap_subset <- subset(lap_times, raceId == 1073 & (driverId == 1 | driverId == 830))
# Define UI
ui <- fluidPage(
```

3.After pressing the “Run App” button, you should get a window pane opened in an R studio.



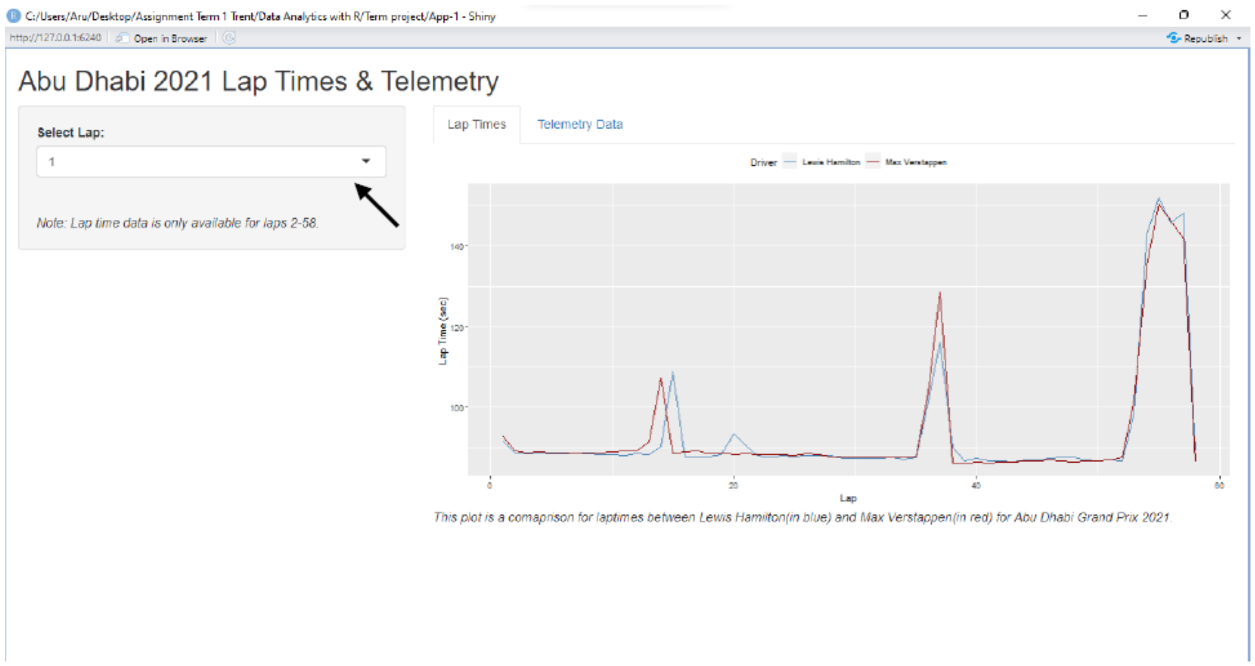
Abu Dhabi 2021 Lap Times & Telemetry

Select Lap: 1

Note: Lap time data is only available for laps 2-58.

This plot is a comaprison for laptimes between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021.

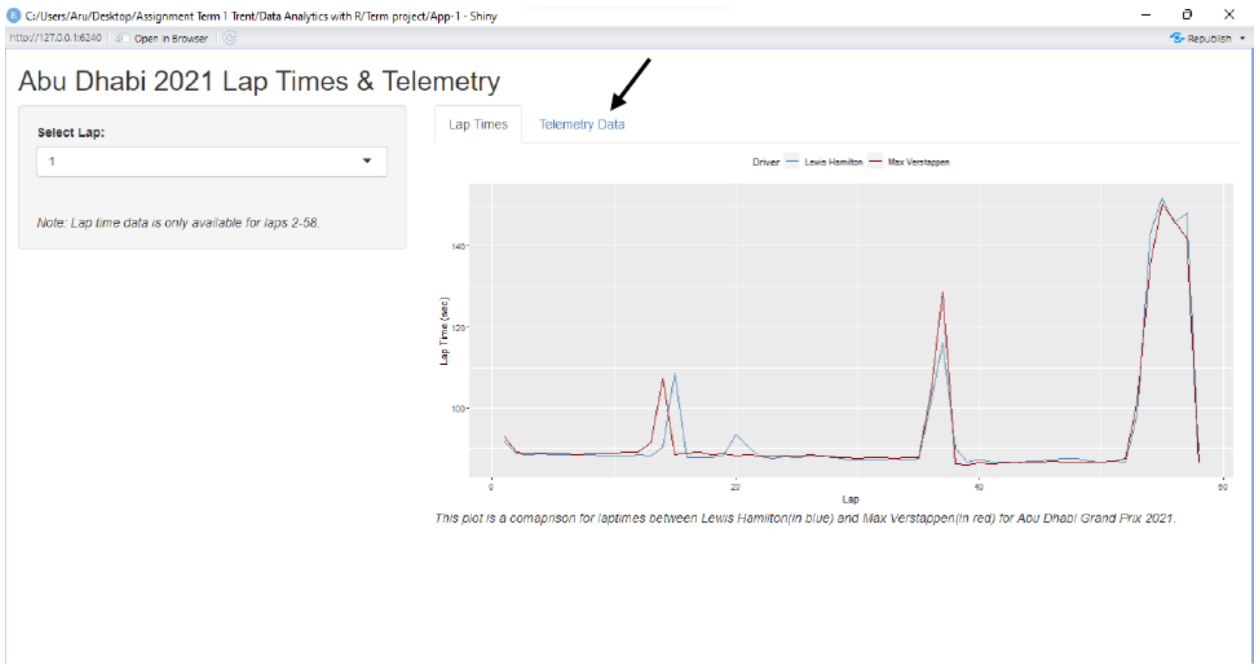
4.In this, the lap number (1-58) can be selected via a dropdown button on the left side panel.



Note: Lap time data is only available for laps 2-58.

This plot is a comaprison for laptimes between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021.

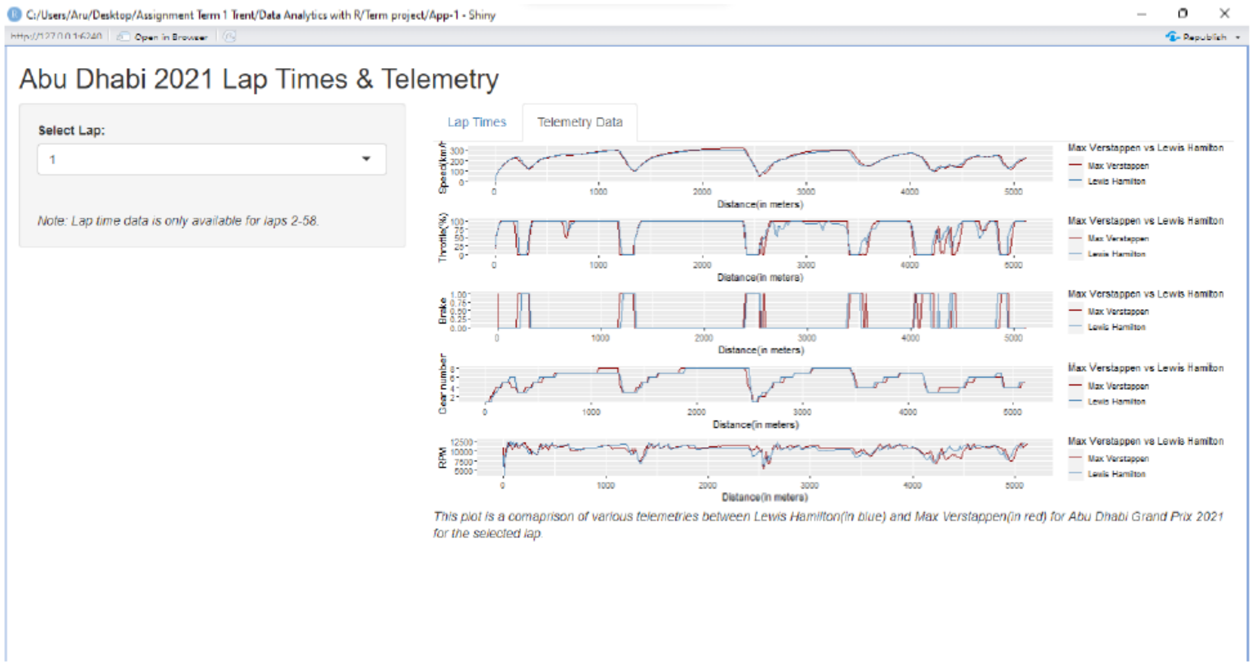
5.We have 2 windows in the main panel, one for Lap time plot of both drivers(Max Verstappen and Lewis Hamilton) for the whole race and other window for Telemetry plot for the lap selected from drop down. To see the telemetry plot, click on the ”Telemetry Data” button on the main panel.



Note: Lap time data is only available for laps 2-58.

This plot is a comaprison for laptimes between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021.

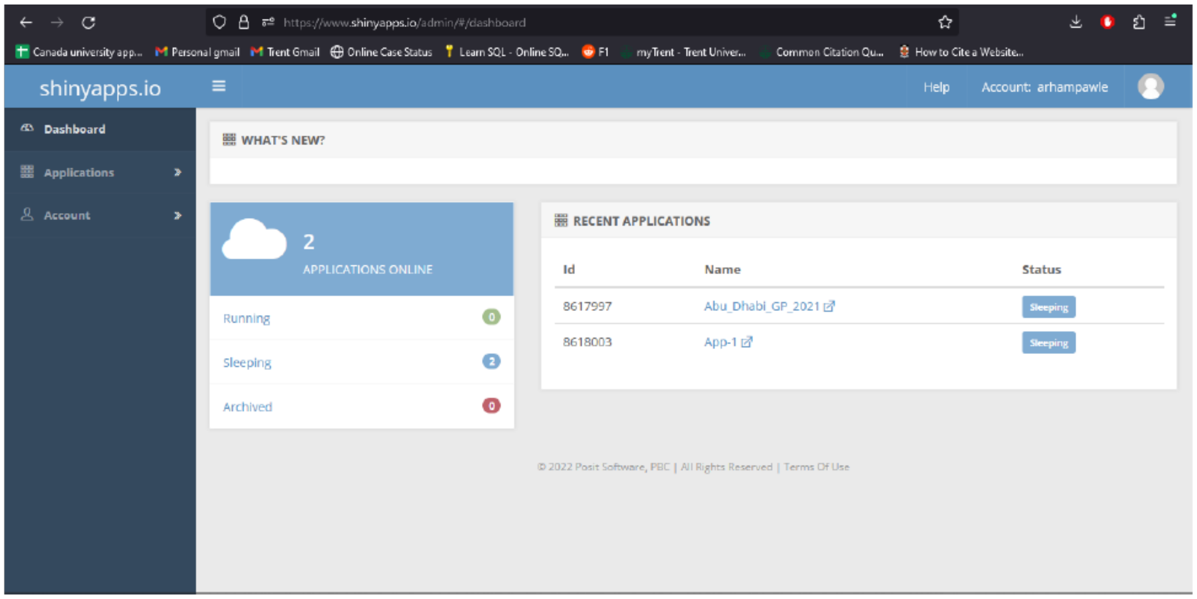
6.You should get a window pane as shown below.


C:/Users/Aru/Desktop/Assignment Term 1 Trent/Data Analytics with R/Term project/App-1 - Shiny
Open in Browser
Republish
Abu Dhabi 2021 Lap Times & Telemetry
Select Lap:
1
Lap Times
Telemetry Data
Speed(km/h)
Throttle(%)
Brake
Gear number
RPM
Distance(in meters)
Max Verstappen vs Lewis Hamilton
Max Verstappen
Lewis Hamilton

Note: Lap time data is only available for laps 2-58.

This plot is a comaprison of various telemetries between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021 for the selected lap.

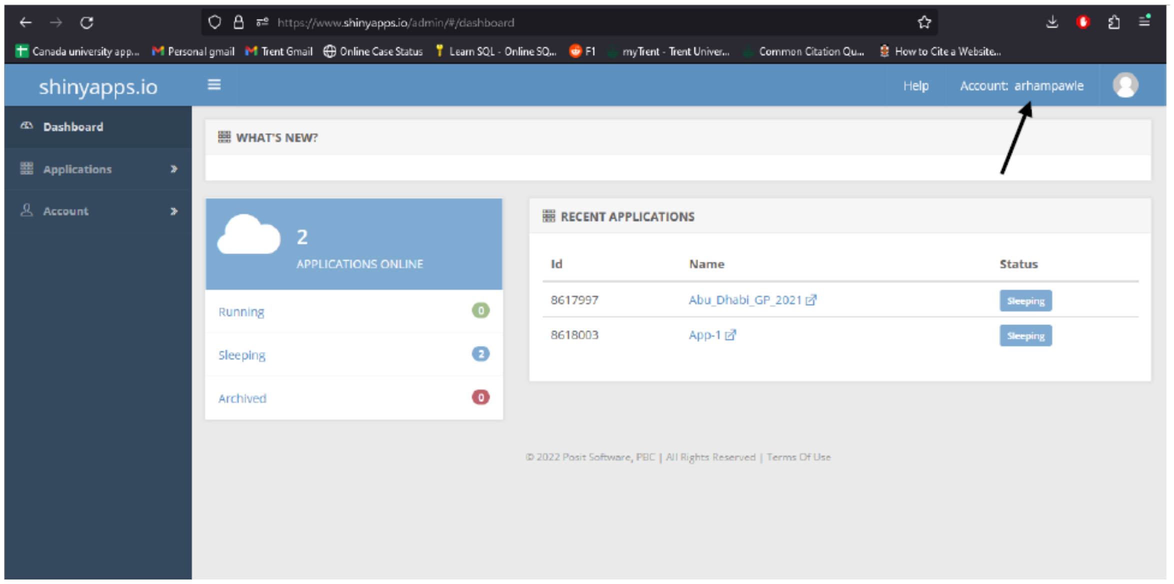
7. If you want to publish in this app via a shareable link, go to www.shinyapps.io, create a account and you should get a dashboard as shown below.



Id	Name	Status
8617997	Abu_Dhabi_GP_2021	Sleeping
8618003	App-1	Sleeping

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

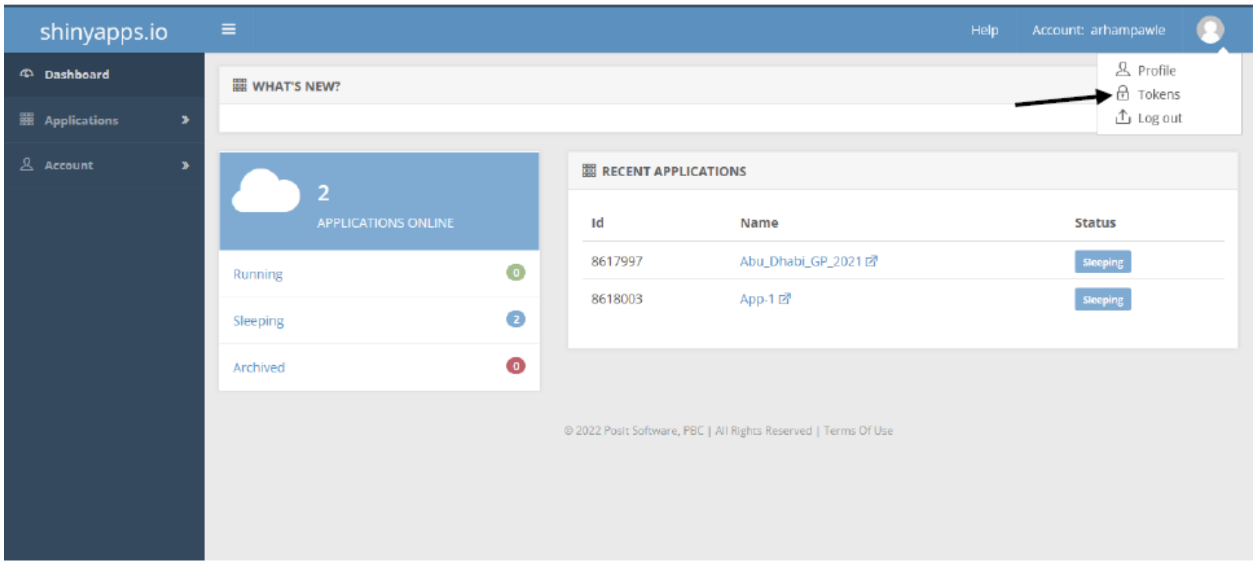
8.Now we need to create a token for our app which can be done by clicking the profile picture right to the username on the top right of the window pane.



Id	Name	Status
8617997	Abu_Dhabi_GP_2021	Sleeping
8618003	App-1	Sleeping

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

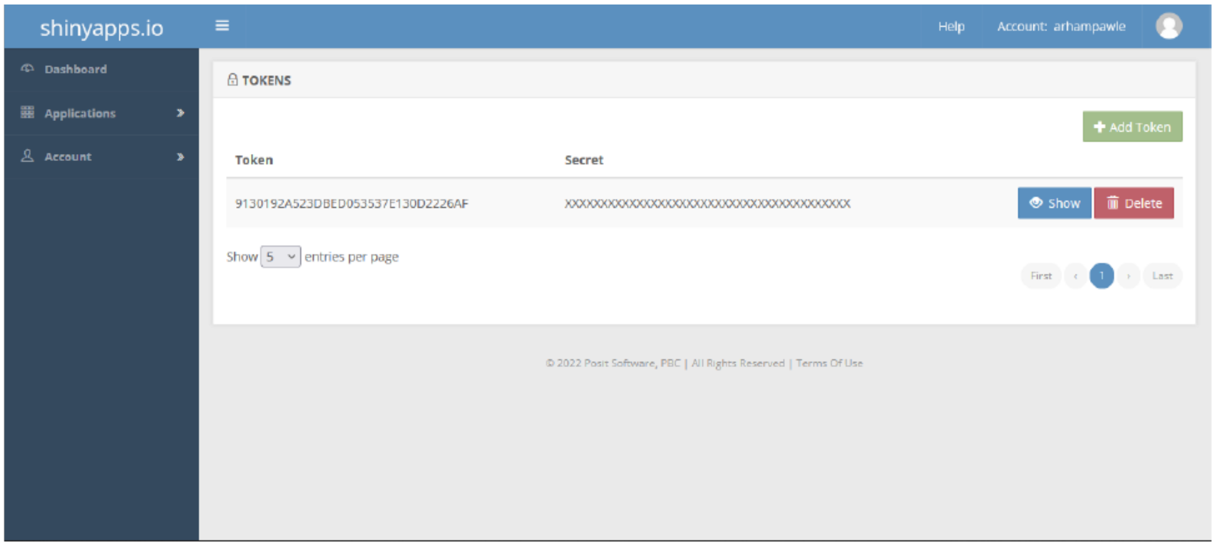
9.You should get a dropdown menu when you click the profile picture, select "Tokens" from the menu.



Id	Name	Status
8617997	Abu_Dhabi_GP_2021	Sleeping
8618003	App-1	Sleeping

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

10. You should get the Token page containing all the tokens that are being used if you have created any R shiny app prior.

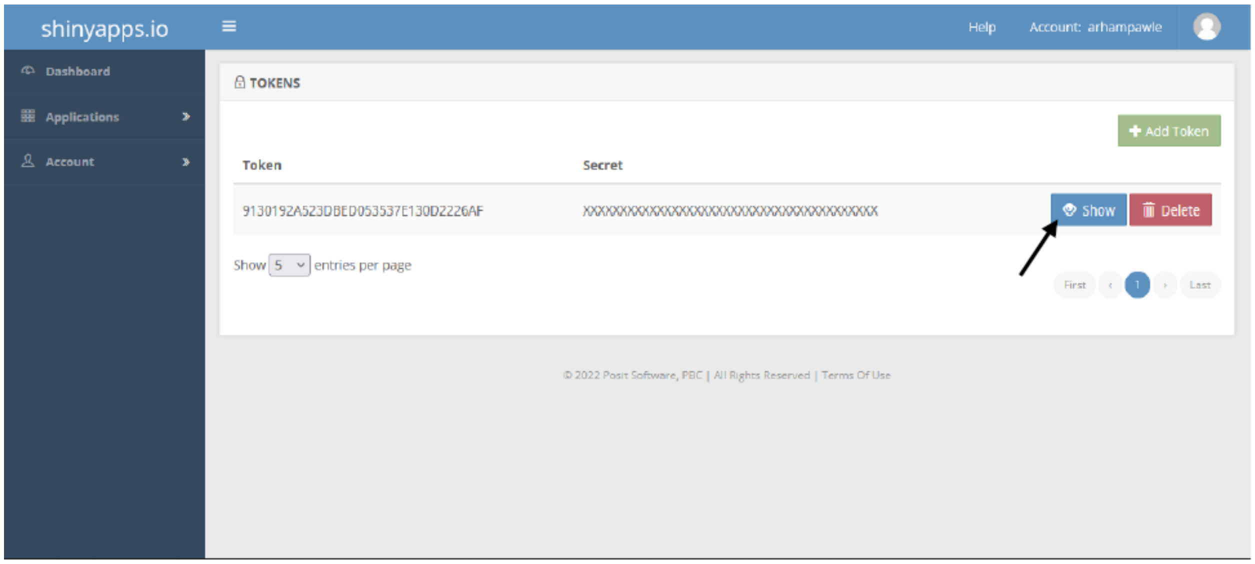


Token	Secret
9130192A523DBED053537E130D2226AF	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Show 5 entries per page

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

11. Click on the show button to reveal the "Secret"

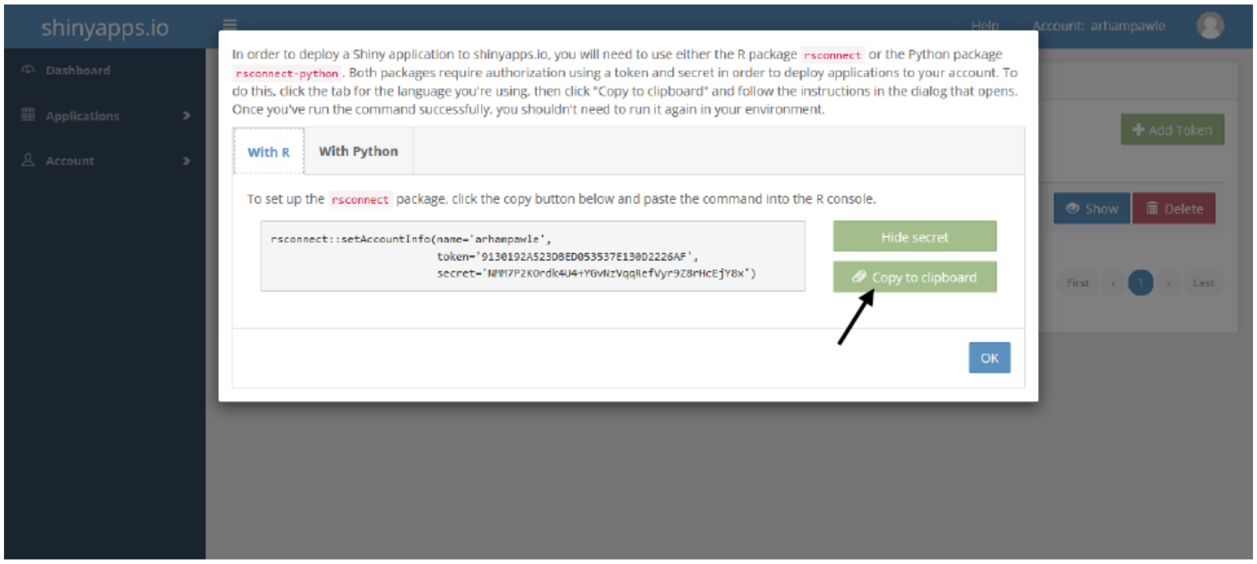


Token	Secret
9130192A523DBED053537E130D2226AF	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Show 5 entries per page

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

12.Click on the “Copy to clipboard” button to get the Token for the app.

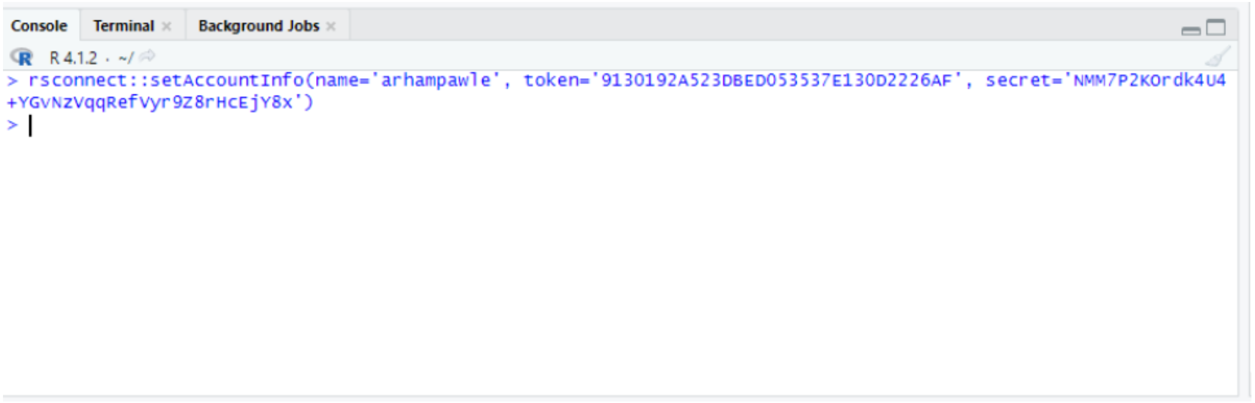


In order to deploy a Shiny application to shinyapps.io, you will need to use either the R package `rsconnect` or the Python package `rsconnect-python`. Both packages require authorization using a token and secret in order to deploy applications to your account. To do this, click the tab for the language you're using, then click "Copy to clipboard" and follow the instructions in the dialog that opens. Once you've run the command successfully, you shouldn't need to run it again in your environment.

To set up the `rsconnect` package, click the copy button below and paste the command into the R console.

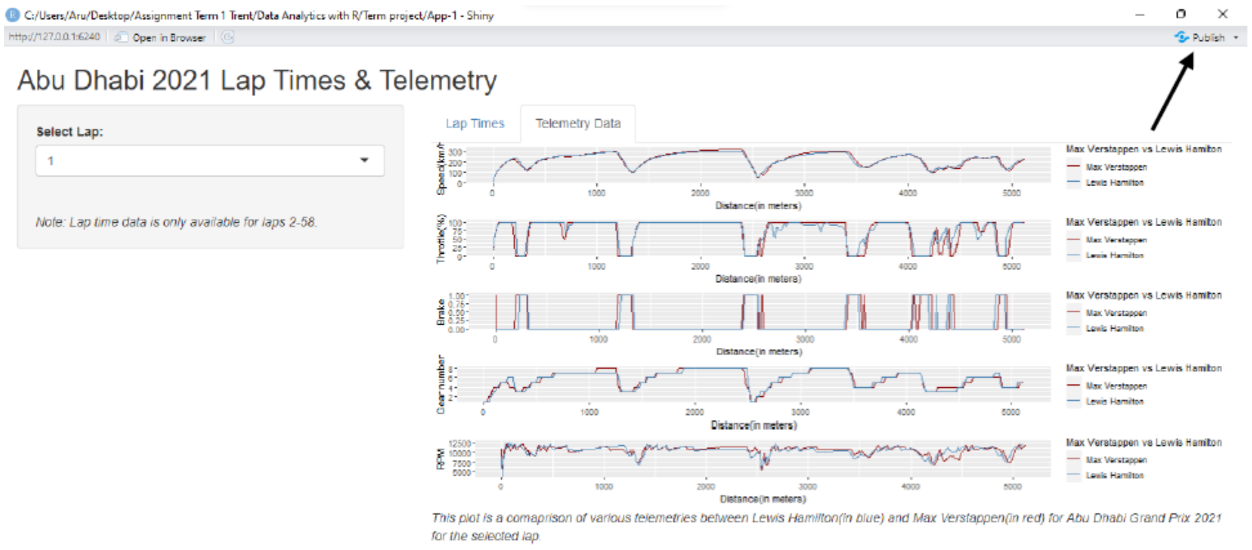
```
rsconnect::setAccountInfo(name='arhampawle',
                          token='9130192A523DBED053537E130D2226AF',
                          secret='NMM7P2KOrdk4U4+YGvNzVqqRefVyr9Z8rHcEjY8x')
```

13.Paste the token in the Console of the R studio and press the Enter button from your keyboard.



```
> rsconnect::setAccountInfo(name='arhampawle', token='9130192A523DBED053537E130D2226AF', secret='NMM7P2KOrdk4U4
+YGvNzVqqRefVyr9Z8rHcEjY8x')
> 
```

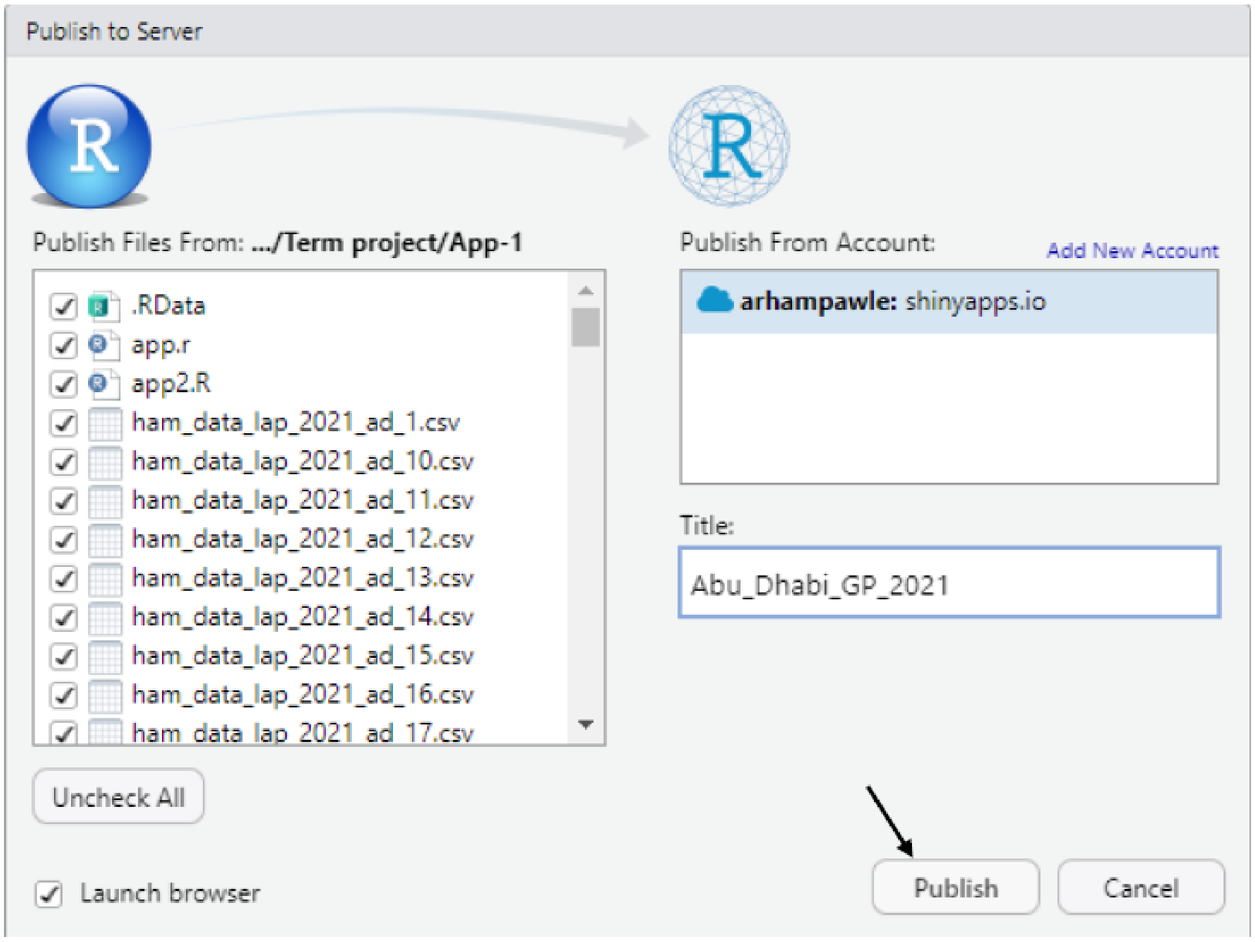
14.If you want to publish in this app via a shareable link, click the button “Publish” button.



Note: Lap time data is only available for laps 2-58.

This plot is a comaprison of various telemetries between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021 for the selected lap

15.You should get a window as shown in below pic. Enter the Title of your choice and press the “Publish” button.



Publish to Server

Publish Files From: **.../Term project/App-1**

- ✓ .RData
- ✓ app.r
- ✓ app2.R
- ✓ ham_data_lap_2021_ad_1.csv
- ✓ ham_data_lap_2021_ad_10.csv
- ✓ ham_data_lap_2021_ad_11.csv
- ✓ ham_data_lap_2021_ad_12.csv
- ✓ ham_data_lap_2021_ad_13.csv
- ✓ ham_data_lap_2021_ad_14.csv
- ✓ ham_data_lap_2021_ad_15.csv
- ✓ ham_data_lap_2021_ad_16.csv
- ✓ ham_data_lap_2021_ad_17.csv

Uncheck All

✓ Launch browser

Publish From Account: Add New Account

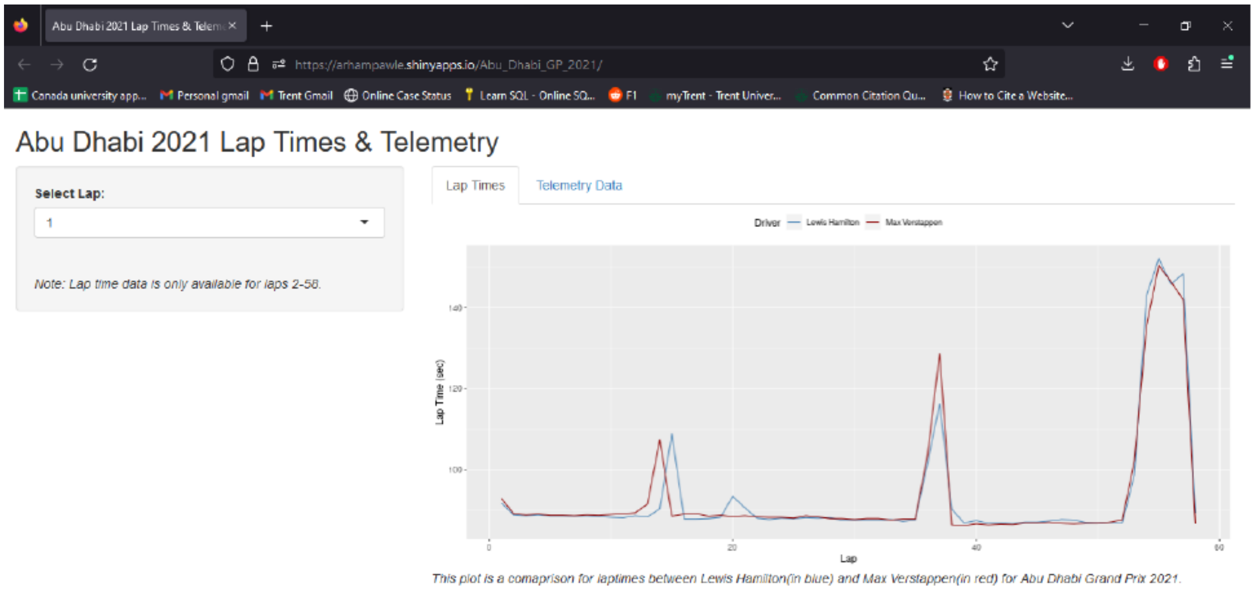
arhampawle: shinyapps.io

Title:

Abu_Dhabi_GP_2021

Publish Cancel

16.After publishing the “Publish” button , it takes few minutes(installs dependencies) before opening the app on your default browser as shown in pic.

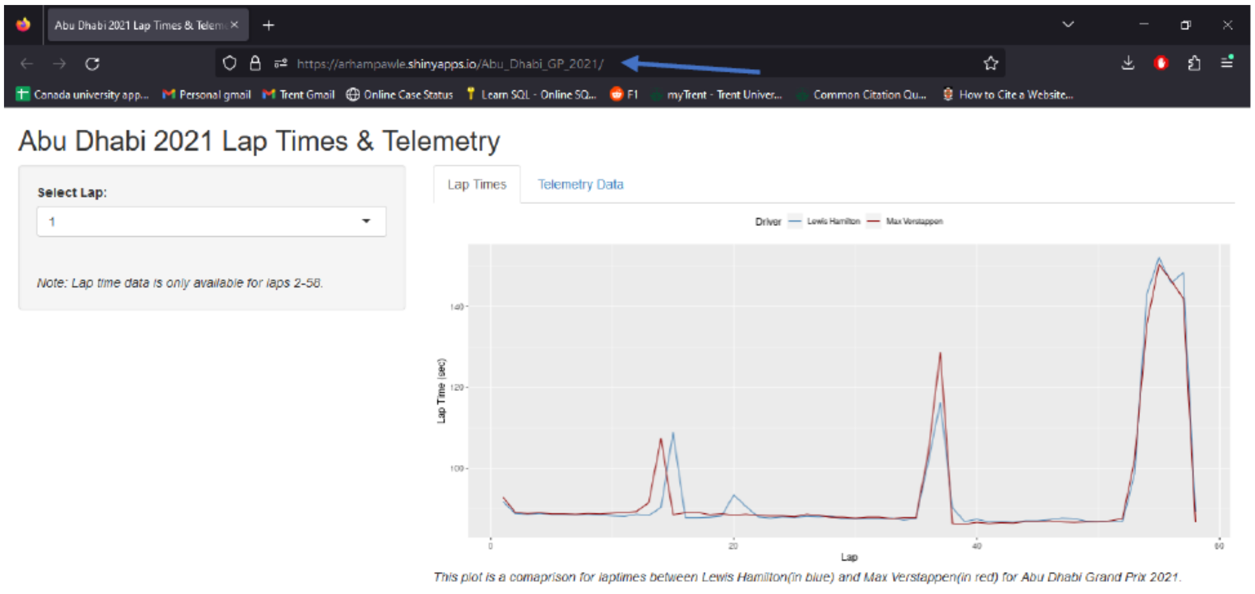

Abu Dhabi 2021 Lap Times & Telem
https://arhampawle.shinyapps.io/Abu_Dhabi_GP_2021/
Canada university app... Personal gmail Trent Gmail Online Case Status Learn SQL - Online SQ... F1 myTrent - Trent Univer... Common Citation Qu... How to Cite a Website...
Select Lap:
1
Lap Times
Telemetry Data
Driver
Lewis Hamilton
Max Verstappen
Lap Time (sec)
Lap

Abu Dhabi 2021 Lap Times & Telemetry

Note: Lap time data is only available for laps 2-58.

This plot is a comaprison for laptimes between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021.

17.The link in the URL bar is a shareable link (https://arhampawle.shinyapps.io/Abu_Dhabi_GP_2021/) which can be shared with anyone irrespective if they have R installed in their device or not.


Abu Dhabi 2021 Lap Times & Telem
https://arhampawle.shinyapps.io/Abu_Dhabi_GP_2021/
Canada university app... Personal gmail Trent Gmail Online Case Status Learn SQL - Online SQ... F1 myTrent - Trent Univer... Common Citation Qu... How to Cite a Website...
Select Lap:
1
Lap Times
Telemetry Data
Driver
Lewis Hamilton
Max Verstappen
Lap Time (sec)
Lap

Abu Dhabi 2021 Lap Times & Telemetry

Note: Lap time data is only available for laps 2-58.

This plot is a comaprison for laptimes between Lewis Hamilton(in blue) and Max Verstappen(in red) for Abu Dhabi Grand Prix 2021.

Limitations

One limitation of this app (but not restricted to) that you described could be that it only provides data for the Abu Dhabi GP 2021. This means that the app may not be useful for people looking for data from other Grand Prix races or seasons. The app's functionality is limited to only one race, which could limit its usefulness to a broader audience of motorsport enthusiasts or analysts.

Another potential limitation could be that the app only displays lap times and telemetry data from a selected lap. This means that users cannot compare multiple laps or create custom analyses based on different combinations of laps or drivers. This may limit the app's usefulness for those who want to perform more in-depth analysis or create their custom reports based on data from multiple laps or drivers.

Lastly, since the app is hosted on a server, it may be subject to downtime or slow response times, especially during periods of high usage. This could limit the app's availability to users who need real-time access to data for analysis or decision-making purposes.

Keeping all these limitations in mind ,we can cover all the above points during the next iteration of the prototype.