Project Report: F1 Analysis Tool

Samuel Mata · 300574763 · DATA-472 · Final Project Report

1. Introduction and Motivation

Formula 1 *(more commonly referred to simply as F1)* is an international motorsport league. Specifically, F1 is the highest class of open-wheel *(where wheels are exposed rather than internally contained within the chassis)* single seater racing, and widely seen as the most prestigious and popular form of motorsport.

Motorsports in general tends to have more complex procedures and rules compared to more traditional sports events. This is especially apparent in F1, mostly due to the highly technical nature of building and racing in such complex vehicles. Furthermore, the exact rules and specifications for F1 championships and entrants has changed drastically and continuously with various rule and regulation changes since its inaugural 1950 season.

It is this complexity and highly technical nature that lends towards a sport that is highly data-oriented yet often difficult to understand. In this context, this project *(simply titled "F1 Analysis tool")* aims to provide a solution.

This project developed an interactive web-app allowing users to perform customized analysis on various aspects of current Formula 1 data. The final outcome was largely successful in project aims and can be accessed online at https://sammata.shinyapps.io/f1-app/ . This report covers various aspects of the completed project, including initial goals, data sources, the application itself, and insights made from the development process.

1.1 Goal

The goal of this project is to utilize the large quantities of collected data surrounding F1 to create an interactive tool allowing users to manage and visualize various aspects of the sport. Specifically, this involves the creation of an interactive web application *(built with R and Shiny)* to allow for users to perform queries and analysis on current and historical F1 data. The tool should be accessible to those who may not be familiar with F1 by providing a simple interface for various types of user-selected analysis.

1.2 Target Audience

As the project aim is mostly focused on giving a tool for custom broad high-level analysis of F1, the target audience includes anyone who may need to analyse the sport. This includes those who may be unfamiliar with F1 and want to learn more about the sport, and those who need customized analysis of the sport. This userbase is broad and contains groups with varying degrees of domain knowledge on both statistics and F1, so interpretability and ease of use are highly important.

2. Data Overview

Finding a suitable data source is crucial to allow for a wide range of accurate analysis to be performed. Such a dataset needs to be accurate, extensive, and publicly accessible for this project to be viable. Thankfully datasets are readily available for nearly every detail of F1 due to the inherently high levels of engineering and regulation.

2.1 Source

This project uses the Formula 1 World Championship (1950 – 2023) dataset made publicly available on Kaggle. This dataset extensively details data on all races, drivers, constructors, qualifying sessions, race sessions, circuits, lap times, pit stops, and championships from 1950 *(F1's inaugural season)* to 2023 *(The most recent complete season)*. The data is compiled from the commonly used community-run Ergast API, which is updated live as seasons progress.

This dataset lists a multitude of records, including current and former circuits, constructors *(teams)*, drivers, season results, race results, lap times, pit stop timings, qualifying times, disqualifications, and crashes.

2.2 Use

The aforementioned dataset provides 14 CSV *(comma-separated-value)* files, of which 7 were used in the final product *(See Table 1)*. For the sake of performance *(as all files have to be loaded into system memory)*, unused files were discarded.

While various checks were made throughout development to confirm the authenticity and accuracy of the data, no errors were found. Furthermore, all files were correctly formatted, and no empty values were detected. As such, no preparation steps were required for data cleaning or preprocessing.

2.3 Structure

The final dataset used those 7 files, which contained a total of 28 features. This included many IDs, which are simply used to match data across several files. Excluding the 13 IDs, we are left with 15 features for analysis.

File Name	**Features *(Including IDs)***
constructor_standings.csv	*raceId, constructorId, points, position*
constructors.csv	*constructorId, name, nationality*
driver_standings.csv	*raceId, driverId, points, position*
drivers.csv	*driverId, forename, surname, dob, nationality*
races.csv	*raceId, year, round*
results.csv	*resultId, raceId, driverId, constructorId, position, points, statusId*
status.csv	*statusId, status*

Table 1: Used files and features

3. App Overview

The application is split into 3 tabs, each accessible through a hamburger menu located next to the title. Each tab is used for a different aspect of analysis, depending on what is required by the user, these include:

1. **Season** – The initial default tab, which allows analysis of a specific F1 season for both championship types and with a variety of features.
2. **Driver** – To examine and analyse a specific driver across multiple seasons. Includes driver details, a breakdown of points scored across seasons, average placings, and typical race performance.
3. **Constructor** – For analysis of a specific constructor *(team)* across multiple seasons, with regards to points scoring, and driver contributions.

3.1 Season Tab

The Season tab contains the most UI, with 3 drop down menus:

1. **Championship** – Allowing selection between either the drivers or constructors championship.
2. **Season** – Allowing selection of F1 season by year, ranging across the dataset from 1950 *(the inaugural F1 season)* to 2023 *(the most recent complete season)*.
3. **Feature** – Allowing selection of the feature to be analysed. This can be points, race wins, podiums *(referring to a top 3 race finish for a driver/team)*, and finishes *(referring to a completed race where the driver/team does not crash, become lapped, or otherwise not finish normally)*.

These can be seen in Figure 1 below:

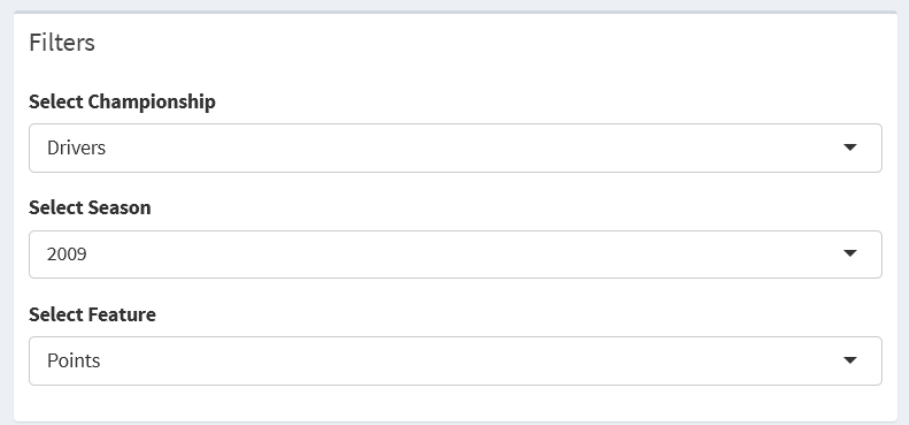

Figure 1: Season Tab Input UI

These inputs are used to create three graphs:

First, a **Championship Plot** showing a bar graph of the selected feature and season, categorised by the selected championship. For example, if the user selects the points gained in the drivers' championship of the 2019 season, the graph will return Figure 2.

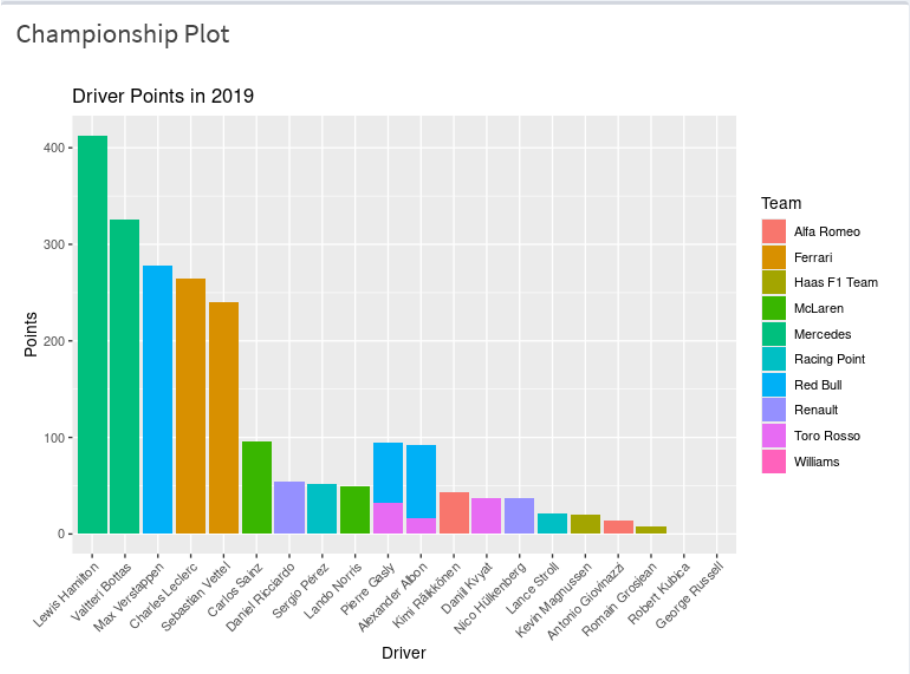

Figure 2: Plot of points by each driver in the 2007 season.

Additionally, if the user selects the drivers' championship, the graph is coloured by the team that driver scored points for, as this allows for better analysis between teammates and also shows drivers switching teams mid-season *(See Pierre Gasly and Alexander Albon above)*.

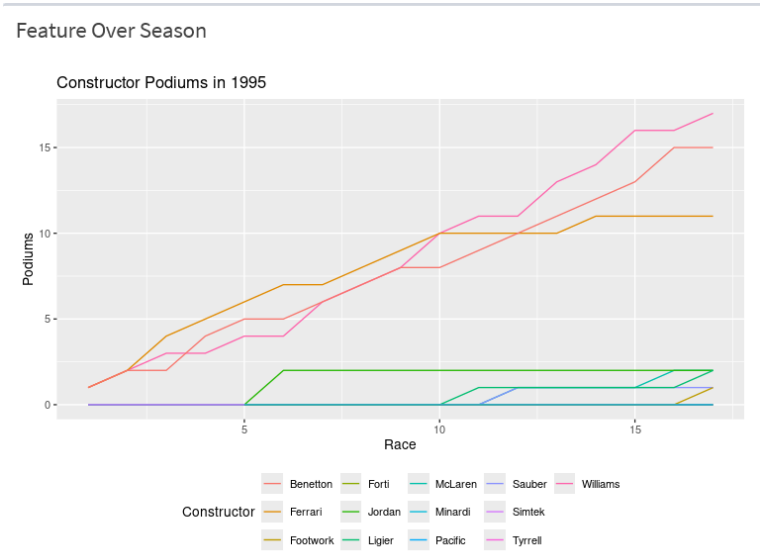

Figure 3: Plot of podiums for each constructor in the 1995 season.

The second graph created shows a line graph of the selected feature for the selected championship, across the selected season. For example, if the user wants to analyse the podiums for the constructor's championship of the 1995 season, the graph returns as seen in Figure 3. This can be used to analyse changes in standings across the season.

Finally, a line chart detailing the placings of each entrant in the selected championship is plotted across the selected season. This plots the changes in leaderboard positions across the season. This can be seen in Figure 6.

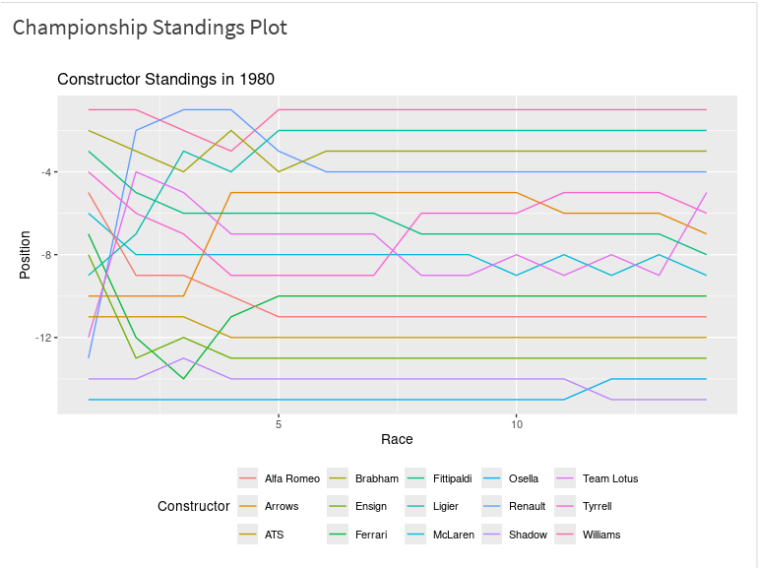

Figure 4: Plot of constructor standings across the 1980 season

3.2 Driver Tab

The driver tab allows for analysis of a specific driver. This page has one input, a search bar for drivers which can be used by first or last name. If multiple entries are found, as can happen with common names, the most recent driver is selected. Initially this was a drop down selector, but I found that to be far too cumbersome, and so elected for a search bar instead. Driver details are also printed to confirm which driver has been found by the search.

Figure 5: Search bar displaying details for Alex Albon

This is used to plot 3 graphs, beginning with a bar chart plotting the points scored by that driver in each season.

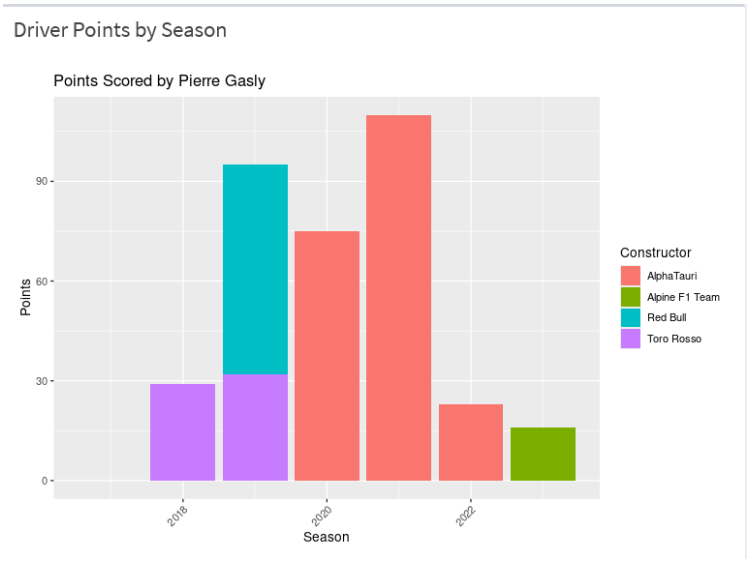

Figure 6: Plot of points scored by Pierre Gasly across his career.

This allows for a broad overview of driver performance in each season, as well as comparisons between their time spent at each team.

Next, a pie chart is plotted showing the distributions of driver status at race end over their career *(i.e. How often does this driver crash, have engine issues, etc)*.

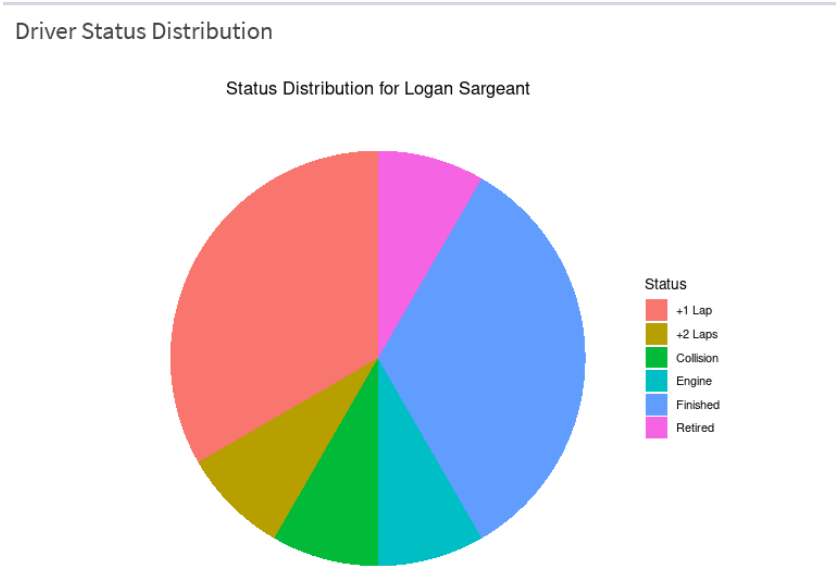

Figure 7: Pie chart showing distribution of race ending status for Logan Sargeant.

Finally, a line chart is plotted detailing the average placing that driver achieved for each season they competed in.

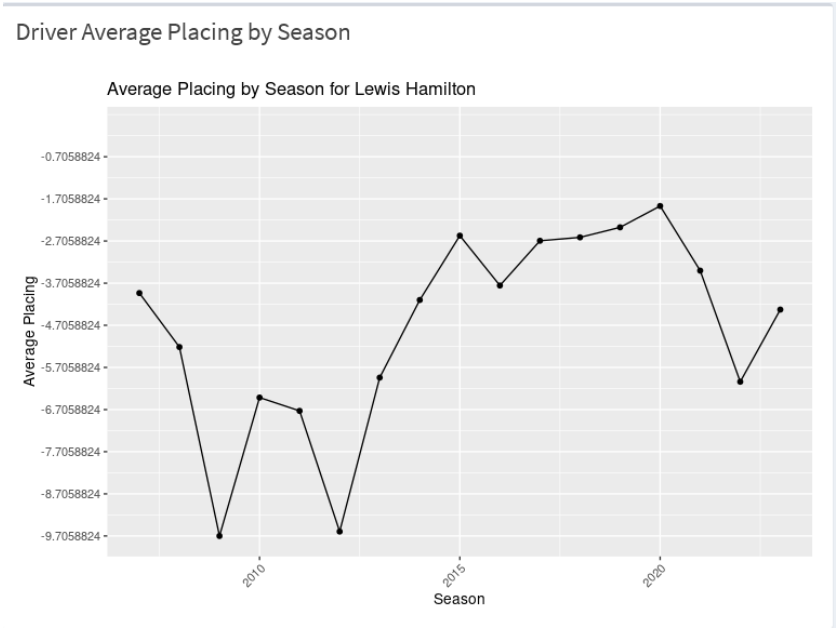

Figure 8: Line chart showing average placing in each race (From 1st).

3.3 Constructor Tab

The constructor tab shares a similar purpose to the driver tab, but instead focuses on the analysis of specific constructors. As such, the same search functionality is implemented to allow users to search for teams.

This produces two graphs: First, a bar plot of the total points scored in each season *(Coloured by the driver that scored them)*, and second, a pie chart displaying the proportion of total points that each driver scored.

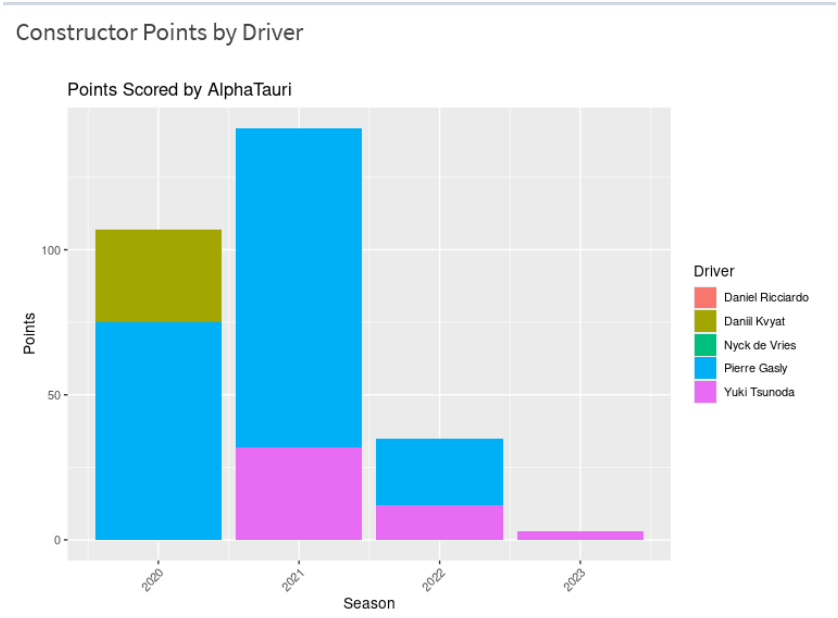

Figure 9: Points totals for Alpha Tauri.

This graph shows the changes in performance and points across the seasons, as well as the contributions of each driver.

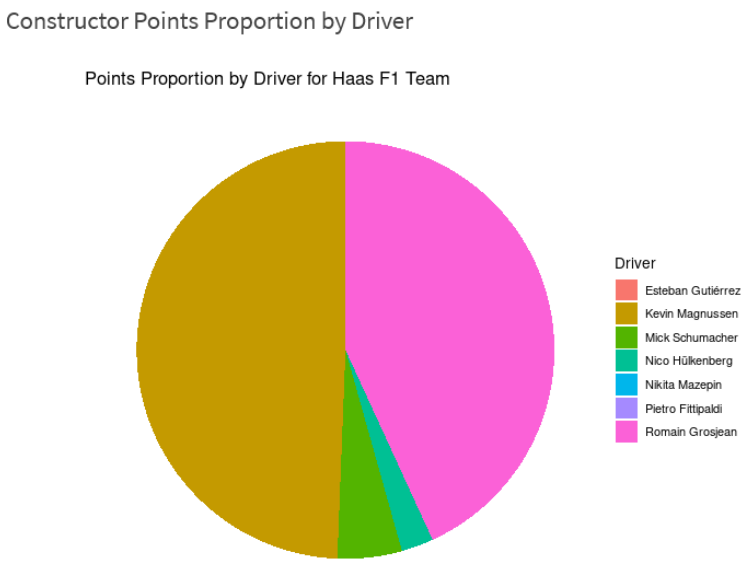

Figure 10: Driver points proportions for Haas F1.

4. Insights

Several areas of this project presented challenges throughout development. Foremost was my initial lack of experience with R and particularly with Shiny. While I have some understanding of R programming, this project was my first attempt at using the language for developing an app.

Particularly difficult for me was the data management aspects, mostly contained in the server file, as this was the area I was least knowledgeable in. This was compounded by a lack of specificity in logs and tracebacks whenever there was an error relating to Shiny, leading to some frustrations. Often these logs would provide such minimal detail into what the issue was that I was left without a clear path to fix the code for several hours.

In contrast, the UI development in Shiny is very similar to more typical web app frameworks, which made this aspect relatively easy. Despite this, the project itself provided a good opportunity to develop skills with these tools, and I soon became more capable of identifying errored code.

4.1 Comparison to Initial Goals

The initial project proposal detailed a similar, but not identical approach. Specifically, the project proposal detailed only including the season and driver tabs, but not the constructor tab that was also actualised. In this regard, the final outcome of the project exceeds the initial proposal. However, the proposal also details the driver page containing more options relating to comparisons between several drivers, which was not implemented. Overall, I would consider this project mostly on-target as compared to the proposal, but with some minor differences.

4.2 Future Changes

While the app is fully functional and does not contain any critical errors *(as far as I know from personal testing)*, there are still several areas I would like to make changes to in the future.

Foremost, I significantly underestimated the number of drivers and teams in early seasons of F1. Instead of the consistent modern field of ~20 drivers across ~10 teams, many of these early seasons reach ~30-40 drivers across ~15-20 teams. This causes graphs plotting seasons around the 1960's to be much harder to read and interpret. A solution to this would be to allow the limiting of input data with filters. For example, graphs could be limited to only plot the top X drivers, where the user could pick X. I attempted to implement this but found some difficulties and could not achieve this in time.

Similarly, the pie chart on the driver tab often displays too many different statuses to be easily readable. It would be better to group similar statuses together *(e.g. Power loss with power unit, which are essentially the same)*.

5. Conclusion

This project, while not without some issues, has been successful in its aims. An interactive web application was developed and deployed that allows users to perform custom analysis on various aspects of F1. This application was tested, and while some problem areas were identified for future work, these do not significantly impact the usability of the final outcome.