Visualisation of Formula One Racing Results

Giuseppe Callari
Katholieke Universiteit Leuven
giuseppe.callari@student.kuleuven.be

Michael Vincken
Katholieke Universiteit Leuven
michael.vincken@student.kuleuven.be

Stefan Pante
Katholieke Universiteit Leuven
stefan.pante@student.kuleuven.be

ABSTRACT
This paper describes the design and implementation of a visualisation of Formula One racing results. The user can interact with the visualisation using a draggable timeline. We use coloured bar charts for a visual representation of the strengths and weaknesses of specific drivers or constructors over the years. Moreover, our visualisation gives the user the opportunity to compare two drivers based on their finishing positions. Through this paper, we will shortly discuss the problems we have encountered and lessons we have learned in the process of making this visualisation.

Author Keywords
Information visualisation; Formula One; Race results; Motor sport; Seasonal wins; Drivers; Constructors;

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
The FIA Formula One World Championship, better known as Formula One or F1, is one of the most popular motor sports in the world. The F1 season consists of a series of races, also called Grands Prix, held throughout the world. Each season consists of a certain number of races. During a season, a number of teams or constructors (we will use these names interchangeable in this paper) compete with each other. Each team consists of two drivers and some additional secondary drivers. It is important to note that a team may only use two cars during a race. After each race, drivers receive points depending on their finishing position. In the current system, the top ten driver-car combinations are awarded points. Based on the racing result of their drivers, the team with the highest cumulative score wins the constructors title. At the end of the season, the driver with the highest individual score wins the World Championship title.

There already exist visualisations for consulting the statistics of Formula One racing results[1][2]. Most of these visualisations give the user an overview of results per season. The specific focus of our visualisation is that we want to make it easy to compare driver (or constructor) results over several years.

The goal of this visualisation is to present an overview of seasonal race results for specific drivers throughout their active years, while keeping an overview of the opponents. This visualisation is interesting for motor sport fanatics or those who want to know more about driver performance.

In this paper, we first give some background information about Formula 1 racing statistics. Next, we describe the data set we used for our visualisation. Finally, we discuss some related work.

FORMULA ONE RACING STATISTICS
As stated previously, a Formula One racing team consists of several drivers. In general, two drivers are the main drivers of the team. The other drivers are test drivers or can take the place of one of the two main drivers in case one is not able to participate in a race (due to medical conditions or a disqualification). Each driver can earn points in a race when finishing in the top 10. In the current point system, the winner gets 25 points, the second 18 points, third 15 and so on. It is import to note that the scoring system has been changed several times over the years. This complicates comparing drivers or constructors over the years. That is why we used the number of wins as well as seconds and thirds for both drivers and constructors to overcome this problem.

DATA SET
For our visualisation, we made use of two data sets, namely Ergast API and Formula Racing Data Set. The Ergast API provides data for the Formula One and Formula E serie from the beginning of the world championships in 1950 and 2015 respectively. At first, we wanted to make a visualisation with detailed information including the duration of pit stops and lap times. A problem we encountered was that the Ergast API only provides this kind of data starting from 2008. To overcome this problem we shifted our focus on driver and constructors performances through the years.

OUR VISUALISATION
Our visualisation as shown in Figure XXX consists of two parts: the time line is the main part of our visualisation, where we provide an overview of seasonal wins for the different drivers and constructors. The second part, situated under the timeline, focuses on two specific drivers, so as to enable a comparison of their performance.

Time line
The time line contains the active years of the selected driver. For each year, we plot the number of wins using bar charts.

Each individual bar represents a constructor who competed for the title in that specific year. Figure 1 illustrates that a bar is divided in two colours, namely blue and grey. The blue bar, the bottom part, represents the number of wins of the driver who performed the best for that team. More specifically, this bar represents the driver who has earned the most points and wins during the season. The grey bar, the top part of the bar, represents the number of wins of his opponent from the same team. These two bars are stacked on top of each other. In this way, we visualise the individual wins of the drivers, but also the number of wins for a constructor.

Another reason why we visualised the scores like this is to enable interaction with the user. First the user selects a driver in the bottom part of the visualisation, e.g. Vettel. The lower part of the visualisation is now updated with the data of Sebastian Vettel. When the user now hovers over part of a bar in the upper part (as discussed above), a tip pops up and displays the name of the corresponding driver, as well as his number of wins. By clicking on that part of the bar, the data of the corresponding driver is loaded and the visualisation in the lower part is updated.

Navigator

The draggable slider and the miniature version of the timeline are used to navigate throughout the years. This miniature version shows the time line as a whole, from the beginning to the end without sideways scrolling. This provides an overview for the user. The miniature version has not the goal to already provide all the information to the user, but it rather is an incentive for the user to explore certain periods. It could for example trigger the user to drag the slider to the start of some drivers career when exploring the end of it. The start could have same peaks which the user did not remark on the selected period of the main timeline. The miniature version only shows the bars of the non selected drivers and selected drivers in the corresponding colors as the main timeline. The bars are not navigable in the navigator and are not hoverable either. This in order to reduce interference when navigating throughout the years. A miss click when selecting a new year would trigger the change driver scenario which may be unwanted.

Comparing two drivers

The user can compare two drivers based on their finishing positions. In order to get an overview, the user clicks on part of a bar, as described above. Now the number of times the driver has finished first, second and third pops up. When the user clicks on another part of a bar, the same statistics are placed next to the statistics of the first driver. In this way, one can simply compare the performances of the two drivers. When the users clicks another part of a bar, the driver that was selected first will disappear and be replaced by the last selected.

An alternative functionality considered is the following: hovering over a bar lights up all bars corresponding to the same driver. It makes the user clear which driver he/she will select. The user gets an overview in what teams the hovered driver has participated. Not all lit up bars are label shown with their label, but this view gives the user an idea in what kind of teams the driver participated and if he was dominant in these teams. For more details, the driver must be selected.

Statistics

Below the main timeline and the navigator, another line is visualised in parallel. This line is divided in year blocks similar to the main timeline. In every block, more information about the chosen drivers is shown. These blocks are preserved for analysis of the selected drivers only. Three metrics are compared: number of first places, number of second places and number of third places. These correspond to rankings for which a driver is allowed on the podium during the podium ceremony. During this ceremony, the best three drivers are celebrated after each race. The best racer, the one who won the race, receives its price. Nowadays, the second and third driver also receive something as souvenir for their achievement. These prices and souvenirs differ throughout the history of this motorsport. Podium finishes are considered an important statistic used in sports to analyse an athlete's career (or driver's career in motorsport). Achieving a podium finish puts the driver in the spotlights by honoring its achievement publicly. During the olympics, medals and thus podium places are an important metric as well. Countries are usually ranked by the number of medals achieved by their athletes (Referenties). MotoGP, the premier class of GP motorcycle racing, (Referentie) provides a search functionality on its official website in order to let user explore their database of races, drivers and results. "Podium with riders of the same nation" is a search space users can explore.

For each place (first, second and third), the number achieved is visualised using circles. The number of circles indicate the number achieved for a place. The color of a circle corresponds to the driver. Each place has two horizontal rows of circles, in which each driver corresponds to a row. This design decision facilitates comparison of two drivers for these these metrics. (Grouping, color, discrete values etc referenties). Furthermore, in this block, the names of the teams the drivers made part of are shown as well. Place and color of the constructor in the statistics block indicate who is the corresponding driver. The position in the block corresponds to the position of the name of the driver shown in the left upside corner of the visualisation. Events like team switches or temporary stops are visualised this way. These events are thus implicitly visualised, instead of explicitly show the user where a selected driver switched constructor. Next to each team, a medal is shown to indicate the final ranking for the corresponding year. A first place and thus championship win is decorated with a gold medal, a second place is decorated with a silver medal, a third place is decorated with a bronze medal and a place lower than three is decorated with a dark blue medal. In each medal, the number corresponding with the place is shown. We believe the podium metrics provide context for the results (points or wins) shown on the main time concerning the selected drivers. We already mentioned in section [ref] which discusses the dataset that a significant amount of drivers have gathered points in a year, but did not achieve a single victory. Let us consider drivers who achieved no or very few (one or two) wins. These drivers however

could have achieved many podium places during a year. Second and third places have always be rewarded by points (voorlopig wikipedia referentie). This could trigger the user to switch to the points view for the main time line to further investigate the selected drivers.

Points vs wins [MSS TOCH EVENTJE VOOR CHANGE IN POINTSYSTEM TOEVOEGEN?] Note that point- systems are regularly changed. Therefore, this metric is not ideal to compare drivers in the main time line which competed in different periods. The amount of wins however is a metric which does not change in value. We believe that the statistics provide the context to interpret results of drivers and that they can be an incentive to change the view, as discussed in statistics.

IMPLEMENTATION

The implementation of the visualisation consists of five parts: data gathering, the slider, the timeline, the statistics and presentation.

Data gathering

We used the Ergast F1 API to gather our data. We originally planned to connect directly to the API to retrieve the necessary data. It quickly became apparent that a direct connection was not suitable because of the structure of the API. Because our visualisation aggregates a lot of data, too many requests would be required, which would have resulted in very poor performance, numerous time-outs and overall, a worsened user experience. After downloading a database dump, the reasons for the bad performance were immediately clear. The relational database was heavily normalised, requiring multiple joins for each API call we made. We opted to transform the data to a JSON-object representation, which is downloaded in its entirety every time a user accesses the visualisation. We choose this approach instead of using a backend NoSQL database because we strongly believe such a database would not provide additional benefits to our users and we wanted to adhere to the KISS principle.

We developed a Node.js and python script to collect our data. These scripts automate the collection and transformation of our data. The drawback is that the scripts should be run each week to keep the data up-to-date, a process which also could be easily automated on a server.

Navigator

The slider is implemented without using a specialised library. Plain JavaScript in combination with jQuery in order to manipulate CSS, was sufficient to code the slider. The miniature timeline, which shows all the bar charts (selected and not selected drivers of all the years) has not all the functionality the main timeline provides: the stacked bars are not clickable and thus navigable, they are not hoverable and there are no trendlines for the selected drivers visualising the up and downs of their careers. The aforementioned functionalities are not only redundant for the mini timeline, but interfering too. Clicking on a year in the mini timeline redirects the slider to the clicked year in order to visualise the corresponding period. An unintentional click on the bars would trigger the switching scenario. Therefore, the code is not the same as the code for the main timeline. Making the code reusable, and thus avoiding code duplication, resulted in conflicts between svg-elements which were easily fixed by just duplicating the d3 code for the timeline. After duplication, refactoring followed in order to scale and place the miniature timeline under the slider and in order to remove redundant functionality.

Timeline

The timeline which holds all the bar charts is the backbone of our visualisation and its functionality is completely implemented using the d3.js framework. First, the data concerning the not selected drivers is visualised: these are represented by the grey stacked bar charts. Subsequently, the selected drivers are plotted, in a different color. Note that for a stacked bar with no selected driver involved, the lower bar represents the best driver in a team and the upper one the one who collected the least number of championship points. We believe that designing the non selected drivers in this way results in a more consistent visualisation. Making the stack order random would be a confusing factor and could interfere with our goal of visualising information in a clear way. This decision however has a consequence for our gathered data. The data concerning the selected drivers and all drivers who competed in the same years is processed and manipulated before it is visualised: the drivers of a team are sorted by the number of wins for each year. This is not done in the data gathering step, but processing the data after gathering them takes care of this sorting step.

After this step, the selected drivers are visualised on the timeline. The bars representing the selected drivers have a different color than the non-selected ones and are always shown as the bottom bar in a team, even if a non selected team-mate had most wins in the team. The performances based on the criterium of wins are represented by the height of the bars corresponding to the drivers. When two drivers are selected, the visualisation puts these two in focus by giving their bars a different color. It's tolerable if two not selected drivers cannot be easily compared because one is a non-dominant driver and thus the highest bar on a stacked bar. The bars of drivers in focus however must start at the same height (the bottom) to compare their heights. Starting at the bottom has the consequence that the performance can be read on the provided vertical axis.

A second consequence is that the line charts complement the bars of the selected drivers. The lines hit the top of the bars of the selected corresponding drivers.

To show which bar corresponds to which driver, we implemented tooltips using the library d3-tips http://bl.ocks.org/Caged/6476579 When hovering over a bar, the bar lights up and the name of the corresponding driver is shown together with the amount of wins and the team to which it belongs.

Statistics

The process of creating and displaying the statistics underneath the timeline is handled by the EJS templating engine. Instead of having to build each statistic by appending strings, the templating engine automates this process, which allows

the developer to separate HTML from javascript. We considered using handlebars [...] or EJS [...]. We decided to use EJS because it provides more flexibility than handlebars by allowing complex javascript statements inside templates.

RELATED WORK

For related work, we focus on visualisations that represent an overview of results from team sports, e.g. basketball, football, soccer, cycling and so on. The FIFA Ranking For The Gulf National Teams[1] by Marcelod uses a hemisphere to represent a time line with all the years in which a championship was organised. The y-axis represents the position of the soccer teams at the end of a season. This is an interesting way for visualising scores throughout the years. Overall, we think this visualisation is a bit bombastic. Marcelod used data from 1993 until 2014 to visualise 20 years of soccer championships. Our data ranges from 1960 until 2015. Therefore , we opted that its clearer to use a draggable time line instead of a hemisphere. The user can interact with our time line by using the slider. In addition, Marcelo also uses ”events” - these are the long vertical blue lines representing the various world cups - to divide the visualisation into regions. [This is a feature that we have also implemented in our visualisation, to indicate the year where a pilot has changed teams. [Office20] [MV21]]

Another interesting visualisation is the Formula 1 Lap Charts by Davidor, which also relies on the Ergast web service for the dataset. This chart visualises the performance of the Formula 1 pilots in the first Grand Prix of the year. The x-axis includes a vertical line for each lap of a specific race. When we expand this visualization for one full season instead of one race we can clearly see the fluctuations in the scores of the pilots. By following one line you can see the switches in position for that driver. These fluctuations are very interesting to get an overall view of the performances of certain drivers. We used this kind of visualisation as well for our trend line .