Analyzing F1 Race Outcomes Using Data Correlation Methods

Tungjai T. Mady

Computer Game Programming, Lulea University of Technology

S0007E: Specialization Project

12 February, 2025

Table of Content

Abstract

Formula Racing is a sport that depends on many factors, and the ability to gather data, and analysis gives the team huge advantages. This paper will present methods to identify key variables that influence the race outcomes, and then give a conclusion on which methods give the most accurate result. The data sent will be taken from OpenF1, and the application will be written in python, one of the most common data analysis languages. By understanding different variables we could see the strength, weakness and with that information the team could come up with a strategy to encounter it. What is the most effective data analysis method to test variables that influence F1 race outcomes using OpenF1 data API [1].

Introduction

Formula One (F1) is the highest class of international racing, 10 teams of 2 compete around the globe for the world championship position. With modern technology, all parts of the cars are technologically advanced and highly sophisticated. Software use such as data analysis and machine learning are not excluded. F1 teams have sensors around the cars, and its surroundings, with the help of this software, it could potentially output the team's desired data for further analysis and improvement.

Correlation between variables could be found with data from open source API such as OpenF1 API. Data will be processed in python with different libraries such as pandas [2] and matplotlib [3], then will be outputted visually depending on the individual.

This paper will be extracting information from the API and see the correlation between variables, and try to determine the variables that have high correlation with winning. The method that will be used to find the correlation is pearson spearman and kendall correlation . This will give a different perspective on how different correlation methods interpret the relationship between variables.

Literature Review

In the Data Analysis using Python paper [4], the paper shows comprehensive steps for data analysis. From collecting data, processing, cleaning, using algorithms, and generating data products. The paper followed this method rigorously. With the help of different API methods in

OpenF1 [5], and pandas [2] built-in function, it managed to gather data from the cloud, organize it and output it into a database, hence simplifying the process tremendously.

The three methods that will be used to analyze the data are Pearson, Spearman, and Kendall correlations [6]. The difference is that people expect the two variables to be somewhat linear, so at least we have to assume that there's a relationship between them. While Spearman and Kendall do not carry any assumptions, instead they rely on a rank-based system to measure the relationship between variables. This makes Spearman and Kendall correlation more useful in many cases, since it is able to analyze nonlinear relationships or dataset with outliers.

Methodology

Pandas [2], and OpenF1 makes data gathering relatively easy. Open F1, have an example on their API methods page [5]. The problem is that there's a limitation to the number of requests per period of time. To overcome this limitation, all data is gathered and exported to a database such as Microsoft Excel. This can be done by using the pandas library that exports data frames into excel. To overcome the API limitation we will be using Python's built-in sleep function [7], the algorithm will sleep for a couple of seconds before requesting another request to the API server.

By exporting all data into excel, it means that the data could be fetched spontaneously without any limitation. The downside of this is only needing the storage to store the data itself.

When testing between two variables, it is necessary to check that both variables are within the same time frame (race, hour, min, sec), and that the value is not null. With the built-in function in pandas, it is relatively easy to check if the data is in the same time frame, or if variables are null.

All three correlation methods Pearson, Spearman, and Kendall are also available in the pandas library [8]. This makes working with all the data much easier, for both scaling and optimizing.

Another variable that was tested is the air temperature, the data of the air temperature is from OpenF1 API [1]. Air temperature in celsius is divided into two categories, high and low based on the mean air temperature of all the races. The mean air temperature is calculated by, getting the mean of each race, and sum it all up and divide by the number of races.

Each race will be labelled as either high or low air temperature. The average position for high and low temperature is being calculated, and the difference in those two will indicate the performance difference. The goal here is to showcase that this type of comparison is available, and could act as a template of the future.

Implementation

In terms of implementation, the code itself consists of 4 different modules. One of them is working with the API, so everything related to calling the URL, fetching the data and turning it into a data frame is within that module.

Data Utilities, this module consists of all the functions that are related to data cleaning, whether it is removing data, filtering out the desired data, and data manipulation, so everything related to changing the data format to be more versatiles, and easily understood.

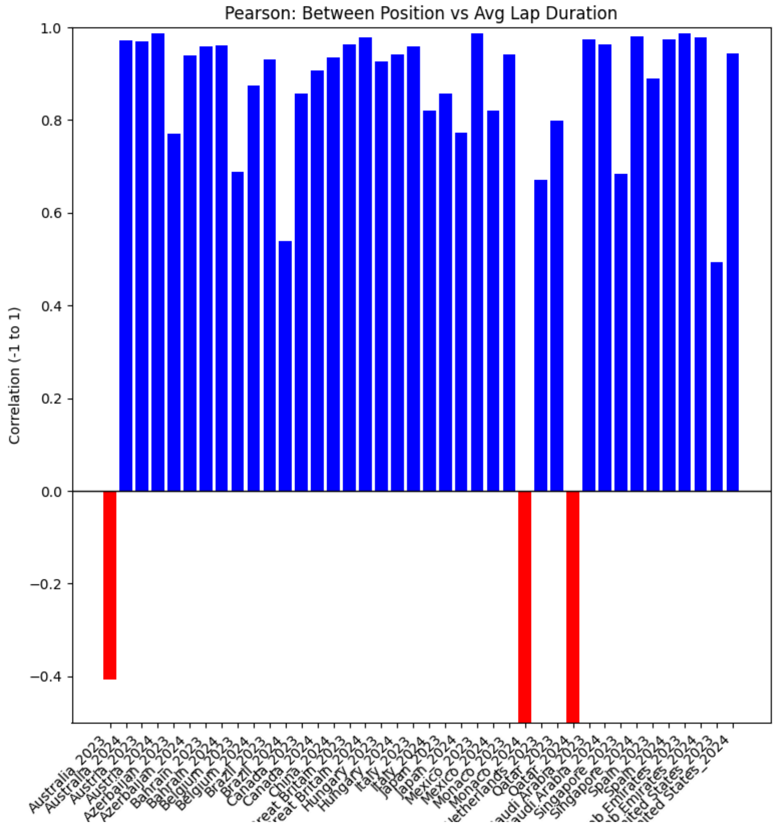
Main modules, which is what we called at the first of the program. This will call all the necessary functions, and act like a centralized hub.

Result and Discussion

Using Pearson linear correlation [6] to test "average lap time" against "winning position" should in theory show a high correlation, since lower lap time means you're the fastest hence more likely to win. However the result shows a mixed result depending on the race. After taking a look at the data, the data shows many data points that are null, this means there's a problem with data cleaning. A null data point really messed up the whole study.

A measure taken was to use the pandas library to discard all variables with NaN (Not a Number) or empty values. However, there is an edge case where the driver did not complete the race but is still being taken into account for the correlation. Since the driver did not finish the race, this will mess up the average lap duration, hence messing up the result. This happened in 3 of the races. **Figure 1.1** highlights three red bars, which indicate discrepancies when drivers who did not finish a race are included.

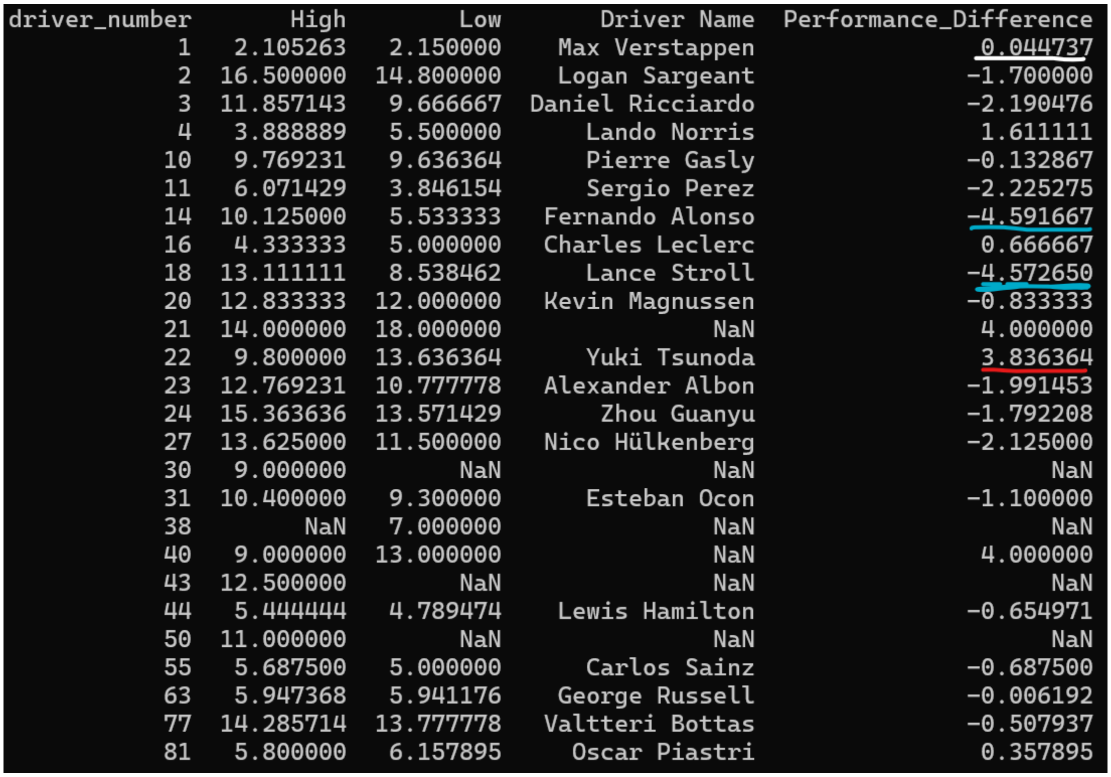
Figure 1.1



Shows the Pearson Correlation between position and average lap duration.

Air temperature and performance differences are also being tested. The result **Figure 1.2** shows that some drivers are better in hotter air (if performance difference is positive), and some are better in colder air (if performance difference is negative). NaN could mean that the driver never finished the race in hot or cold temperature, or there's some external factor that the variables are not usable.

Figure 1.2



```
driver_number       High        Low       Driver Name  Performance_Difference
            1   2.105263   2.150000    Max Verstappen                0.044737
            2  16.500000  14.800000    Logan Sargeant               -1.700000
            3  11.857143   9.666667  Daniel Ricciardo               -2.190476
            4   3.888889   5.500000      Lando Norris                1.611111
           10   9.769231   9.636364      Pierre Gasly               -0.132867
           11   6.071429   3.846154      Sergio Perez               -2.225275
           14  10.125000   5.533333   Fernando Alonso               -4.591667
           16   4.333333   5.000000   Charles Leclerc                0.666667
           18  13.111111   8.538462      Lance Stroll               -4.572650
           20  12.833333  12.000000   Kevin Magnussen               -0.833333
           21  14.000000  18.000000               NaN                4.000000
           22   9.800000  13.636364      Yuki Tsunoda                3.836364
           23  12.769231  10.777778   Alexander Albon               -1.991453
           24  15.363636  13.571429       Zhou Guanyu               -1.792208
           27  13.625000  11.500000   Nico Hülkenberg               -2.125000
           30   9.000000        NaN               NaN                     NaN
           31  10.400000   9.300000      Esteban Ocon               -1.100000
           38        NaN   7.000000               NaN                     NaN
           40   9.000000  13.000000               NaN                4.000000
           43  12.500000        NaN               NaN                     NaN
           44   5.444444   4.789474    Lewis Hamilton               -0.654971
           50  11.000000        NaN               NaN                     NaN
           55   5.687500   5.000000      Carlos Sainz               -0.687500
           63   5.947368   5.941176    George Russell               -0.006192
           77  14.285714  13.777778   Valtteri Bottas               -0.507937
           81   5.800000   6.157895     Oscar Piastri                0.357895
```

Performance Different in each weather condition

Reflection

Open F1 Api [1] is a third party open source API, therefore some of the data is missing, broken, or missing a dataset. Gathering data from more than one place might be a good first change to the project, making it even more reliable since the source is distributed.

Exploring different algorithms not just finding correlation, pearsons, spearman, and kendall all try to find the correlation between two variables. Coming up with an algorithm that is able to compare different variables might give a different perspective, and see which variables have a heavier weight than others.

The main improvement is to see tested out the most important variables when it comes to winning in one season, and then tested to see if those variables correlate to winning in another season. This will also be an additional proof that these methods have high correlation to winning position.

Conclusion

Most of the variables tested have shown a linear relationship, therefore the graph tends to show a high correlation. From the results tested, it can be concluded that average laps duration is highly correlated with winning position. For air temperature it is very subjective to the driver, this could be the driver 's personal preference, and some shows high consistency despite the temperature.

The main take is that data cleaning is very important, since data analysis works like GIGO (Garbage in, garbage out). If the data input is not poor or error prone it is more likely to mess with the outcome.

Reference

[1] OpenF1, "OpenF1: F1 Telemetry Data API,"

[2] Pandas, “Python Data Analysis Library”

[3] Matplotlib, “Visualization with Python”

[4] DataAnalysisPython, “Data Analysis using Python”, pg. 463 - 464

[5] OpenF1, "OpenF1: API Methods,"

[6] Correlation, “Correlation (Pearson, Kendall, Spearman)”

[7] PythonTime, “Time access and conversions”

[8] PandasCorrelation, “Data Frame Correlation”

[9] GithubF1Repos, “Main Repository”