**FEED FOR SPEED**

**Project Diary Entry**

**R - Language**

Team Members : Naman Rastogi and Shivanshu Tandon

1. **Week 1 – [21/09/2023 – 28/09/2023]**

For the very first we scraped all the data from official formula1 website, including the tables ‘Race Results’, ‘Starting Grid Position’, ‘Pitstop Summary Table’, ‘Fastest Laps’. And automated the process of scraping also based on user input. Since the bigger goal of our project was to make the web app that works intuitively as per user’s command. We worked on making functions for scraping the required data based on input.

1. **Week 2 – [29/09/2023 – 06/10/2023]**

Overlapping Features – Our next goal was to make something useful around overlaps of all the races for all the drivers from 1950 – 2022. The idea roughly documented during the time is mentioned below and was achieved successfully during the second week.

Aim: To evaluate which circuits have been the most interesting in terms of overtakes.  
  
Description: As we know more the number of overtakes the more interesting the race is. So, we will select 1 circuit (get this from user). We will scrape the data for this circuit for each year. We will get the number of times a race has been held at that circuit.  
  
Next, we will scrape the driver results data for each year for that circuit. For a particular year, we will calculate the “Finger factor”.   
  
Fresh Idea: Whenever we see a race on TV, the cameraman is interested only in the interesting content. Interesting content is overtaking, understeer, oversteer, collisions, car failure, pitstop time and sector 1,2,3 comparison of drivers. Let’s take the overtakes first.   
  
Overtakes can happen in multiple scenarios.  
1. In a normal race where both cars are fit and fighting to go faster. (The ideal case). Now if we just compare the starting grid and ultimate race results, it won’t consider the DRS zone fights where the drivers keep on overtaking each other without effecting the end positions. We want to take these cases as well as they are exciting.  
2. Overtake as someone took pitstop so you overtook him. This might not sound interesting as you just passed the car in pitstop, but you see that car will come back with fresh cars and overtake again. Now this is interesting.  
3. Overtakes due to car failure, oversteering, understeering, driver error again, these are not interesting but as an overtake is an overtake we will consider this case as well. Think from a driver’s perspective, you want to overtake everyone no matter whether it is an easy overtake or difficult.  
4. Overtake during a slow pitstop. A case where, you both were at Px and Py, both of you entered the pitstop at the same time but as your crew messed up so you lost your position. This is interesting.  
  
So any overtake is interesting and hence proved that total number of individual overtakes throughout the race will make the race more exciting.  
  
So, to get that, we will scrape the data from https://www.statsf1.com/en/2022/italie/tour-par-tour.aspx.  
  
We need to get the table in Lap by Lab tab. That table has the data for all positions on a lap-by-lap basis. This is better as we will calculate the positions gained after each lap. We will just take the positive number. We will ignore positions lost. For the positions gained, we will calculate it for each lap and add it to get the final number of overtakes. This will give a good number for how interesting the race was as a battle of speed.   
  
Also, it will reflect on the number of overtakes taken by each driver. Along with that you will get the data for number of times that driver was overtaken throughout the race.  
  
So, you will get a table where you will have starting grid position, race position, number of overtakes made and number of times driver was overtaken.  
  
The sum of total overtakes by drivers will give the “Finger factor” for that circuit for that year. We will do the same analysis for the same circuit over the years and get the “Finger factor” for each year. We will do an average for all the finger factors which will provide an average “Finger factor” for that circuit. This will work as an attribute for that circuit.  
  
This way by comparing the “Average Finger Factor” we can comment on the circuit whether it is an interesting circuit to watch or not. In further analysis we can comment on the circuit whether it is a “STREET CIRCUIT” or a normal circuit.  
  
Also, we can calculate the fight between 2 drivers in the number of overtakes. It will give the number of times any driver has overtaken the other.

1. **Week 3 – [06/10/2023 – 13/10/2023]**

Description:

Our next target was to make different charts like animated bar race, line plot for overlaps, gauge charts for circuits, where all drivers will be on y axis, laps will be on x axis and we will run a race plot till finish. In this animation, you'll see lines moving from the start to the finish line, each line representing a different driver. As they race, you'll be able to see when they pass each other and how they progress through each lap.

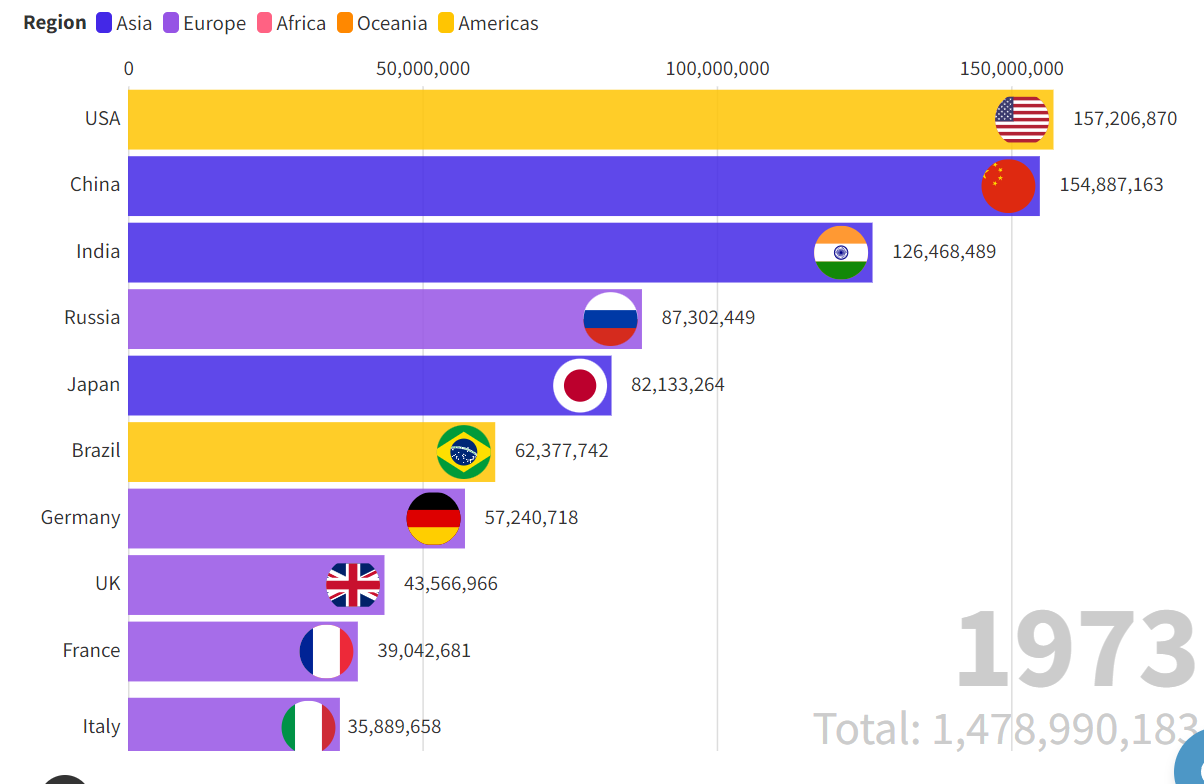
Which was implemented using ggplot2 and gganimate using the lap-by-lap data present on statsf1.com website. It was achieved by calculating the position of each driver at the end of every lap and storing it in a data frame which then was converted to a long dataframe to plot as a bar chart that was animated using transition\_states from gganimate package. The idea is documented below:

1. **Race Bar Chart:**

We're creating this graph to make it easy for beginners to grasp the race dynamics. In this graph, each horizontal bar represents a driver, and the length of the bar reflects their position in each lap. To generate a pseudo-speed indicator for our race chart, we've come up with a formula(Formula 1 :D). We've established fixed points for each lap, which are allocated to drivers based on their positions in that lap.

For example, let's say there are 20 drivers and 53 laps in a race. We assign 20 points(number of drivers) to the driver in 1st position, 19 points to the 2nd position, 18 points to the 3rd position, and so on, decrementing by 1 point for each lower position. These points are then multiplied by the lap number(n) to determine the driver's score for that specific lap.

For instance, if a driver is in the 7th position on the 15th lap, they would receive 14 (position) x 15 (lap) = 210 points for that lap. Importantly, we don't accumulate the points assigned to the driver on each lap; instead, we simply assign and plot these points. This approach allows us to ensure that the longest bar on the graph represents the fastest driver for each lap and, consequently, for the entire race. Ref:



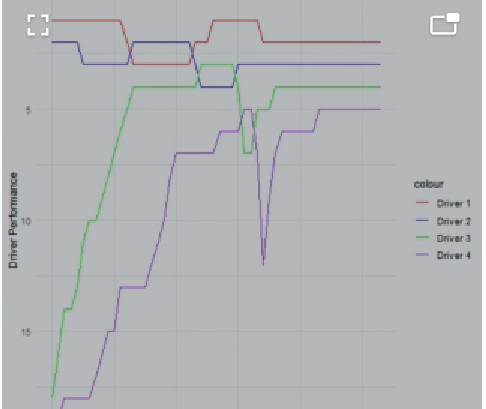
As you can see the bars move with respect to the number of years, in our case it will be number of laps and the value of bars will be the value we assign to the drivers on each lap(Our formula 1). We will be giving a toggle button as well to select the number of bars. This will be user specific.  
  
Using this race chart, we will achieve an impressive visual representation of the entire race, effectively highlighting the race winners and providing a clear indication of overtaking maneuvers as the bars shift whenever an overtake occurs. This visualization will also capture instances where cars retire from the race after a specific lap, as the respective driver's bar will cease advancing at that point.

Therefore, by utilizing both of these graphs, we will gain a comprehensive understanding of the race. The bar chart offers a more prominent visualization of the race winners and those who couldn't complete the race. It also presents the intriguing dynamics of overtaking maneuvers in an aesthetically pleasing manner. On the other hand, the second graph provides a detailed analysis for each driver, showcasing the total number of overtakes through the breaks in their lines, placing greater emphasis on overtaking analysis.  
  
Bar line graph was implemented for head-to-head driver overlaps since all the data was parsed from lap-by-lap table, this graph helps to visualise the overlapping between two drivers selected by the user a particular race, the idea documented is mentioned below:

1. **Basic Line Graph:**

Each line corresponds to an individual driver. The y-axis scale commences from the top, placing the fastest driver at the highest point and the slowest car at the bottom. We are charting the race with laps plotted on the x-axis, and the lines represent the drivers' positions on each lap. When a driver overtakes another, their lines intersect, with the quicker driver's line positioned above the slower one. Each break in the line signifies an overtaking maneuver or a change in position. This approach allows for a detailed analysis of an individual driver's overtaking actions during the race and at which lap. Also, we have inculcated the cases where the car retires after a certain number of laps. In those cases, the line for that driver stops at that respective lap.

Additionally, it provides an overall view of the race's excitement by illustrating the number of overtakes made by each driver, offering insights into the race's dynamics. Ref:

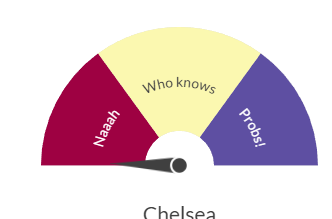


Also, we will be providing a toggle button to select the number of drivers to see the analysis for. This way the user will have the power to see the analysis for the drivers he/she are interested in.

Although the graph throws light on overtaking aspect for multiple drivers, it fails to give a good visual representation for the whole race. Also, it doesn’t take into consideration the speed for each driver as the lines run at the same pace. In order to give it a better look and inculcate the speed of each driver, we have introduced race bar chart to fulfil our aim.

1. **Gauge Charts:**

As, we have successfully done the race analysis and the overtaking analysis, we will now do the circuit analysis. We will use the Finger factor to decide which circuit was more interesting and which was less. For this as discussed above in Aim 1. We will calculate the “Finger Factor” for all the circuits and get an average for those numbers. Let that be called the “mean finger factor”. Now we will use gauge chart to represent the “Finger factor” for our circuit as shown:



The mean finger factor will lie in the middle and the extreme finger factors will lie on the extremes. So, whenever we input a circuit, the output should provide the finger factor on this chart with the needle pointing to its value. On extreme right we will have “Red Hot” Color, in middle we will have “Yellow” and on left we will have “Cold blue”. This will be a good way to represent how interesting that race was.  
  
We will be giving 2 gauge charts. So, the first gauge chart will give the finger factor for that race w.r.t that circuit itself. That means suppose the race selected was Monza 2022. Now the mean finger factor for this chart will be the average of finger factor for Monza across years. So, this will depict the hotness of the race w.r.t Monza standards.  
  
The second finger factor will depict the finger factor w.r.t to all the races over the years at every circuit.  
  
We will add hyperlink to extreme values so that user can open cases where there were maximum overtakes and minimum overtakes throughout F1 history. Both of these charts will depict how interesting the race was.

1. Week 4 – **[13/10/2023 – 19/10/2023]**

Final week we implemented the web app using shiny in R and the map section for all the races from 1950 to 2022 to be plotted on the map. The web app was achieved with shiny and map with leaflet R package. The idea documented is mentioned below:

We will be asking the user the year number in a dropdown button. Once the year is selected, we are interested in getting the circuits on a world map like this: 

The aim is to get the locations on the map for each year (it will be different for each year) and the user should just select the location. Once done the race bar chart should come in the next step with the gauge chart as well.  
  
We are planning to make different tabs on our web-app. First tab will have dropdown of years and will give circuit map for the year selected. The user then selects the location on map and we return the race chart along with gauge charts (1 for circuit and 1 overall).

Second tab will be the Lap wise overtaking tab. We will ask the year, location the same way we did in Tab 1. Then we will return a table with Driver name, Total number of overtakes, Starting grid, ultimate race result. Next, we will give toggle button next to each driver to select the driver which will show the line chart for those drivers only. It will have a “Run” button to execute the graph whenever the drivers are selected and the button will be disabled when the graph is rendering.  
  
**Project Diary Entry**

**Julia - Language**

**1. Project Diary Entry - [05/10/2023 - 12/10/2023]**

Team Members : Ashly Thomas Panangadan, Ans Mariya Joy

Task : Scraping and Wrangling Data from Formula1 Racing Website using Julia

Description :

Worked on the scraping and wrangling of data from the Formula1 racing website. Our goal is to collect data on Grand Prix races, race results, drivers, and starting positions for the years 2012 to 2022.

* We started by defining the URL of the Formula1 racing website, "https://www.formula1.com/en/results.html" and specifying the range of years we wanted to scrape, which is from 2012 to 2022.
* We created arrays to store data related to Grand Prix names, race result links, winners, winner IDs, race years, and winner starting positions. We also initialized an empty DataFrame named 'df.'
* For each year in the specified range, we implemented the 'scrape\_data' function. In this function :
* We constructed the URL for the specific year's race results page.
* Sent an HTTP GET request and parsed the HTML content using Gumbo.
* Extracted data from the table, such as the Grand Prix name and links to race results.
* For each Grand Prix, we followed the link to the race results page, retrieved the winner's driver ID, and added data to the arrays.
* For each winner, we took the starting position from the starting grid page using driver ID to identify the row.
* After scraping data for all years, we created a DataFrame 'df' containing columns for Grand Prix names, race years, driver IDs, driver names, and starting positions.
* We used the 'starting\_positions\_counts' function to count the occurrences of each starting position for each Grand Prix.
* The most common starting position for each Grand Prix was determined using the 'most\_common\_position' function.
* We grouped the DataFrame by 'Grand\_prix' and displayed a resulting DataFrame showing the common starting position and the number of occurrences for each Grand Prix.
* We defined a threshold value (3) and filtered the data to select Grand Prix races where the common starting position occurred more than the threshold times.

**2. Project Diary Entry - [13/10/2023 - 18/10/2023]**

Team Members : Ashly Thomas Panangadan, Ans Mariya Joy

Task : Data Analysis - Common Starting Positions in Formula1 Grand Prix Races

Description :

We focused on analyzing the data collected from the Formula1 racing website. Our aim was to identify the common starting positions for Grand Prix races and highlight any exceptions. This analysis provides insights into the starting positions that are most frequently associated with race winners.

* We started by analyzing the data to find the most common starting position for each Grand Prix race. Our analysis revealed that the majority of Grand Prix races had the starting position 1 as the most common. However, we identified exceptions where the most common starting position was not 1 but 2 as well.
* To perform this analysis, we used the data collected in the earlier stages of the project, where we scraped and cleaned data on Grand Prix races, winners, and their starting positions.

Notes:

The analysis revealed that, in most Grand Prix races, starting position '1' was the most common, with the exception of a few races where starting position '2' was more common. This insight can be valuable for understanding race dynamics and potentially developing race strategies in the sport of Formula1.