FormulaTour

Gianluca Capozzi
"La Sapienza" University of Rome
capozzi.1693255@studenti.uniroma1.it

Marco Costa
"La Sapienza" University of Rome
costa.1691388@studenti.uniroma1.it

Figure 1: Framework Views: (a) Home view, with world interactive map, winners of the season and interactive drivers trend; (b) Circuit view, with drivers standing and final ranking of the race; (c) General info view, with best driver and constructor, most successfull drivers and constructors; (d) Correlations view, with scatter plot showing correlation between driver's or constructor's nationality and wins; (e) Circuit info view, with scatter plot showing time trend in the years for each circuit; (f) Menu

Abstract

The purpose of our application is to provide a framework able to graphically display data about Formula 1 through a user-friendly interface. It's possible to navigate through different views in order to get visual informations about this sport. The application allows you to analyze the history of Formula 1 from the first World Championship (1950) to the last (currently 2019).

1. Introduction

Formula 1 has a very long history, as it has reached the 70th season. During all these years many drivers and constructors took part to the races, which took place in different circuits located in different countries. This means that it is very difficult to navigate this history, since it is characterized by many components, each related to the other. The goal of this project is to provide a way to navigate the history of the Formula 1 by highlighting each component and their relations. Our framework is divided into four main views each of which contains multiple sub-views. An initial view summarizes the races of the selected season by showing the countries and the exact location of the circuit in which they took place. A general view summarizes the most important results achieved by the drivers and the constructors during the Formula 1 history. A correlation view that shows if it exists a correlation between nationalities and number of victories/podiums. The last view is a times update views that summarizes the evolution of the lap times on a selected circuit during the years. So, our project is composed by:

- A visual analytics framework that supports the examination and exploration of Formula 1 history;

- A visual representation of the features selected by the users.

The project is a web application and it is developed using the d3.js [1] library and the python3 language for performing PCA.

2. Related works

There are other frameworks that show the evolution of Formula 1 through its different elements, so races, drivers and constructors. An example is the History of F1 [2] framework, based on the same dataset used for this project. The main difference is that in that project there is not a division of the races with respect to the years and also the user is not able to locate geographically the position of the circuits.

3. Dataset

The dataset used for this project is obtained from the Ergast Developer API [3] in .csv format. It is composed by 13 tables: circuits, constructorResults, constructorStandings, constructors, driverStandings, drivers, lapTimes, pitStops, qualifying, races, results, seasons and status. The tables considered for this project are:

- races: a table that contains informations about each race, as the year and the date in which it was played, the circuitId (which identifies the circuit in which it took place), the name of the Grand Prix (such as Italian Grand Prix) and so on;

- circuits: a table that contains informations about each circuit, such as the name of the circuit, its coordinates, the country and so on;

- drivers: a table that contains informations about each driver, such as his name and surname, his date of birth, his nationality and so on;

- results; a table that, for each race and driver who took part in that race, shows the result of that driver in that race and other infos;

- driverStandings: a table that, for each race and driver who took part in that race, shows the driver's point into the drivers' standing after the conclusion of that race;

- qualifying: a table that, for each race and driver who took part in that race, shows the results that the given driver achieves during the qualifying for that race;

- constructors;

- constructorStandings.

The .csv files are manipulated using both d3.js and python. Some of the queries performed over the dataset needs the interaction of the user, while others are performed without parameters chosen by the users.

4. Design Overview

The visualization is composed by 4 main views, accessible from a simple side menu. Each of these views provides some processed information about the dataset.
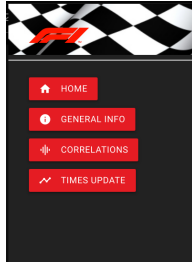
Figure 2: Menu

In the first view the statistics of a season are analyzed, selectable by the user. The second view provides general information for all seasons, highlighting the best drivers and constructors. In the third view the correlation between the nationality of the drivers or constructors and the victories is shown. The last view shows the trend of the best times over the years.
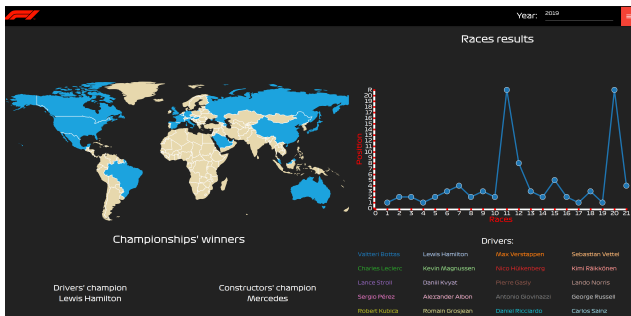
4.1. Home view

Figure 3: Home view

This is the main view. Here the user is able to select the desired year from the special selector: the available years are from 1950 to 2019, so from the year in which there was the first F1 season till the year in which there was the last one. The home view is divided into four sub-views:

- a map that highlights the countries in which there was at least one Grand Prix during the selected year;

- a scatter plot that shows, for each driver, his final position for each race of the selected season;

- a sub-view that shows the driver and the constructor that have won the two championships at the end of the selected season;

- a sub-view that allows the users to selected a driver: by clicking on the name the scatter plot will be modified showing the results of the selected driver.

The map is interactive, indeed the user can click on one of the highlighted countries (a country is colored in blue if during the selected year there was at least one Grand Prix played in that country): at this point there will be a zoom on the selected country and a red circle showing the position of the circuit in which the race was played. By clicking on the red circle the following view will be shown to the user:
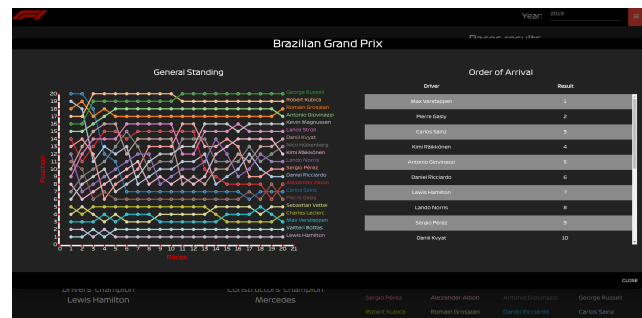

Figure 4: Race infos

On the left there is a scatter plot that shows the evolution of the drivers' standing till the selected GP, on the right there is a table showing the order of arrival of the selected race.
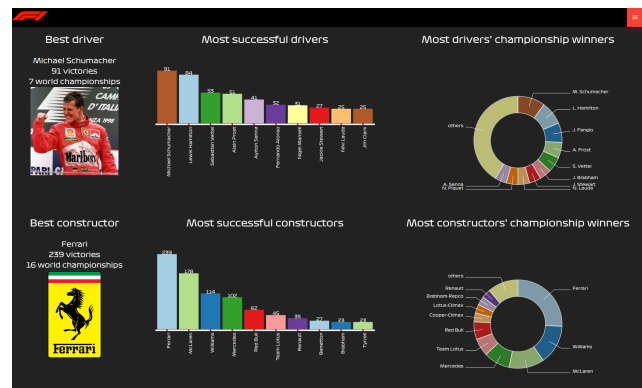
4.2. General info view

Figure 5: General info view

In this view we show some statistics about the general world of the Formula 1. It is a grid view divided in two rows and three columns: the first row highlights the drivers while the second one the constructors. The

three elements of the first row describe the following statistics:

- Best Driver: this is the driver that has won more races that the other drivers. Here we show the number of victories of this driver and the number of drivers' championship that he has won;

- Most successful drivers: this is a top 10 based on the number of races won by a driver and it is presented using a bar chart. Each bar is associated to a driver and represents the number of races won by that driver; on top of the bar there is a number representing this statistic;

- Most drivers' championship winners: this is a top 10 based on the number of drivers' world titles won by a driver (we have added also the number of championships won by all the other drivers) and it is presented using a donut chart: each slice is associated to a driver and by sliding the mouse over the slice it is possible to see the number of championships won by that driver.

The two charts presented in this first row are linked together using of a double interaction: by clicking on a bar of the first chart it will be highlighted together with the slice of the other chart corresponding to the same driver (if it exists). The same happens by clicking on a slice of the donut chart.
As said before, in the second row there are some statistics about the constructors:

- Best Constructor: this is the constructor that has won more races than the other constructors. Together with its name we show also the number of race victories and the number of constructors' world titles that it has won;

- Most successful constructors: this chart shows the 10 constructors that have won more races. It is a bar chart where each bar refers to a constructor with on top of it the corresponding number of victories;

- Most constructors' championship winners: this chart shows the 10 constructors that have won more world titles: it is a donut chart where each slice is associated to a driver (we have added also a slice representing the number of world titles

won by the other constructors); by sliding the mouse over the slice it is possible to see the number of championships won by that constructors.

Also in this case there is a double interaction between the two charts, similar to the one present for the other two charts presented before.
In both the bar charts by sliding the mouse over a bar it is possible to have some informations:

- drivers: date of birth, nationality, teams for which it has driven, number of races that it has started and number of podiums

- constructors: nationality, number of started races and number of podiums

In both the rows there is a consistency between the colors of the bar and the slice associated to the same driver/constructor.
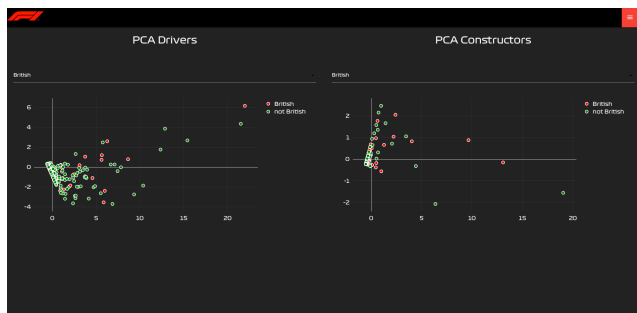
4.3. Correlations view

Figure 6: Correlations view

This view is designed to show if there is a correlation between a specific nationality (selected) by the user and some statistics about a driver or a constructor of that nationality. The view is divided into two subviews: one for the drivers and one for the constructors. The user is able to select two different nationalities for the drivers and constructors using two different selectors. The result is plotted in the form of a scatter plot. In this two scatter plots the red point represents the elements of the nationality selected by the user while the green points represent the elements belonging to all the other nationalities.
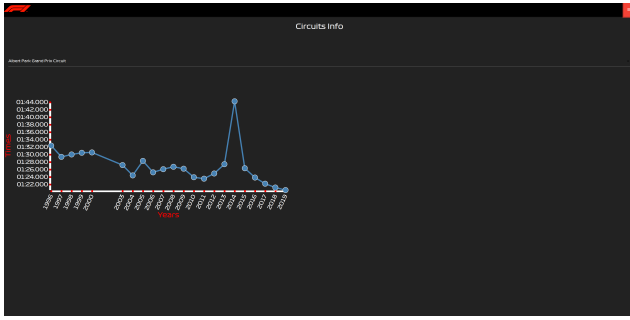
4.4. Times update view

Figure 7: Times update view

This last view shows the evolution of the lap times during the years for each circuit. The user can select one of the proposed circuits from the selector and then the scatter plot will be updated. The lap times that are considered for creating this view are the pole position times for each Grand Prix over the years, so the fastest time recorded during the qualifyings. Further informations about the selection of this lap time are provided in the Analytics section. On the x axis there are the years for which the best lap time is available while the y-axis shows the times at intervals of different frequencies (the frequency depends on the number of different lap times for each year that we have for each circuit).

5. Analytics

The views are produced by manipulating the dataset in different ways starting from an input given by the user. This manipulation is done using the d3.js [1] library, since it allows to do operations easily on .csv files and using Python 3 (in particular the pandas [5] and scikit-learn libraries [6]) for analyze the correlation between nationality and number of wins/podiums. Most of the operations are characterized by a `JOIN` operation between the tables of the dataset and a subsequent `SELECT` operation over the desired attributes with a `GROUP BY`.

An example could be the map in the main view: in this case we are interested in the circuits where there was at least one race during the year selected by the user. So, first of all we make a `JOIN` between the two tables races and circuits over the attribute `circuitId` `WHERE` the year in which the race was ran is equal to the one selected by the user and then we `SELECT` the name of the circuits together with the corresponding country name and the name of the races. In SQL this query is like the following:

- `SELECT c.name, c.country, r.name`

- `FROM circuits c, races r`

- `WHERE c.circuitId == r.circuitId AND r.year == y`

where `y` is the year selected by the user. At this point we are interested in two main values: the country in which the race was ran and, in order to put the red circle on the map, the coordinates of the circuit. In order to highlight in blue the country on the map, for each country we check if its name is into the set of country names resulting from the query described before. To add the red point at the coordinates of the circuit we check, for each circuit, if the `circuitId` attribute is into the set of ids resulting from the previous query and then we take the corresponding coordinates.

The Correlations View is based, as the name suggests, on the concept of correlation between attributes. In order to find if there is this correlation between attributes we have decided to use the PCA algorithm, that is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of summary indices that can be more easily visualized and analyzed. In our case we have used PCA to check if there is a correlation between a nationality (selected by the user) and a set of features: the number of victories, the number of podiums, the number of attendances and the number of pole positions done by a driver/constructor. The computation of PCA is done using Python 3 and in particular using the implementation provided by the scikit-learn library [6]. The idea is the following: if `n` is the nationality selected by the user then we create a new dataset in which we divide the drivers/constructors with respect to their nationality using the following binary classification: `n` vs non-`n`. Each row of this new dataset is composed by the following fields: number of victories, number of podiums, number of attendances and number of pole positions. Once done this we compute PCA over this dataset performing the following three steps:

1. Separate the labels (`n` and non-`n`) from the features

2. Standardize the features (such that the variance of the data is 1 while their mean is 0)

3. Perform the dimensionality reduction using the PCA algorithm (the data are projected on a 2D space)

Since the user can select different nationalities through a selector, we decided to deploy the script that runs PCA on Heroku[7], creating a simple server with Flask[8]. The resulting dataset is plotted using d3.js [1].

As said in the previous section, the Times Update View is characterized by the evolution of the best lap times for each circuit selected by the user. A key point of this view is how to compute this best lap times. Since during Formula 1 history in addition to technology changes there have also been changes in the race regulations (such as changes applied to pit stop rules) we have decided to take as best time for each circuit and each year the time of the pole position, so the fastest lap time registered during the qualifying of the race. In the current regulation, the qualifying is divided into three phases: Q1, Q2 and Q3; the pole position is the fastest lap of the Q3 phase. But, by exploring the dataset, we have found that there are some qualifyings for which one of the phases is not available (or no one of the phases; in this case obviously the race is not considered) so the fastest lap time is chosen with the following criterion: if Q3 is not available the fastest lap time is the one registered during Q2; if Q2 is not available the fastest lap time is the one registered during the Q1 phase.

6. Future Works

As the dataset provides a lot of data, it's possible to add more views to the application. We have in mind to add an interface that allows you to compare the statistics of the various drivers (e.g. number of wins, number of podiums, number of poles, etc.). The same could be done for constructors. Another interesting thing we would like to do is to have a projection of the remaining part of the current season considering both the races already raced in that season and the results obtained by the various drivers in previous seasons.

7. Conclusion

The project is useful to be able to retrace the history of Formula 1 easily, through intuitive graphics and a "sports" interface. The application could be used for different purposes, for example by analyzing the times over the years it is possible to guess how the technology or the skill of the pilots has progressed. Learning how to use the framework is very simple, as we have chosen to use immediate and easily interpretable graphics.

References

[1] Data-Driven Documents: https://d3js.org/

[2] The history of Formula 1: https://f1.bitmetric.nl/formula.html

[3] Ergast Developer API: http://ergast.com/mrd/

[4] Plotly JavaScript Open Source Graphing Library: https://plot.ly/javascript/

[5] Pandas library: https://pandas.pydata.org/

[6] Scikit-learn library: https://scikit-learn.org/stable/

[7] Heroku: https://www.heroku.com/

[8] Flask: https://flask.palletsprojects.com/en/1.1.x/