

Analysis of the Various Factors Influencing the Success of FRC Teams

Troy Edwards

2023-06-16

Contents

1	Introduction	1
2	Method	2
2.1	Analysis of individual variables	2
2.1.1	Drivetrain	2
2.1.2	Scoring capability	3
2.1.3	Consistency	3
2.1.4	Team budget	4
2.1.5	Team size	4
2.1.6	Team age	4
2.2	Predictive models	4
2.3	Data collection and sources	4
3	Findings	5
4	Analysis	5
5	Conclusion	5
6	Bibliography	5

Abstract

TODO

1 Introduction

FIRST Robotics Competition is a robotics competition in which teams of high schoolers build robots to compete in a game that is different every year. FIRST is the organization that runs these competitions, and the acronym FIRST stands for “For the Inspiration and Recognition of Science and Technology.” Each year, FIRST reveals a new game in which

alliances of 3 industrial-size robots compete against each other on a field to complete various tasks. Teams have limited time after “Kickoff” (the game reveal) to design, build, wire, and program their robot to perform the tasks required for the game. For example, in this year’s game, CHARGED UP, robots must travel across the field to the Substation Area to pick up one game piece (a cube or a cone) at a time and then travel back to their alliance’s Grid and deposit the pieces onto Nodes.

There are many factors that could have an effect on a team’s success. These include general things about the team, such as the team’s age, its budget, and its size, and also things about their robot in a specific season, such as drivetrain type and scoring capability. The goal of this research is to determine which of these factors have the greatest effect on a team’s success.

TODO: ADD MORE CONTENT HERE

2 Method

2.1 Analysis of individual variables

I plan to analyze whether several variables have a significant influence on a team’s success.
TODO: ADD MORE HERE (HOW IS SUCCESS MEASURED?)

2.1.1 Drivetrain

A robot’s drivetrain can have a huge influence on how successful it is. FRC games are typically quite fast-paced, so a better drivetrain can make a huge difference. For most games, the playing field is quite flat, with few obstacles to overcome. This is true of both RAPID REACT and CHARGED UP. In games with few obstacles, teams will typically use some sort of “normal” drivetrain (i.e. one without huge wheels or anything like that). The three main types of drivetrain typically found in FRC are:

1. Differential
2. Mecanum
3. Swerve

A differential drive has normal wheels on the left and right sides of the robot. A Mecanum drive uses Mecanum wheels, which are special wheels with diagonal rollers mounted around the circumference. A robot with a mecanum drivetrain can drive forwards and backwards, similar to a differential drivetrain, but it can also drive sideways by driving specific wheels in specific directions. A swerve drive is another type of drivetrain that can drive in any direction, but unlike the Mecanum drivetrain, the swerve drive achieves this by using two motors for each wheel: one to drive the wheel, and another to steer the wheel. This allows each wheel to be steered independently of the others, which allows for complex maneuvers.

These can be grouped into two categories, namely holonomic (Mecanum and swerve) and non-holonomic (differential). The difference between these two categories is that while a

non-holonomic drivetrain constrains a robot to only move forwards and backwards such that it must turn to change its direction of motion, a holonomic drivetrain is free of these constraints.

A differential drive is usually the cheapest, since it requires no special parts. The next cheapest of these three is the Mecanum drive because you only need the 4 wheels and a motor for each. However, one major drawback of a Mecanum drivetrain is that the wheels are more prone to slipping. This means that a robot with a Mecanum drivetrain has a lower maximum acceleration before the wheels start to slip. It also means that in FRC, a robot with a Mecanum drivetrain is very easy to play defense against because a robot with a differential drive or a swerve drive can easily shove it around. Finally, the most expensive, but also the most effective is the swerve drive. The swerve drive combines the lack of slip (and by extension, pushing power) of a differential drivetrain with the flexibility of a holonomic drivetrain. The main drawback of swerve drive is that it can at times be prohibitively expensive, with a single swerve module typically costing upwards of \$300 not even including the motors. This is because they are very complex, with lots of gears and custom machined parts.

To analyze the effect of drivetrain on a team's success, I will perform two inference tests:

1. A two sample difference of means t -test to compare the success of teams using holonomic and non-holonomic drivetrains
2. A one-way ANOVA test to see if the difference in success is any greater if Mecanum and Swerve are not grouped together

2.1.2 Scoring capability

A robot's scoring capability can have a very direct effect on its success. If a robot can only score in locations that give low amounts of points, such as the LOWER HUB in RAPID REACT or the HYBRID NODES in CHARGED UP. Robots that are able to score in locations that give higher amounts of points are obviously going to score more points than robots that can't in almost every situation. From what I've seen, robots built by "good" teams tend to have this capability.

Also, in CHARGED UP, some teams can only score cubes or cones. Many teams can do both, but teams that can only do one could potentially be at a disadvantage.

In the event of a specific robot being a chassis bot (i.e. only a drivetrain, can't score any pieces), the scoring capability will be NA.

To analyze the effect of scoring capability I will do a different set of inference tests for RAPID REACT and for CHARGED UP:

1. For RAPID REACT, I will do a 2-sample difference of means t -test to compare robots that score low against robots that score high.
2. For CHARGED UP, I will do a one-way ANOVA test to see if there is a difference in the success of robots that can only score in the HYBRID NODES, robots that can

score in the MIDDLE NODES, and robots that can score in the HIGH NODES. I will also do a 2-sample difference of means t -test to compare the success of robots that can only score one type of game piece against that of those that can score both types.

2.1.3 Consistency

A common characteristic of robots that people would consider “good” is consistency. These robots tend to perform very consistently throughout an event and throughout the season, with the main source of inconsistency being an improvement in the skill of the driver. This will only be analyzed for the Pacific Northwest because the scouting data is the only data source detailing the performance of individual robots.

To analyze the effect of consistency on a robot’s success, I will calculate the standard deviation of the change in score between consecutive matches for all of a robot’s matches. Then, I will create a linear model using that standard deviation to predict win rate and perform a t -test for slope to see if the relationship is significant.

2.1.4 Team budget

It is easy to see why a team’s budget can affect its success. A higher budget can allow for a more complex robot with more expensive features such as a swerve drive.

To analyze the effect of budget on a robot’s success, I will create a model using budget to predict win rate and perform a t -test for slope to see if the relationship is significant.

2.1.5 Team size

A team’s size can also have a huge effect on its success. A larger team means that more work can get done, which means that the robot can be more complex and advanced. This will likely be correlated with some of the other variables, so it may not be included in the final model to avoid multicollinearity.

To analyze the effect of a team’s size on its success, I will create a model using team size to predict win rate and perform a t -test for slope to see if the relationship is significant.

2.1.6 Team age

An older team will likely have more advanced techniques and manufacturing processes at its disposal, while a younger team will typically not have these things. Team number is a good proxy to team age because the team numbers are assigned sequentially. This variable will also likely be correlated with many of the other variables, so it may not be included in the final model.

To analyze the effect of a team’s age on its success, I will create a model using team age to predict win rate and perform a t -test for slope to see if the relationship is significant.

2.2 Predictive models

After analyzing individual variables, I will create two models:

1. A multiple linear regression model using a combination of the above variables to predict a team's overall win rate
2. A multiple logistic regression model using a combination of the above variables and various metrics from a robot's allies and opponents to predict whether a robot will win a specific match

2.3 Data collection and sources

My data comes from three main sources:

1. Evan Kuykendall's 2023 scouting data from Glacier Peak, Auburn, and the Pacific Northwest District Championship
2. Jake Benjamin's 2022 scouting data from Auburn and the Pacific Northwest District Championship
3. The Blue Alliance

The scouting datasets contain information about each robot in every match that was scouted. An example of something that would be included in the data for 2022 would be number of pieces scored in the UPPER HUB while an example of something that would be included in the data for 2023 would be number of pieces scored in HYBRID NODES. The Blue Alliance only has more general data, such as match results (win/loss). However, it also contains information about who was on what alliance for a specific match. This will be useful for gathering data about alliance members to use in the logistic regression model.

```
union3 <- function(x, y, z) union(x, union(y, z))

all_available_teams <- union3(
  unique(wasno$Team.Number.you.re.scouting),
  unique(waahs$Team.Number.you.re.scouting),
  unique(pncmp$Team.Number.you.re.scouting)
)

teams <- sample(all_available_teams, 30)
print(teams)
```

```
## [1] 3711 2907 4089 6350 8896 4131 6443 8051 4180 4512 8386 7461 3049 1425 3786
## [16] 360 3674 4579 3826 NA 5827 6076 488 4911 3636 3393 955 3219 2928 9036
```

- 3 Findings
- 4 Analysis
- 5 Conclusion
- 6 Bibliography