

# Selected Topics in Information Visualization - Project

---

## Interesting Formula 1 Statistics

Made by: Máté Vass (76220125)

GitHub repository: [Check out the repository!](#)

Deployed version: [Check out the application!](#)

Demo Video: TODO

---

## Description

Formula 1 is the highest class international open-wheel single-seater formula racing series in the world, with hundreds of millions of followers worldwide. Every year, unbelievably large amounts of money is spent by the constructors to develop the fastest car possible in order to attract wealthy sponsors and create great advertisement for the brand itself. Winning a season in Formula 1 is considered to be the highest achievement in the world of motorsports. Following the sport is very easy due to its popularity. However, only a few tools are publicly available for free to analyze results and statistics at a deeper level. I found a dataset on Kaggle representing Formula 1 results, statistics and metadata from the first season in 1950 until this year, 2023. I follow a few social media pages that showcase some really interesting facts and statistics about F1 race after race. I was partially inspired by them, and I think it would be very exciting to create various visualizations from the dataset mentioned above and provide a free tool for the hardcore fans.

## Dataset

**Formula 1 World Championship (1950 - 2023)**

Data Card    Code (74)    Discussion (34)    1231    New Notebook    Download (6 MB)    :

**circuits.csv** (10.1 kB)

Detail    Compact    Column    9 of 9 columns

**About this file**

Circuits where F1 races are held

circuitId	circuitRef	name	location	country
1	albert_park	Albert Park Grand Prix Circuit	California Barcelona Other (73)	USA France 95% Other (59)
2	sepang	Sepang International Circuit	Melbourne	Australia

**Data Explorer**  
Version 21 (20.14 MB)

- circuits.csv
- constructor\_results.csv
- constructor\_standings.csv
- constructors.csv
- driver\_standings.csv
- drivers.csv
- lap\_times.csv
- pit\_stops.csv
- qualifying.csv
- races.csv
- results.csv
- seasons.csv
- sprint\_results.csv
- status.csv

**Summary**

- 14 files
- 120 columns

[Check out the dataset!](#)

## Tasks

In this section, I'm going to describe a few tasks that a user can perform using the visualization tool. These tasks are described using the "Why" dimension of Munzner's taxonomy.

- Explore: with different views, the users can explore various aspects of Formula 1. For example, they can get to know some important drivers and constructors that were dominant in certain periods of time in the past years.
- Browse: after acquiring some basic knowledge, users can browse the dataset intentionally based on some attributes. In some views, for example, they can set certain variable values and narrow down their query to the desired subset.
- Compare: some views provide the ability to compare two plots at the same time. For example, it might be interesting to see which tracks are usually won from the first row, or which are ones where qualifying grid position matters less.
- Discover: the users can verify their different hypotheses in some of the views. For example, they can see that as technology is evolving, qualifying lap times are getting lower in the sport. However, other factors such as rain during qualifying result in spikes in the decreasing lap time trends.
- Enjoy: sometimes the only goal of visualizing some data is just pure joy without any deeper understanding of the topic. For example, in the view with the driver wins in the history of Formula 1, the users have the ability to start an animation. It loops through every race from the beginnings of the sport, and shows the users how different drivers in certain eras dominated the sport.
- Locate & Lookup: for more hardcore fans, there is a possibility to look for and locate certain facts with the visualization tool. They can even specify their desired time periods and tracks in the views and look up the exact information they are interested in.

## Tools used

The visualization tool is going to be built using:

- working with data: R
- plots: ggplot2 & plotly
- interaction, UI: R Shiny & plotly
- web application: shinyapps.io

The application is deployed using [shinyapps.io](https://shinyapps.io), for the purpose of the presentation and testing the trial with 25 hours of active time is enough.

## Views

In this section, there are short descriptions of the different views that the tool is going to support. In every one of the views, the user can modify the time period, this is one the temporal attribute in the application.

- View 1: Driver Wins

In this view, there is a treemap showing the most successful drivers in the history of Formula 1 based on their number of wins. Besides the time period, users can specify the minimum and maximum number of wins. This way, they can specifically look at "equally successful" drivers based on the number of races won.

- Views 2 & 3: Driver & Constructor Championships

These two views are kind of similar to the first one, but it is about driver and constructor championship titles. Another difference is that these are just a normal bar charts. There is the time period attribute, and also a minimum and maximum number of titles attribute. Here users can look at "equally successful" drivers and constructors based on the number of titles won.

- View 4: Best Lap Times

In this view, users can look at certain tracks and see how lap times have improved over the years. There is an ability to compare two different tracks, and look at the differences in characteristics. It can be interesting to think about the potential factors that lead to the results presented (such as unpredictable weather conditions, changes in the track layouts, etc).

- View 5: Wins From Grid Position

This view gives the users two pie charts, where they can compare two different tracks as well. Here they can see information about from which grid positions is it most likely to win a race in the selected track based on the historical statistics. On most of the tracks, more than half of the wins came from pole position, but there are some exceptions that are really interesting.

## Interaction

As mentioned earlier, the users can interact with the plots:

- with the sidebar on the left, they can adjust their settings and the changes will appear instantly on the right side
- some plots use *plotly* that provides a few basic interaction options as well
- the *year* attribute can be adjusted for every plot
- for each plot, there are other specific attributes that the users can modify:
  - View 1: minimum and maximum wins
  - Views 2 & 3: minimum and maximum titles
  - Views 4 & 5: two tracks to compare

## Design explanations

In this section, I'm going to explain my decisions regarding the design rationale of the final application.

The application has a sidebar on the left, where users can adjust their settings for the actual plot. On the right side, there are different tabs to select between the available plots. The available settings on the left change dynamically based on which plot is selected. This UI layout is self-explanatory, in European countries we read from left to right, so we will not miss any important information.

I chose the same color theme for every plot, this template is suitable for visually compared people, for example who suffer from color blindness.

I used a tab layout in the main plot view, this way only one view is shown. This helps users focus only on one task at once.

The views are linked in a way that if they have the same modifiable attribute, the changes appear on all of them. For example, if we want to look at driver wins between 2000 and 2010, then when we switch to the driver championships tab, it will already display them between 2000 and 2010. It makes navigation between plots easier.

I tried to chose different types of chart for different purposes. With the treemap in View 1, users can easily see the differences between relatively big values. The bar plots in Views 2 and 3 are easily understandable and self-explanatory. With View 4 we can try to understand certain trends in lap times and compare them between different tracks. With View 5, I tried to showcase the different rates from where drivers have won grand prixs at different tracks.

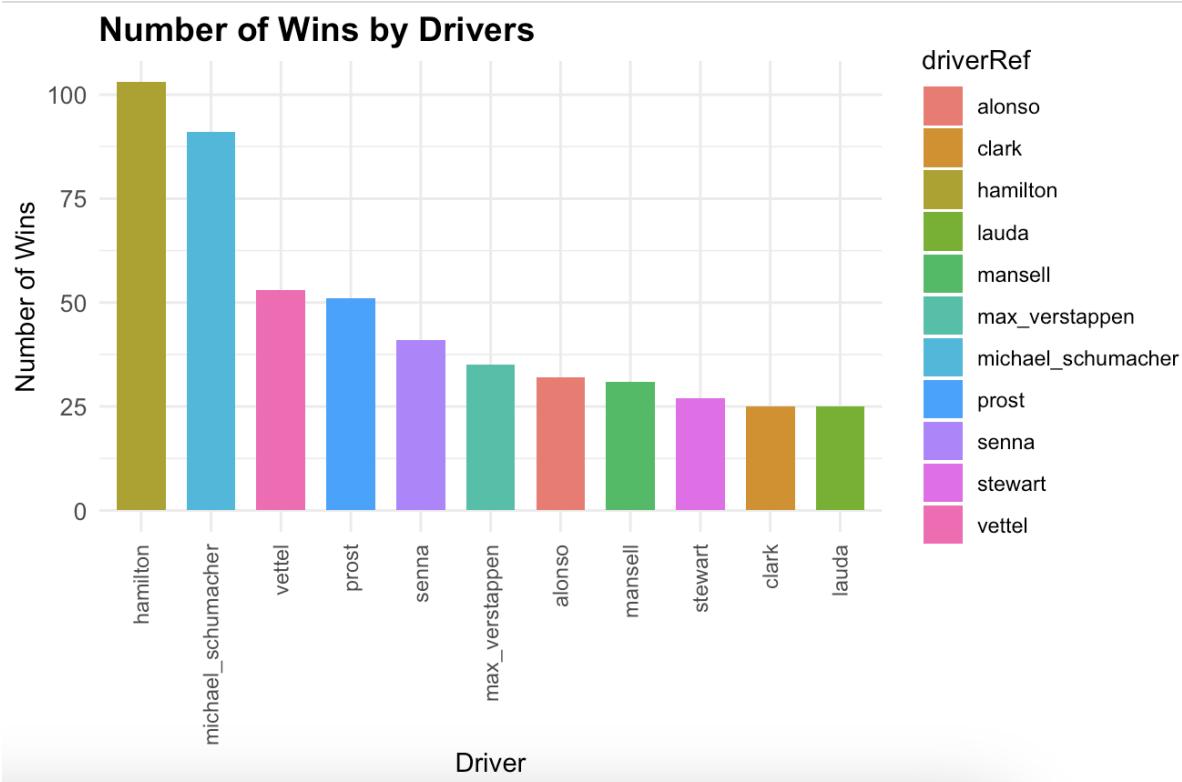
In conclusion, I tried to make an application which is self-explanatory for the average user. However, it provides some options for those who really want to dig deep into some interesting aspects of Formula 1 and derive meaningful inferences from the provided statistics. The tasks described earlier can be easily performed by anyone, even those which are a little more complicated. The design principles for the whole application were chosen based on these goals.

## Prototypes

The development of the code can be tracked in the GitHub repository provided. Every prototype was made in R Studio before the final version was deployed with *shinyapps.io*.

### Prototype 1 - Test plot

To start off, I downloaded the dataset from Kaggle and made a very basic plot to see if everything is working and I can start doing more complex visualizations. This prototype was already presented in the last lecture. Here, everything was hardcoded, and the UI was still missing.



Prototype 1 - Plot

The screenshot shows the RStudio interface. The left pane displays the R script 'formula\_1.R' with code for data manipulation and plotting. The right pane shows the 'Environment' tab with a list of objects and their details, and a plot titled 'Number of Wins by Drivers'. The plot has 'Driver' on the x-axis and 'Number of Wins' on the y-axis, ranging from 0 to 100. The legend lists drivers with their corresponding colors: hamilton (yellow), michael\_schumacher (blue), vettel (pink), prost (light blue), senna (purple), max\_verstappen (teal), alonso (red), mansell (green), stewart (orange), clark (brown), and lauda (dark green). The plot shows that Hamilton has the highest number of wins (approximately 100), followed by Michael Schumacher (around 90).

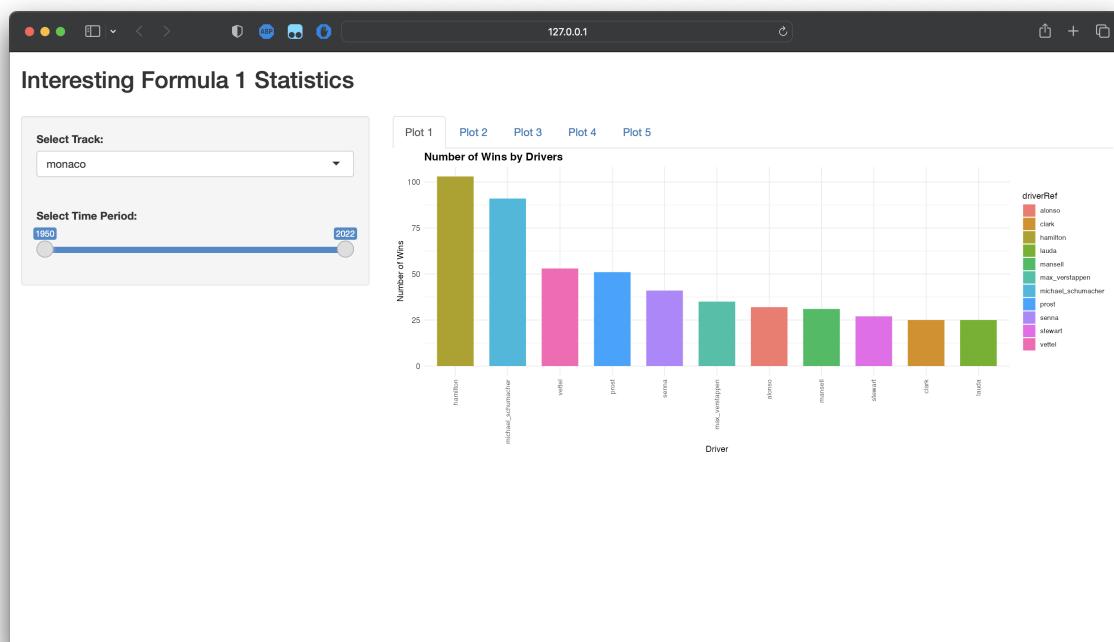
```

1 install.packages(c("tidyverse", "ggplot2"))
2
3 library(tidyverse)
4 library(ggplot2)
5
6 setwd("~/Documents/Egyetem/_MSc/3_szemeszter/erasmus/information_visualization/project")
7
8 results <- read.csv("f1_dataset/results.csv", sep = ",")
9 drivers <- read.csv("f1_dataset/drivers.csv", sep = ",")
10
11 results.filtered <- results %>%
12   select(raceId, driverId, positionOrder) %>%
13   filter(, positionOrder == "1")
14
15 drivers.filtered <- drivers %>%
16   select(driverId, driverRef)
17
18 summarise <- merge(x = results.filtered, y = drivers.filtered, by = "driverId") %>%
19   group_by(driverId) %>%
20   summarise(sum = sum(positionOrder)) %>%
21   filter(, sum >= 25)
22
23 exampleTask <- merge(x = drivers.filtered, y = summarise, by = "driverId") %>%
24   arrange(desc(sum)) %>%
25   select(driverRef, sum)
26
27 ggplot(exampleTask, aes(x = reorder(driverRef, -sum), y = sum, fill = driverRef)) +
28   geom_bar(stat = "identity", width = 0.7) +
29   labs(x = "Driver", y = "Number of Wins", title = "Number of Wins by Drivers") +
30   theme_minimal() +
31   theme(plot.title = element_text(size = 14, face = "bold"),
32         axis.text.y = element_text(size = 10)) +
33   theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .5))
34
  
```

## Prototype 1 - Code

### Prototype 2 - Test plot

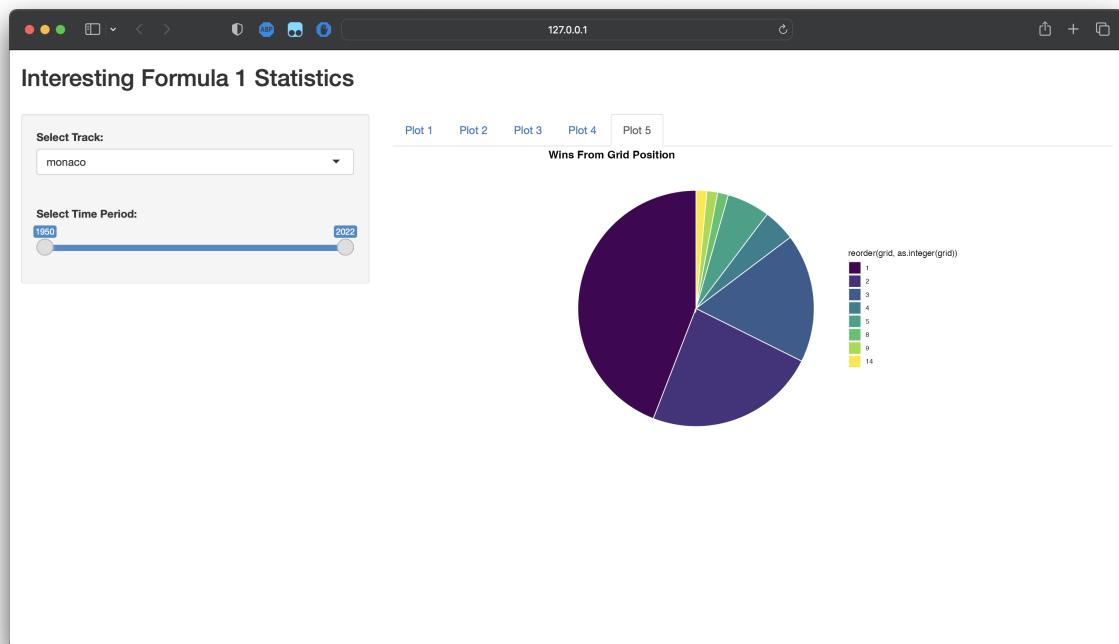
In the second prototype, I created the base of the UI with some controls. However, these controls were still not functional. But for each tab, I added a different plot and the users could switch between them. This was the first interaction they could do with the visualizations.



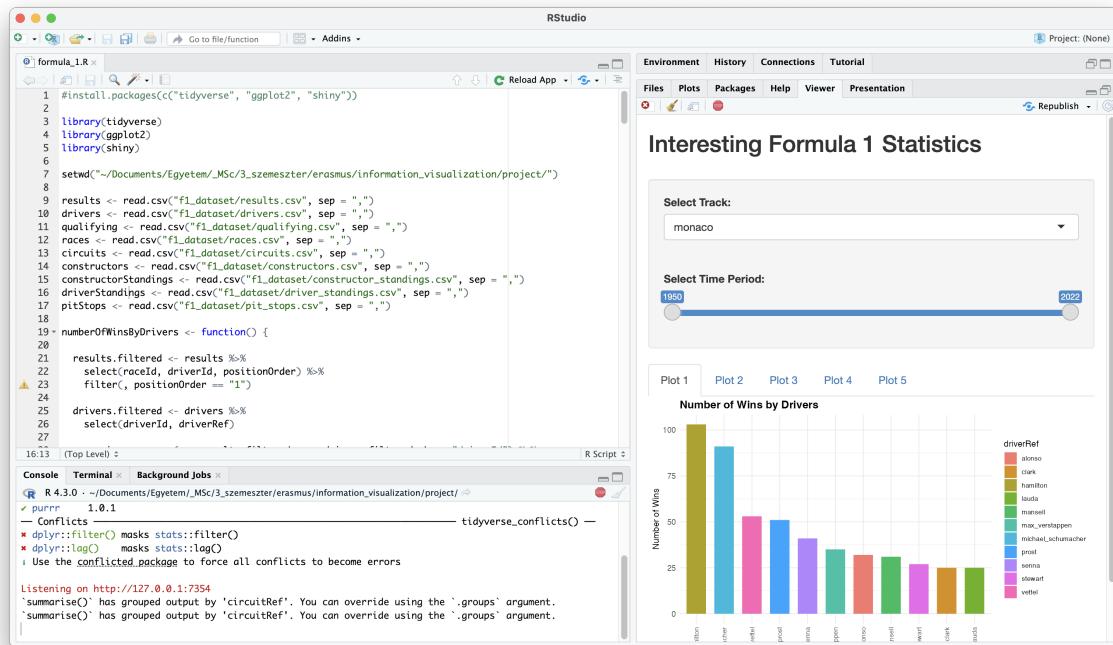
## Prototype 2 - Plot 1



Prototype 2 - Plot 2



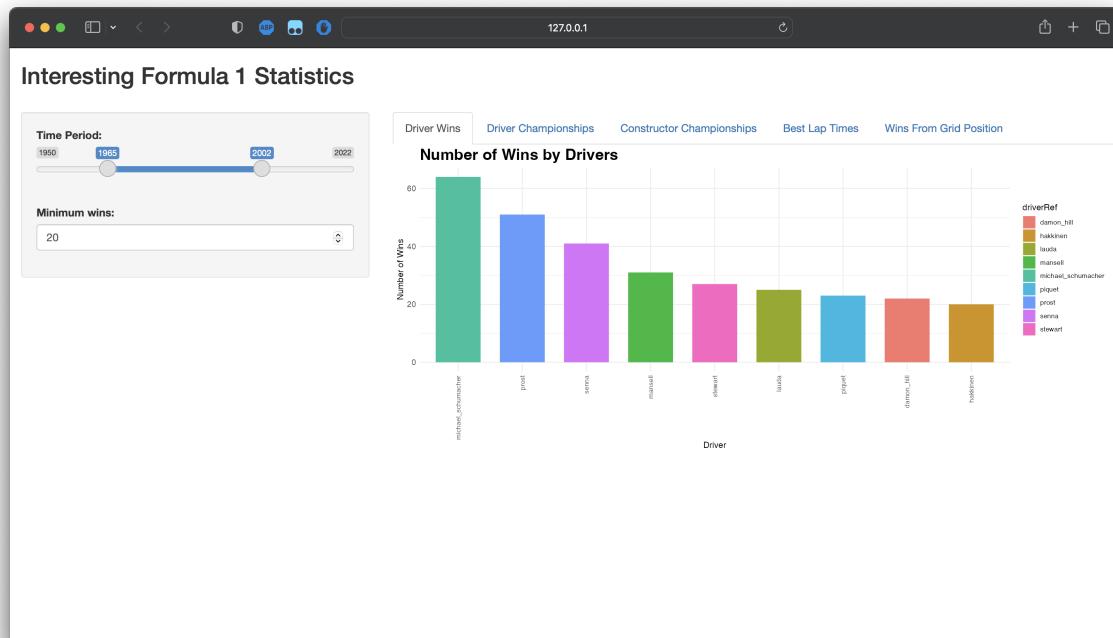
Prototype 2 - Plot 5



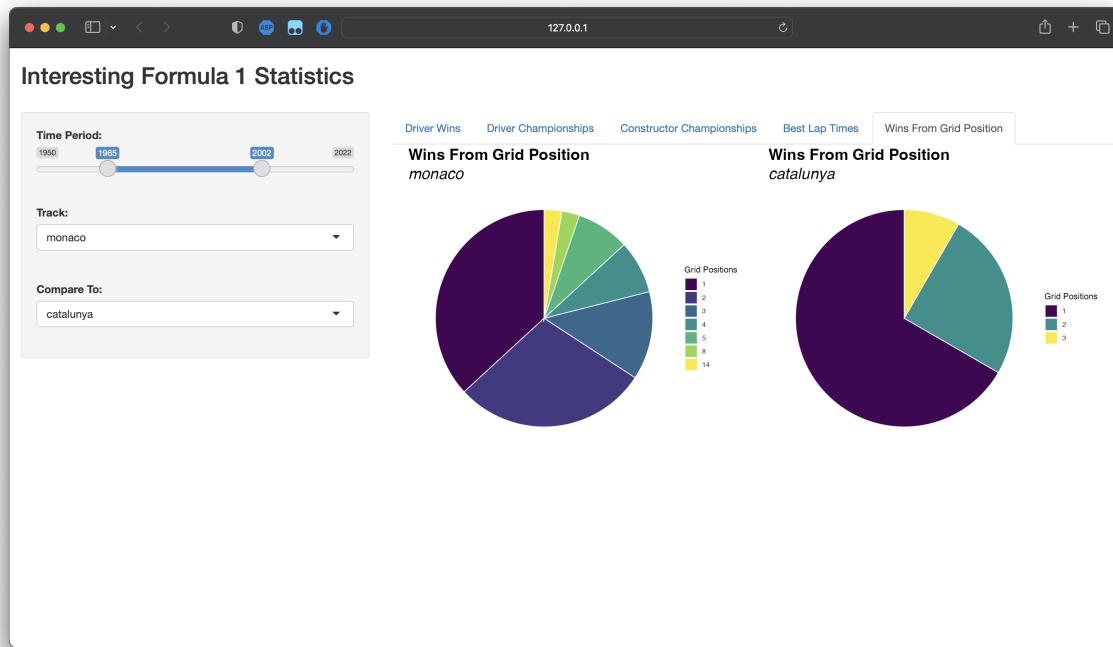
## Prototype 2 - Code

## Prototype 3

In this prototype, I connected the UI controls with the plots itself, so these interactions already worked now. However, the plots were still not polished and the UI needed more interactivity.



## Prototype 3 - Driver Wins



Prototype 3 - Wins From Grid Position

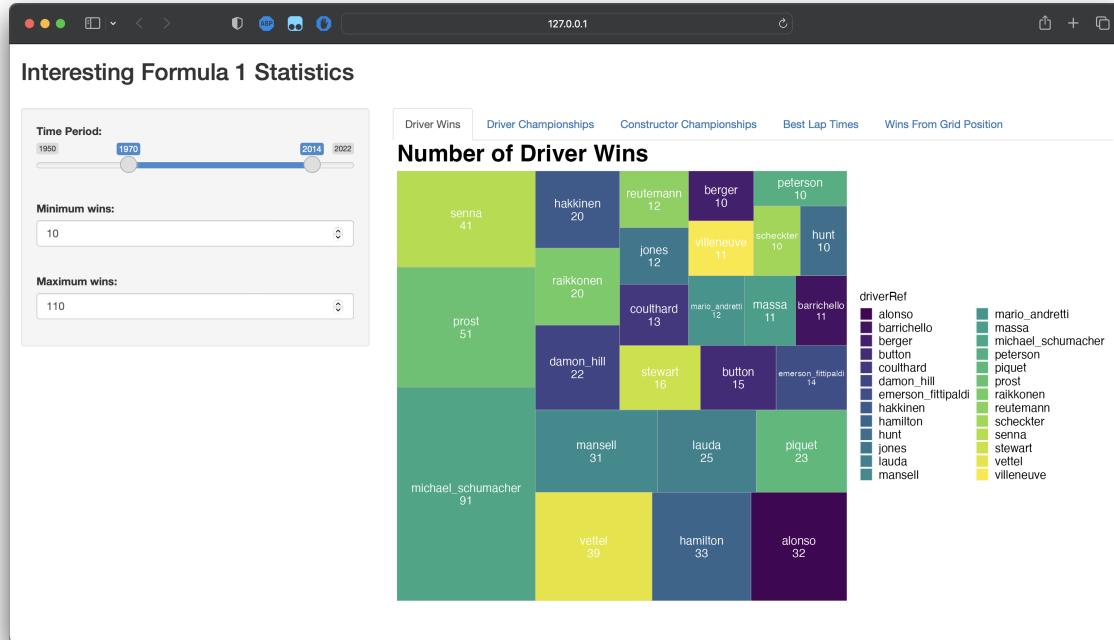
The screenshot shows the RStudio interface with several windows open:

- Code Editor:** The main window displays R code for loading packages and reading CSV files from a local directory.
- Environment:** A sidebar showing the global environment with objects like `drivers`, `qualifying`, `races`, `circuits`, `constructors`, `constructorStandings`, `driverStandings`, and `pitStops`.
- Plots:** A chart titled "Number of Wins by Drivers" showing the count of wins for various drivers. The y-axis ranges from 0 to 60+ wins, and the x-axis lists drivers: damon\_hill, hakkinen, lauda, manSELL, michael\_schumacher, paget, prost, senna, and stewart.
- Output:** The R console shows the command `gridExtra::grid.arrange` being run.
- Terminal:** The terminal window shows the command `R 4.3.0 - ~/Documents/Egyetem/_MSC/3_szemeszter/erasmus/information_visualization/project/`.
- Project:** The top right corner shows the project name "Formula 1 Statistics".

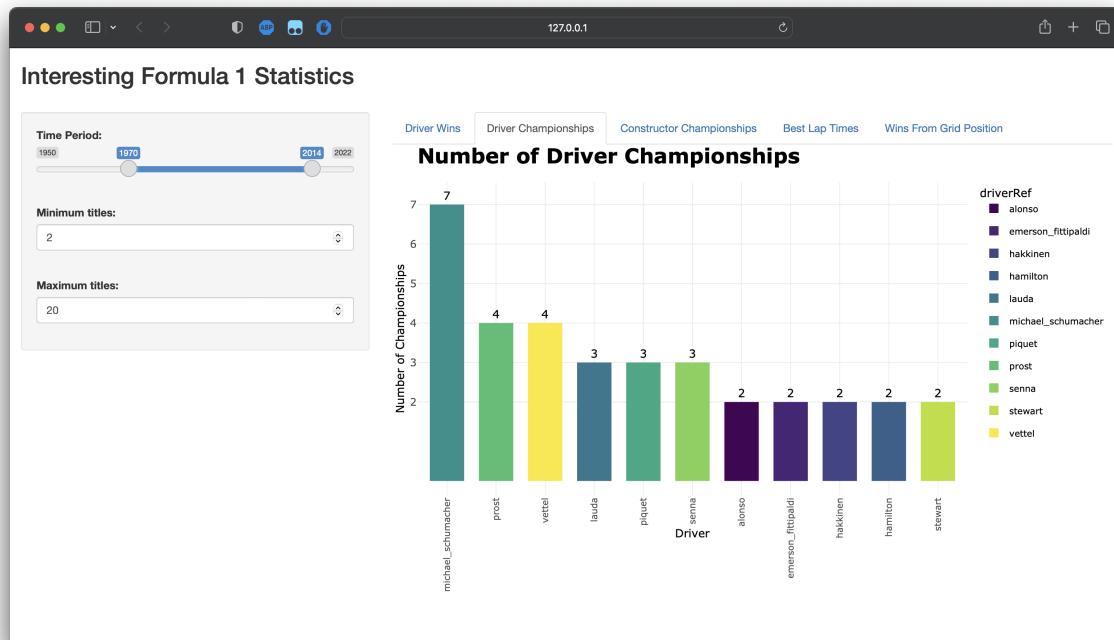
## Prototype 3 - Code

## Prototype 4

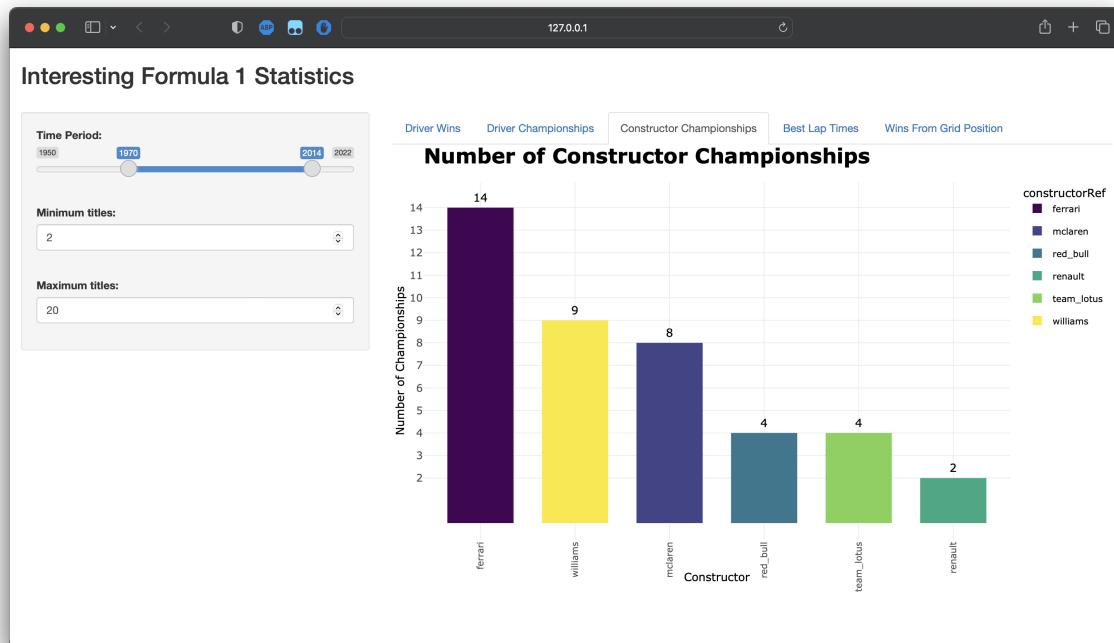
In the final prototype, I tried to fix every bug that I found and made the UI even more interactive with some plots created using *plotly*. I also made the visualizations more interesting, I customized the pie charts even more and I introduced a treemap as well for one of the views.



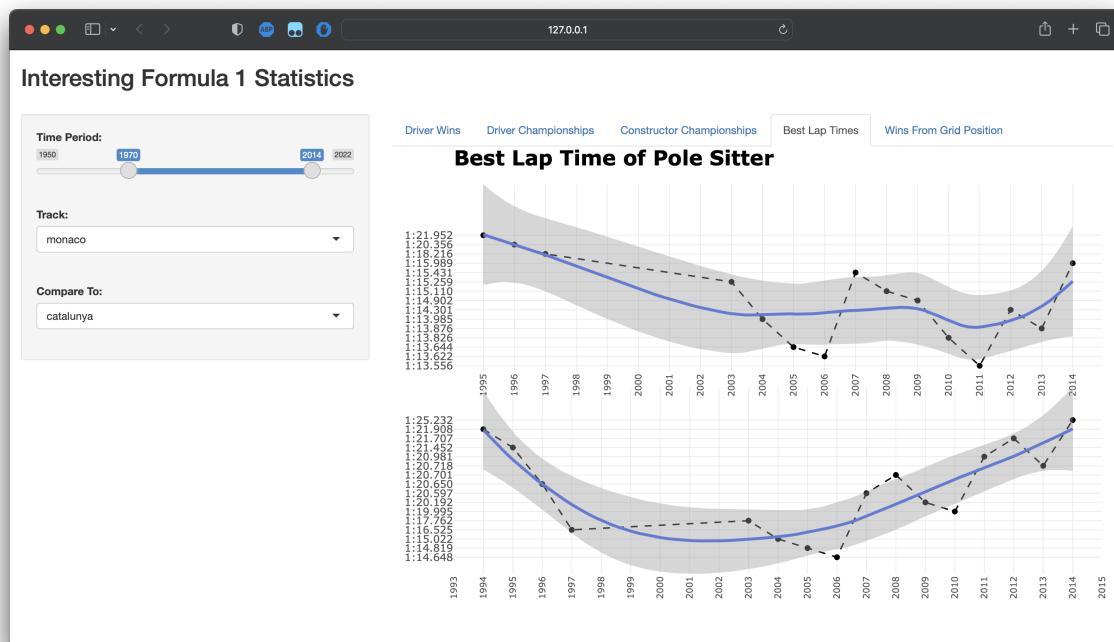
Prototype 4 - Driver Wins



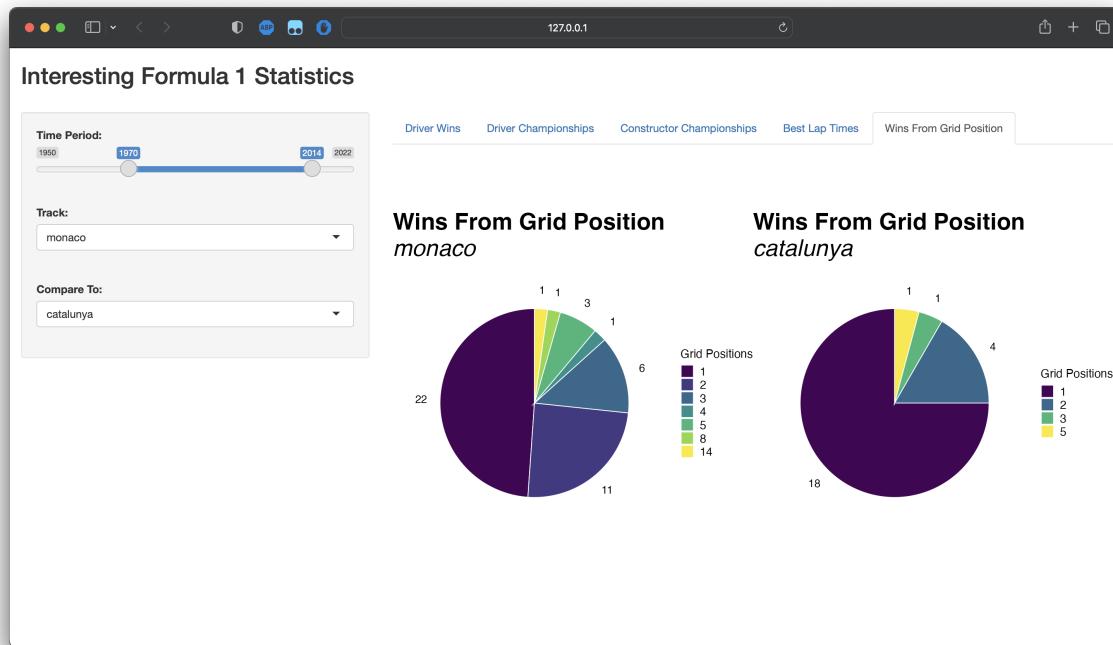
Prototype 4 - Driver Championships



Prototype 4 - Constructor Championships



Prototype 4 - Best Lap Times



Prototype 4 - Wins From Grid Position

The screenshot shows the RStudio interface with two main panes. The left pane displays an R script titled 'formula\_1.R' containing code to install packages, load data from CSV files, and define UI components. The right pane shows a visualization titled 'Interesting Formula 1 Statistics' with a timeline from 1950 to 2022, highlighting the years 1970 and 2014. It includes sections for 'Minimum wins' (10) and 'Maximum wins' (110). Below the timeline are tabs for 'Driver Wins', 'Driver Championships', and 'Constructor Championships'. Under 'Driver Wins', a treemap chart titled 'Number of Driver Wins' shows the distribution of wins by driver, with the largest share going to Senna (41).

## Prototype 4 - Code

## Scenario

TODO