Using Analytics to Better Understand Chess Outcomes

By
Cedric Harper, Adonis Lu, Kevin Malloy, Shenghua Wu

Introduction

With the recent conclusion of the Riga Grand Prix, one of the most important chess tournaments of the year, we were interested in examining and analyzing chess data. As casual players and passionate spectators, we hoped to discover which factors influence the outcomes of matches, and to what extent they do so. This analysis could then be used to help players formulate their own preparation and strategies to improve their own gameplay.

The other main question we decided to inspect was the accuracy of the Elo rating system, a system which rates players according to their performance levels, in the context of our data. Since its adoption by the US Chess Federation in 1960 and by the World Chess Federation in 1970, more recent rating systems, namely Glicko and Glicko-2, have been developed for other competitive sports. Since these other systems are also widely adopted, we were curious if the Elo system was still an accurate indicator for a player's level. We hoped to do this by comparing a logistic regression on our data against the hypothetical logistic elo rating function, which predicts a player's chance of winning given the rating difference between the two players.

Background

To obtain this data, we downloaded a dataset from Kaggle (https://www.kaggle.com/datasets), a website dedicated to providing open data. Data on Kaggle is user-submitted, so we were lucky to have a dataset that was very clean. It was evident that the submitter of this dataset was careful because the dataset contained no missing values, input errors, or anything suspicious.

In order to understand this data, we first introduce a fundamental understanding of chess, chess matches, and the Elo system. In chess, two players face off, each controlling an army of 16 pieces. Players are randomly assigned either white or black, the two opposing sides, and they alternate making turns back and forth, starting with white. A match can end in a victory/loss or in a draw. Achieving a victory can occur through checkmating the opponent, in which a player captures the opponent's king, the other player running out of time, or even resigning. On the other hand, a draw can occur in even more ways, but generally, when it is deemed that neither side will win. In the context of our dataset, the victory status is divided into checkmate, running out of time, and resignation, but all forms of draws are labeled simply as draw.

Since chess matches are timed, there are many accepted time formats for chess matches, and even more on online chess websites, which often allow a user create their own time format. However, in all these different formats, the time format is always comprised of two components: start time (minutes) and increment (seconds). Each time it is a player's turn to make a move, his time bank will then decrease. Once he completes his move, his timer stops and his opponent's clock starts ticking down. For the completion of each move, the player is then rewarded with the increment number of seconds to their current time bank. If at any point a player's time runs out, then the other player automatically wins.

Fundamentally, the Elo system is a rating system which assigns each player a numerical score that measures a player's performance. With each victory, a player's elo score will increase, and losses will lower his elo. To balance this, players are rewarded or lose rating depending on the relative strength of their opponent. So, higher rated players would gain fewer points if they defeat players with lower Elo scores, but would lose more points if they are defeated.

Exploratory Analysis

With regard to our dataset, we chose to only use the variables Rated (bool), Turns (int), Victory status (str), Winner (str), White rating (int), Black rating (int), Increment (str), and Opening play (int). Of these, only Rated, Victory Status, Increment, and Opening play were a bit confusing. Rated was a boolean value in which True indicated that the match was rated, and elo scores would be affected by the result of the match. Victory status was a string that detailed the outcome of the match, which included victory by checkmate, resignation, the out of time condition, or if a draw was reached instead. Increment was by far the most misleading, since it actually combined the start time and increments into strings formatted as "x+y". Finally opening play measured the number of turns from the opening, and not the actual sequence of moves.

After finding the data set and doing some background research on chess and the ELO rating system, we next wanted to do some exploratory analysis to learn some basic information about the data set. The first thing that jumped out at us was that for the variable "rated," 81% of the games were listed as true, and 19% as false. We understood this variable to mean that if rated is true, the players are playing with their ELO rating on the line; losing will result in a decrease in rating, and winning will increase it. If rated variable is false, the players are playing without any consequences to their ELO ratings. We decided to do filter our analysis only on rated games, because we felt these games ensured that players were playing to the best of their ability. This would help us remove any possible outliers in which players created matches with time formats set at absurdly low or high levels, which could influence the data greatly.

The next observation we found important was the breakdown of who was winning these matches. In our data set, approximately 50% of the total matches were won by the player who moved first (the player with the white pieces), 45% of matches were won by the player who moved second (black pieces) and 5% of matches ended in draws. This supports the general assumption among chess players that moving first is a sizeable advantage overall. With this information, we decided that we had to take into account who moved first when computing the probability of winning. We ultimately decided to frame our logistic regression models in terms of the probability of the white side winning, given the white and black sides' ratings. We will discuss in further detail our logistic regression models later, but for now it is simply important to note that moving first in chess is a pretty clear advantage.

In addition to observing the winner of the match, it was also interesting to look at how these matches ended. The most common ending, which occurs 56% of the time, is that a player resigns, most likely because he or she foresees themselves losing in a handful of moves. The next common ending, occurring 32% of the time, is the checkmate. The remaining 12 percent of matches either end in a player running out of time (7%) or a draw (5%).

All of these initial observations led us to form the questions we wanted to answer using our data set. The most important, we believed, was being able to estimate the probability of each player winning and the probability of a draw, given any information one would have going into the match. Then, we were curious to explore what influences how the match ends. Are the probabilities of draws, resignations, and checkmates dependent on either how good the players are (average ELO of a match) or how competitive a match is (difference in ELO ratings of a match)? Finally, our last goal was to explore what impacted the number of turns in a match. From the data, we found that the median number of moves is 57 moves. We hoped to utilize ELO rankings and information about how much time was given for a match to be able to better predict how many moves a match would take.

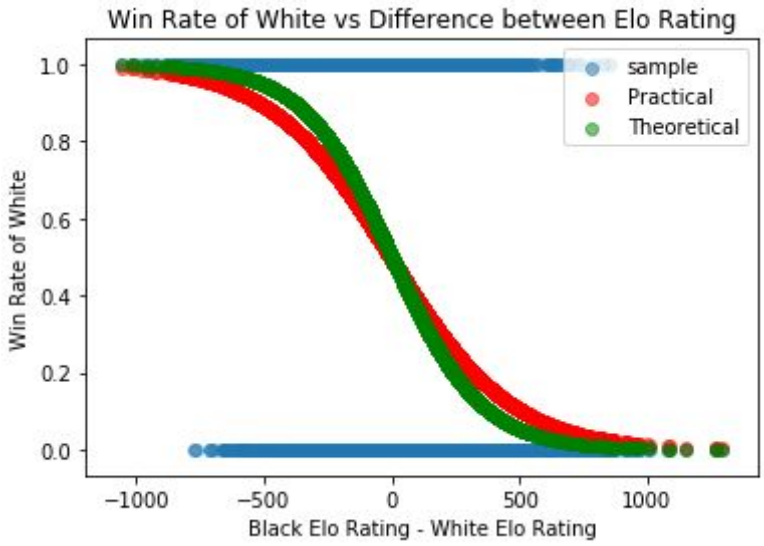
Section 1: Modeling Wins

First, we tried to predict each player's win rate based on their Elo ratings. The Elo system provides two equations that measure the probability of a player has of winning, given the Elo ratings of the players:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} . E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}} .$$

Theoretically, we are able to calculate win rate of a player using the ELO ratings, where R_a and R_b are the ELO ratings of players and E_a is the win rate of player A. From this equation, we can confirm that the win rate of equally rated players is 50%. Based on this, we were curious if a particular color affected win rate, since white is granted the first move. In terms of our model, we decided to frame our question in white's perspective, that is, the probability that white will win the match.

In the context of this model, we care about the winner, the two ratings, and whether or not the game was rated. Since the response variable 'winner' has three values, we assigned win as 1 and draw and loss as 0, which we used to fit a logistic regression. For our two independent variables, we used the difference between the two ratings and their average. For our dependent variable, we used the condition that white won as a 0 or 1. For clarity, we also plotted the theoretical win rate of white based on the Elo formula.



As we can see from the graph, these two functions are similar. From a glance, our empirical function had fatter tails than the theoretical function. However, we also found that our data is slightly translated upward, most notably when players were the same rating. Using the inverse logit of the intercept, we calculated that white's win rate sits at 0.54, which is higher than the theoretical 0.50 level. This finding supports our initial hypothesis that for a large portion of players, white's ability to move first provides a significant advantage, leading to a higher overall win rate.

In addition to the graph, we also outputted a summary statistic of the logistic regression model. As we can see from the table, every unit increase in the difference between ratings results in a decrease of the log odds of white's win rate by 0.43. Also, each unit increase in average Elo results in a decrease of the log odds of white's win rate by 0.01.

	Coefficient	Error	T-score	P-value
Constant	0.1645	0.103	1.595	0.111
Rating Difference	-0.0043	0.000	-40.5	0.000
Average Rating	-0.0001	6.37e-05	-1.865	0.062

However, the probability of drawing is not specified in the theoretical Elo rating system. By chess standards, a draw is treated as half a win (and half a loss), while in our model, we treated a draw as a loss. This explains a major portion of the discrepancy between our empirical model and the theoretical model.

Section 2: Modeling Draws and Taking User Inputs

After creating a model that predicted white's probability of winning, we also needed to calculate the probability of a draw before we would be able to take user inputs. From our analysis, we were able to confidently draw two conclusions. First, the closer two players' Elos were, the higher the chance of a draw. This is shown by regressing the absolute value of the difference of ratings against the binary response variable of whether there was a draw. The coefficient of this estimate is highly significant, with a z-score greater than -7. The coefficient of the slope parameter being negative means that an increase in the rating difference for a match corresponds to a decrease in the probability of a draw. This aligns with our intuition, because we would expect that evenly matched players would draw more frequently than two players with lopsided ratings. We explored the expected probability of a draw given the rating difference in a later section of analysis, but for this portion, it is just important to note that this relationship is significant.

Next, we concluded that draws were more likely to occur in higher levels of play. To understand this, we observe the two extremes. Among very bad players, there are many mistakes, and large mistakes allow for an opponent to turn an even board into a win instead of a draw. In contrast, most elite players rarely blunder, making it more difficult to win rather than draw. To further extend this example, when the two best chess engines play each other, they draw nearly three quarters of the time, simply because they never make mistakes and are able to foresee all the resulting possibilities of the next potential moves. With a z-score of 3.994, we found that the data supported this belief.

With our two models, we felt that we were finally ready to allow for user inputs, which could be useful for chess players and spectators to clarify a potential gap in skill between two players. Although our model only accounted for white's perspective, we are able to compute the probabilities for black by simply subtracting white's respective probability and the probability of a draw from one. We designed code to allow for these inputs, and tested it through a few examples.

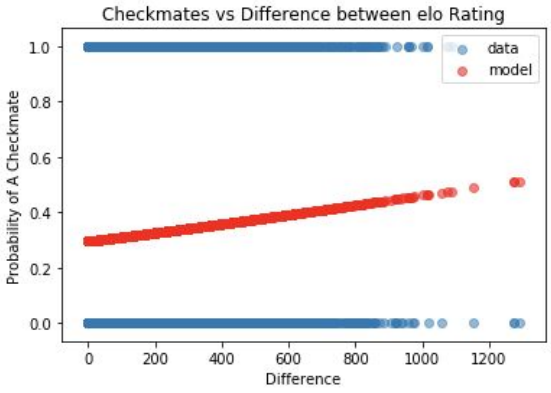
If we input white's and black's Elo to both be 1500, we get the following probabilities: white win = 49.7%, draw = 5.6%, and black win = 44.7%. This shows that though both players are equally rated, being able to move first corresponds to about a 5% increase in win probability. Next, we considered if both players were rated at 2500 Elo. Taken into context, 1500 is considered an average chess player and 2500 is considered to be a very high caliber player. This time, we get these probabilities: white win = 46.7%, draw = 9.4%, black win = 43.9%. As one can see, the probability of a draw increases substantially when the players are much better. Additionally, the gap between the probability of white winning minus the probability of black winning decreases fairly substantially between the two cases. In our last test case, we set both Elos to 800. The results are: white win = 51.7%, draw = 3.9%, black win = 44.3%. As expected, the probability of a draw is the lowest out of the three tests, and white has the biggest advantage of winning in the 800 rated match.

In our opinion, none of these results are incredibly surprising. Perhaps how big of an advantage moving first is could be something that the casual chess fan would not realize. The best explanation is that moving first quite literally allows you to be one step ahead of your opponent, which forces them to play a more defensively oriented game until they eventually catch up if they are even able to. It is also important to note that this model probably will not extrapolate well to Elos beyond 2600. The best chess players who compete professionally have Elos higher than this, and they draw at a much higher rate than our model would suggest. Our data is from an online chess playing website, so it is better suited to predict the outcome of games for the more casual player.

Section 3: Probabilities of Checkmates, Resigns and Draws

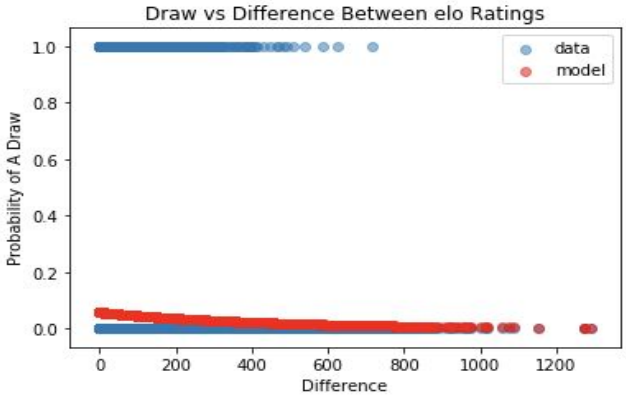
In addition to looking at which side won a chess match (or if it was a draw), we were interested in exploring exactly how matches end. We suspected that the average Elo of two players and the difference in Elos would collectively influence the probability of a match ending in a checkmate or resign. We assumed more matches would end by resignation if both players were highly skilled because a skilled player can recognize several moves in advance when the match is effectively lost. In contrast, lower-skilled players will not foresee their losses as well, or will hope that their similarly-ranked opponent will make some kind of blunder, so they will typically choose not to resign and instead play until checkmate. We also hypothesized that checkmates would be more common as the difference between skill levels increased. When the difference in rating is large, there is a low chance of a draw, and the better player is probably capable of achieving a checkmate relatively quickly. Matches ending in a draw are rare, but we assumed draws would only become even rarer as the difference in skill levels increased.

In our model, both differences in Elos and average Elo were significant in predicting the probability of a checkmate occurring. While holding average elo constant, a unit increase in the difference of ratings contributed to a 0.07 increase in the log odds of a checkmate occurring. On the other hand, holding the difference in ratings constant, a unit increase in average elo rating contributed to a 0.14 decrease in the log odds of a checkmate occurring.



In the resign model, the difference in rating was not significant in determining a change in the probability of a resign while holding average elo constant. However, average elo rating was significant. Holding the difference constant, a unit increase in average elo rating contributed to a 0.10 increase in the log odds of a resignation occurring.

Finally, in the draw model, the difference between the ratings of the players was significant in predicting a change in the probability of a draw. A 1 unit increase in difference of ratings contributed to a 0.24 decrease in the log odds of a draw occuring while holding average elo rating constant.

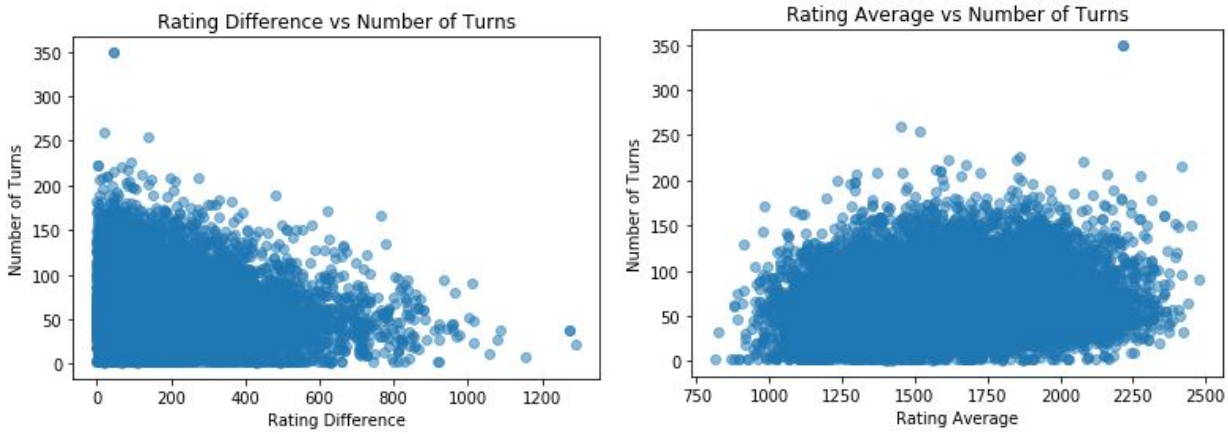


Section 4: ELO and number of match moves

After predicting the victor and outcome of matches, we also wanted to explore the number of moves that were played in a match. Knowing the expected number of moves of a match is not particularly valuable information to a chess player, which is why it is not our primary analysis, but it is still interesting to study nonetheless. From our background information on chess, we expected that the more lopsided a match was, the fewer moves there would be. If one player is much better than the other, he or she should dominate and end the match relatively quickly.

Likewise, we thought that the better the two players both were, the more moves a match should take. The idea is that worse players will make more mistakes, which result in more pieces being lost, and a quicker game. Better players make fewer mistakes because they are much more deliberate in their moves and choices.

Before running any regressions, we first wanted to simply plot the data to get a better understanding of these relationships. These plots are shown below.



Our initial expectations for what influenced number of turns in a match seem to be supported by these graphs. Particularly for the Elo rating difference, there seems to be a clear negative linear relationship between the number of turns and difference. For rating average, which is a representation of the skill level of both players, the relationship is less apparent, but we still expected some significance.

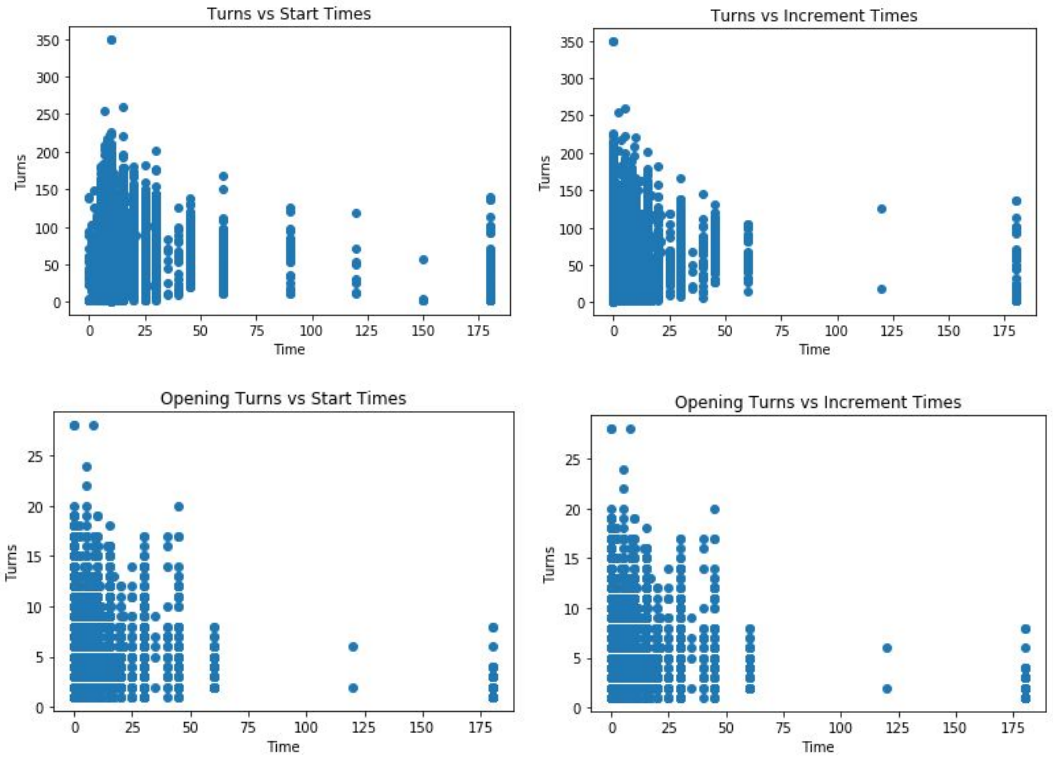
We created a multiple linear regression model using these two variables, and found them both to clearly significant by any p-value cutoff. The absolute value of the difference of the ELO ratings has a t-score of -15.6, and the rating average has a t-score of 22.4. The intercept interpretations are as follows: for a 100 point increase in rating difference between the two players, the expected number of moves decreases by 2.6, and for a 100 point increase in rating average, the expected number of moves increases by 2.2. The r-squared for this model is 0.044. Overall, the general consensus among our group is that we expected these relationships to be stronger. An R^2 of 0.044 is incredibly low, and reveals that these variables play a relatively small role in impacting the number of moves of a match. Though our two independent variables are undoubtedly significant, their impact was not as great as we anticipated.

Section 5: Effect of Time on Number of Moves

Finally, we wanted to test our hypothesis that shorter match times resulted in a fewer number of moves played. We believed that players would have a greater incentive to play more

aggressively. And in order to develop quicker, we conjectured that players would opt for shorter opening sequences that would allow them to be even more aggressive earlier on.

In order to test our two beliefs, we decided to split this into two smaller examinations. First we would test each's effect on the total number of turns. However, in order to even begin testing, we would need to separate the increment into two new columns, start time and increment time, using the '+' as the delimiter to separate the list. After creating two sets of scatter plots just to visualize the data, we can observe that they all behave similarly. The data resides heavily in the lower end, with another subsection toward the extremes, and shows a minor downward trend.



Performing linear regressions on the sets of variables, we first note that all of the regressions have very low R^2 values of 0.003, 0.002, 0.000, and 0.000 respectively, similar to the issue in Section 4 of the analysis. This implies that these factors play an even less important role. However, we cannot simply dismiss them all. The first two relationships, turns versus start times and turns versus increment times are highly significant with p-values of 1.02e-11 and 1.96e-08, so these relationships are very significant. However, the opening turns relationships are barely significant, with p-values of only 0.0693 and 0.0227, so they do not fit the data as well. Therefore, from our models, we can conclude that the time format has a significant effect on the number of total turns, but we cannot confidently conclude the same for the number of opening turns. This suggests that time can pressure players into shorter and more aggressive matches, but they may not play shorter opening sequences. This may be because the vast majority of players memorize a set of openings that they are most comfortable with playing, which they

rarely deviate from across all of their matches. And since these sequences are memorized, they can play these first turns very quickly, which may in fact be beneficial in the interest of a shorter time format.

Unit Testing

In order to perform any analysis, it was important that are supplementary functions worked accordingly. Our first set of unit test tested our functions chessprobwin and chessprobdraw to guarantee that they were outputting the correct probability. These functions are essential since they help us produce prediction outputs. Our second set of unit tests made sure that the increment_to_starting and increment_to_increment functions were accurately splitting the column of misleading increment data. This function was important, since without it, scatterplots and regression models would be very messy. Having these functions allowed our code to be more readable, which is important to reduce any potential future errors.

```
    ...: import unittest
    ...: from project import *
    ...:
    ...: class LogisticTestCase(unittest.TestCase):
    ...:
    ...:     def test_probwin_1(self):
    ...:         self.assertNotEqual(chessprobwin(1550,1450), 0.5)
    ...:         #Making sure our probabilities match how they'd be calculated on pencil an paper
    ...:
    ...:     def test_probwin_2(self):
    ...:         self.assertEqual(chessprobwin(1550,1450), 0.6014597959120163)
    ...:
    ...:     def test_probdraw_1(self):
    ...:         self.assertEqual(chessprobdraw(1550,1450), 0.04621495230703922)
    ...:
    ...:     def test_probdraw_2(self):
    ...:         self.assertNotEqual(chessprobdraw(1550,1450), 0.04)
    ...:
    ...:     def test_increment_to_starting(self):
    ...:         self.assertEqual(increment_to_starting(['1+2','3+4']), [1,3])
    ...:
    ...:     def test_increment_to_increment(self):
    ...:         self.assertEqual(increment_to_increment(['1+2','3+4']),[2,4])
    ...:
    ...:
    ...: if __name__ == '__main__':
    ...:     unittest.main()
......
----------------------------------------------------------------------
Ran 6 tests in 0.006s

OK
```

Conclusion and Future Exploration

Comparing our predicted win rate logistic function against the theoretical elo function, we note that there is a slight discrepancy between the two. However, this can be attributed to the lack of the draw feature in the theoretical version. This makes it difficult to conclude whether or not the elo system is accurate enough according to the data.

On the other hand, the rest of our results confirmed the hypotheses that we expected. For example, the majority of average elo ratings and the difference of elo ratings were significant against the probability of wins, draws, and against resigns. We also found that the difference between the ratings of two players exhibit affects the number of turns in a match through a negative linear relationship. To an even lesser extent, we also showed that the time format of a match affects the number of turns. These results all match our expected hypotheses, but to a much lesser extent than we had hoped.

In our project, we granted functionality to a general user, allowing him to input two elo ratings to find the probability of winning, drawing, and losing. This can be used to roughly predict for the average chess matches. However, chess is played very differently across different levels, so it may have been useful to split the data into categories of elo. This would result in more informative results that more accurately reflect a player's skill. However, this would have been difficult to achieve since there are many cases of a higher level player defeating a lesser player, and categorizing these players into separate bins would nullify this match, since it would not belong to either category.

Similarly, considering that a large portion of the chess community is heavily interested with the results of professional chess matches, subsetting our dataset to only include matches from high-level players adds a level of specificity that provides clearer insight. To pursue this idea, we could have also found a different dataset that focused specifically on professional chess matches to guarantee a sufficiently high number of games.

Another area of exploration could be investigating the use of other rating systems, namely the Glicko-2 system that the Australian Chess Federation has adopted. We could then compare the accuracies of the two models and determine which rating system provides a better measure of players' skill levels.