A Bayesian Approach to Determining Formula One Strength

Vishnu Bindiganavile | vbindiga3@gatech.edu

Abstract

We propose a Bayesian hierarchical regression model to statistically model driver skill ratings for Formula One drivers and determine a historical top driver ranking. We also use the results from this model to determine a numerical strength indicator for constructors and analyze constructor performance over time.

1. Introduction

The top division of motorsports, also known as Formula One, has been around since 1950 [1], drawing in millions of fans worldwide. With a wide variety of drivers and constructors participating in the competition over the decades, fans often speculate exactly who is the best driver or constructor.

Oftentimes, sole indicators of driver or constructor dominance are used. For driver dominance, driver championship wins, grand prix wins, or podium finishes are used. For constructor dominance, constructor championship wins, driver championship wins, or top 3 finishes in constructor championship standings are used.

However, these statistics are not the tell all of driver or constructor performance. A driver that is the best may start or end his career on a poor team, and still manage to perform quite well in comparison to the other teams in the bottom of the standings. The specific year of a driver's participation also matters for their constructor's strength. For example, the Ferrari of 2000 was wildly different from the Ferrari of 2020. When considering constructor performance, a team with consistent finishes in the constructor standings each year, rather than extreme highs and lows, is clearly the stronger constructor over time.

We theorize that driver performance from each race (only considering races where that driver finished the race, in order to ignore the random effects that may lead to race disqualification) aggregated can indicate driver skill over their career.

In this paper, we develop a hierarchical linear regression model using a Bayesian approach that allows us to both statistically determine the top 10 drivers of all time as well and compare constructors performance over time and determine their strength in relation to other teams.

1.1. Terminology

To assist in the understanding of this paper, we define some motorsport-related terms below.

Podium positions are considered first, second, and third place.

Pole position is the highest starting spot in a Formula One grid on race day. If a driver is "starting on pole" for the race, they are starting in the front-most position.

Championships are specific trophies/standings that each driver and constructor fight for every year. The driver's championship determines the top driver each year based on points. The constructor's championship determines the top constructor each year based on points.

2. The Data

We use a dataset that is shared on a known dataset aggregator site called Kaggle [2]. The dataset, named "Formula One World Championship (1950-2024)" [3], contains various information circuits, races, drivers, constructors, results, etc. for all complete seasons of Formula One.

	grid	position	milliseconds	year	driverName	constructorName
0	1	1	5690616	2008	Lewis Hamilton	McLaren
1	5	2	5696094	2008	Nick Heidfeld	BMW Sauber
2	7	3	5698779	2008	Nico Rosberg	Williams
3	11	4	5707797	2008	Fernando Alonso	Renault
4	3	5	5708630	2008	Heikki Kovalainen	McLaren

Table 1: Head of the used dataset following table selection, merges, feature selection, feature preparation.

2.1. Qualitative Analysis of the Features

1. *Grid*

   Grid positions are the starting positions for the drivers at the beginning of each race. They are determined during a qualifying session that takes place a day prior to each race. The grid is formed out of ten rows where two drivers line up next to each other. For example the first and second driver line up in the first row, the third and fourth driver line up in the second row, etc.

   Although qualifying position, which determines grid position, is not the end all be all for a driver's race weekend, it can hugely determine the outcome of the race and the final position they end up in. Furthermore, a driver who is more consistent in qualifying may also be a better driver. We include this feature in order to incorporate the importance of the entire race weekend (including qualifying) in determining a driver's skill.

2. *Position*

   Position indicates the final position result of the race for the driver. The lower the position (the lower the better), the faster the driver was compared to other drivers. Although it would make sense to simply use time to explain how fast a driver was, some race circuits take longer to complete than others. Thus, times are not necessarily normalized and there is no easy way to normalize them. So, we include position as a feature in order to encode this information in the model.

3. *Milliseconds*

   Milliseconds indicates the time it took the driver to complete the race in milliseconds. Since races take hours to complete, the simplest denomination to represent the race time is milliseconds.

4. *Year*

   Year indicates the specific year that each record takes place. Year is important because it is combined with the constructor feature to use as a constructor_year feature, as explained below.

5. *Driver Name*

   Driver name simply indicates the name of each driver and allows us to identify each record as belonging to a certain driver and affecting their driver skill.

6. *Constructor Name*

   Constructors are the specific teams that various drivers race for. Great examples of constructors that are well known are Ferrari, Mercedes, Red Bulls, McLaren, etc.

Sidenote: The constructor_year feature in the model combines the constructor name and year features in the data. We decided to do so in order to highlight the importance of constructor strength being different each year. As explained above, the Ferrari of 2000 is wildly different strength-wise than the Ferrari of 2020. So, we concatenate constructor name and year to include that in the calculation of skill and also to calculate constructor strength over time for separate analysis.

3. Approach

We develop a Bayesian hierarchical linear regression model in order to calculate both

driver skill and constructor strength over time. We define this model as hierarchical because we use the driver and constructor_year features as groupings. Thus, when calculating the overall response, we use group priors to model the effect of those specific groups. We define this model as linear regression because we use multiple coefficients that can essentially be broken down into $y = m_1 + m_2 + m_3 x_3$.

Each model is sampled and trained using the data and the `pymc` Python library. We collect 2000 samples and use 500 samples as burn in.

We use grouped priors for both driver skill and constructor strength, in addition to a coefficient term paired with grid position, in order to predict the observed milliseconds race time value. We do so because there is no inherent driver skill or constructor strength value to train against. Furthermore, race time is an intrinsic quality of driver skill and constructor strength because the faster the driver is the better. By using driver skill and constructor strength grouped priors they can be trained as well through sampling and are also affected by the race time, which is the goal.

Following the model training, we use the posterior mean of the skill level for each driver We then rank the top 10 drivers using this driver skill inference data. Similarly, we analyze constructor strength over time, both within the constructor and relative to other constructors, using the constructor strength inference data.

4. Bayesian Linear Regression

4.1 The Model

The model is defined as follows:

$$\beta_d \sim N(0, 1), \ for\ each\ d \in drivers$$

β_d is the prior for each driver skill coefficient. We place a prior distribution on each driver's individual skill and through sampling, train each driver's skill only on the records that are associated with that driver. By sampling for the observed milliseconds of time taken for each race, the model learns the driver skill coefficient for each driver.

We use a normal distribution with a mean of 0 and variance of 1 since we want each driver's skill to solely be determined based on their race times. We use a variance of 1 because we want to assume that every Formula One driver is skilled enough to participate in the series. Therefore, their skill should not deviate too far from 0 and there should not be any major skill gaps between drivers.

$$\beta_c \sim N(0, 1), \ for\ each\ c \in constructors$$

β_c is the prior for each constructor strength coefficient. We place a prior distribution on each constructor's strength and, through sampling, train each constructor's strength only on the records that are associated with that constructor. Similar to the process used for driver's skill, the model learns the constructor strength coefficient for each coefficient.

Similar to driver's skill, we use a normal distribution that allows the constructor strength to solely be determined based on their driver's race times. Again, a variance of 1 is used to explain that the constructor strength overall should not vary an incredible amount from the others.

$$\beta_g \sim N(0, 100)$$

β_g is the prior for the grid position coefficient. Since grid position is important for quantifying the driver's performance in qualifying, we use a separate coefficient to model the effect of grid position.

We use a normal distribution with a mean of 0 and a variance of 100 because we are unsure of exactly how important grid position is for predicting race time. Usually, the lower the grid position (the further up on the grid a driver starts), the higher the position the driver will end

up. However, driver skill and constructor strength play a huge role as well.

$$\mu_i = -\beta_{d,i} - \beta_{c,i} + \beta_g * g_i$$

μ_i is the calculated mean for the race time for each record. We add and subtract as shown based on the following notions. Driver skill has a negative relationship with race time. In other words, the higher the driver skill (the better they are), the lower their race time (the faster they are). Constructor strength also has a negative relationship with race time. The higher the constructor strength (the better they are), the lower their race time (the faster they are). On the other hand, grid position has a positive relationship with race time. The higher the driver's starting grid position (the farther back that they start), the higher the race time (the slower they are).

We only multiply the coefficient of grid position with the actual record value of grid position because that data point contributes to the prediction of race time. The coefficients for driver skill and constructor strength are based on race time and get trained during the sampling.

$$\tau \sim Ga(0.001, 0.001)$$

τ is the tau, or 1 divided by the standard deviation of the race time for each record. We use this uninformative prior in order to allow the data to truly describe how race time is determined and what it determines about driver skill and constructor strength.

$$m_i \sim N(\mu_i, \frac{1}{\tau})$$

m_i is the distribution that is used to sample race time. We use a normal distribution using the mean calculated from the features and tau calculated from the prior. Each m_i is the race time for a record which is pulled from a distribution involving all of the required features. Furthermore, this is the variable that we link with the observations from the training data.

4.2 Fitting the Model

Using `pymc` to use 2000 samples and 500 as burn in, we obtain the following summary:

	mean	sd	hdi_2.5%	hdi_97.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
driver[0]	-4.664	0.549	-5.752	-3.603	0.005	0.006	10516.0	6464.0	1.0
driver[1]	-0.822	0.938	-2.742	0.947	0.007	0.013	16596.0	5031.0	1.0
driver[2]	-0.110	0.915	-1.980	1.594	0.007	0.012	15689.0	5016.0	1.0
driver[3]	-0.077	0.950	-1.878	1.809	0.008	0.012	15192.0	5233.0	1.0
driver[4]	-10.058	0.362	-10.740	-9.335	0.004	0.004	7925.0	6278.0	1.0
...	...	...	...	...	...	...	...	...	...
constructor[678]	-1.730	0.700	-3.122	-0.376	0.006	0.008	12122.0	6154.0	1.0
constructor[679]	-1.793	0.797	-3.308	-0.148	0.007	0.010	14074.0	5817.0	1.0
constructor[680]	-0.979	0.814	-2.599	0.630	0.006	0.011	16134.0	5621.0	1.0
beta_grid	0.700	0.010	0.681	0.718	0.000	0.000	2537.0	4109.0	1.0
tau	0.089	0.003	0.084	0.095	0.000	0.000	2711.0	4710.0	1.0

Table 2: Returned table of the `az.summary(trace)` function

4.3 Interpretability

Our model has great interpretability because you can easily see the calculated skill for each driver and strength for each constructed. Furthermore, you can see the weight placed on grid position for the calculation of race time.

4.4 Driver Results

The top 10 drivers list that our model generates are as shown (along with their generated driver skill, for reference):

1. Ayrton Senna (10.297)
2. Alain Prost (10.058)
3. Michael Schumacher (9.924)
4. Lewis Hamilton (9.526)
5. Nico Rosberg (9.434)
6. Sebastian Vettel (9.018)
7. Max Verstappen (8.953)
8. Gerhard Berger (8.936)
9. Nelson Piquet (8.854)
10. Fernando Alonso (8.824)

These results should compare to some baseline of statistically defined top 10 Formula One drivers. We use a baseline as defined in an Autosport article [4], where drivers are ranked based on number of world championships, wins,

pole positions, and career points. The ranking is as follows:

1. Lewis Hamilton
2. Michael Schumacher
3. Max Verstappen
4. Sebastian Vettel
5. Alain Prost
6. Ayrton Senna
7. Fernando Alonso
8. Nigel Mansell
9. Jackie Stewart
10. Niki Lauda

There are several drivers that overlap between the lists. However, the order of those overlapped drivers are quite different. Additionally, there are many drivers that are removed and replaced by new names in various positions. Using driver skill to rank drivers as shown in the paper leads to more consistent drivers that may not have stacked world championship or outright grand prix wins as in excess as the drivers in the baseline list.

4.5 Constructor Results

Using the constructor strength calculated during MCMC sampling, we can compare various constructors to numerically understand how they differ between each other and themselves over time.

First, we compare the four constructors that have repeatedly been in the top four of the constructor championship for the most of the past decade. The constructors up for comparison are Mercedes, Red Bull, Ferrari, and McLaren.

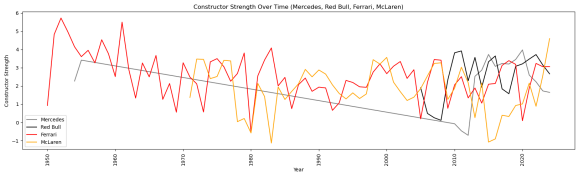

Table 3: Constructor strength over time for Mercedes, Red Bull, Ferrari, and McLaren.

Ferrari dominated for many years in the decades prior to 2020, but has become mediocre in relation to the other giants of Formula One. Mercedes standard off with relative mediocrity, but quickly rose to the top following 2010. Although Red Bull are a relatively new team, they have been better than most of the giants and and even better than all for a few years.

We also add a fallen giant such as Williams in order to view how they compared to those four teams both in the past and the present.

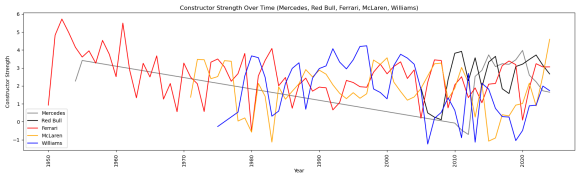

Table 3: Constructor strength over time for Mercedes, Red Bull, Ferrari, and McLaren.

Williams competed with the strength of the other giants for many decades. They were even better than all of the other constructors for the better part of the 1990s. However, during the past couple of decades, they have fell to mediocrity and have reached lows lower than the other giants.

5. Future Work

There is much room for improvement in the development of the model. The use of other race weekend statistics such as fastest lap time, free practice performance, overtakes, pit stop, yearly championship performance etc. may lead to a better performing model and better assessment of driver skill and constructor strength. The inclusion of adaptive constructor strength over time, not simply per year, may also yield interesting results

There are many interesting applications of this model. One such application is in the use of race simulations. Following qualification, the starting grid is set. Using the yielded driver skill and constructor strength, the model estimates

can be used to predict race time for each participating driver.

We show an example of this by using the 2024 Japanese Grand Prix results.

After qualifying, the standard grid was the following:

1. Max Verstappen
2. Sergio Pérez
3. Lando Norris
4. Carlos Sainz
5. Fernando Alonso
6. Oscar Piastri
7. Lewis Hamilton
8. Charles Leclerc
9. George Russel
10. Yuki Tsunoda
11. Daniel Ricciardo
12. Nico Hülkenburg
13. Valtteri Bottas
14. Alexander Albon
15. Esteban Ocon
16. Lance Stroll
17. Pierre Gasly
18. Kevin Magnussen
19. Logan Sargeant
20. Guanyu Zhou

The actual race results were the following:

1. Max Verstappen
2. Sergio Pérez
3. Carlos Sainz
4. Charles Leclerc
5. Lando Norris
6. Fernando Alonso
7. George Russel
8. Oscar Piastri
9. Hamilton
10. Yuki Tsunoda
11. Nico Hülkenburg
12. Lance Stroll
13. Kevin Magnussen
14. Valtteri Bottas
15. Esteban Ocon
16. Pierre Gasly
17. Logan Sargeant
18. Guanyu Zhou (DNF)
19. Daniel Ricciardo (DNF)
20. Alexander Albon (DNF)

The predicted race results are the following:

1. Max Verstappen
2. Lando Norris
3. Carlos Sainz
4. Sergio Pérez
5. Lewis Hamilton
6. Charles Leclerc
7. Fernando Alonso
8. Oscar Piastri
9. George Russel
10. Daniel Ricciardo
11. Nico Hülkenburg
12. Valtteri Bottas
13. Yuki Tsunoda
14. Alexander Albon
15. Esteban Ocon
16. Lance Stroll
17. Pierre Gasly
18. Kevin Magnussen
19. Logan Sargeant
20. Guanyu Zhou

The simulated results can be improved in their predictions of potential missed performances by superstars. The model predicted that Lewis Hamilton would recover from his poor qualifying result. However, his position worsened.

The model can also be improved to predict drivers that don't finish the race (DNF). However, this would require the initial model to use other features such as race status and other indicators that specify whether race status is on the fault of the constructor or the driver.

6. Conclusion

In this paper, we formulated a Bayesian approach to model Formula One driver skill and constructed strength based on grid position and

race time for various completed races. With this model, we determined a historical top 10 drivers list, analyzed constructor performance over time, and provided an initial attempt at raceday simulation using the generated driver skill and constructor strength.

Appendix

[1] F1. (2024, December 31). *Everything you need to know about F1*. Formula 1® - The Official F1® Website.

[2] Kaggle. (n.d.). https://www.kaggle.com/

[3] Vopani. (2025, January 29). *Formula 1 World Championship (1950 - 2024)*. Kaggle. https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

[4] Jeffries, T. (2024, December 8). *The 10 best formula 1 drivers ever: Hamilton, senna & more*. Autosport. https://www.autosport.com/f1/news/whos-the-best-formula-1-driver-schumacher-hamilton-senna-more-4983210/4983210/