Modeling Outcomes of League of Legends Matches

Vinh Phan

12/11/2020

Introduction

League of Legends (abbreviated as LoL) is a multiplayer online battle arena game developed and published by Riot Games. In LoL, there are two teams of 5 players where the main objective of the game is to destroy the opposing team's "Nexus", a structure that is heavily defended in the center of each team's territory. Since LoL's inception in 2009, the game has experienced tremendous growth with over 100 million active monthly players in 2020. With such a large playerbase, LoL has developed an extremely active international professional scene. There are professional leagues worldwide such as in North America, Europe, China, South Korea, and many more regions for top level professional LoL. These professional leagues consist of teams that play one another in regional competitions which culminate with an annual World Championship.

The initial goal of this paper is to model win probability specifically for professional teams in the North American region. The ability to predict match outcomes between two teams from win probabilities generated by a model is naturally a valuable tool for the betting industry. A significant amount of money is usually involved and people will always search for a slight edge in order to make more accurate predictions. In addition to benefitting the betting market, a model for win probability also provides insight into the game itself by identifying elements that have the most influence in determining a match's outcome.

The second goal of the paper is to then simulate head to head matches between two professional North American teams using the win probabilities generated for each team from the model mentioned above. These simulations will follow Condorcet's model for majority voting. The original context in which Condorcet applies the majority voting model is in a courtroom where a jury must arrive at a Yes decision. There are an odd number of voters in the jury and each voter has probability, p, of voting Yes and probability, $1 - p$, of voting No. Condorcet then determines the probability that the majority of voters vote Yes. In the context of LoL matches, an odd series of n head to head matches played between two professional LoL teams will represent the number of voters in the jury. Similarly each game will have probability, p, of Team A winning and probability, $1 - p$, of Team B winning. I wish to determine the probability that Team A will win the majority of their games against Team B. For example, if there is a series of five games that the teams must play, I wish to find the probability that Team A wins the majority of these five games. I will then verify Condorcet's Jury Theorem in the context of LoL matches. This section of the paper involving Condorcet's Jury Theorem and his model for majority voting will be discussed in more detail after the creation and explanation of the win probability model.

Derivation & Explanation of Win Probability Model

A logistic regression model will be used to model win probability since the dependent observation we wish to observe, match outcome, is binary with values being either 1 (win) or 0 (loss). Thus match outcome, follows a Bernoulli distribution with unknown probability, p. The logistic regression model will estimate the value of p for any given linear combination of independent variables that we choose for our model. The independent variables chosen for our model will be the following selected data for any given match:

- towerKills: The amount of enemy towers each team destroys
- baronKills: The number of times a team killed Baron Nashor
- deaths: The total death count for each team
- gold_spent_percentage_diff: The percentage difference of gold spent between a team and the opposing team

From our chosen independent variables, we can then construct the following linear equation to estimate the natural log of the odds ratio of winning a match:

$$\ln \frac{p}{1-p} = \beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0$$

$$p = \text{probability of winning the match}$$

This equation is known as the logit in logistic regression and is derived from Bayes' Theorem. The logit is the link between the linear equation of independent variables to the dependent variable's Bernoulli distribution. The full derivation of the logit from Bayes' Theorem is as follows:

First we define the probabilities that we will work with:

$P(Y \mid X) =$ posterior probability of winning a match given input data
from our linear equation of independent variables
$P(X \mid Y) =$ probability of having certain input data values given that we won the match
$P(Y) =$ prior probability of winning a match
$P(X) =$ probability of having a certain set of input data in our linear model

The probability that we are interested in is, $P(Y \mid X)$, because this represents the win probability given any input data into our linear model. Estimating this probability is the purpose of creating a logistic regression model. To accomplish this, we solve this equation which is the formula for Bayes' Theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

We solve this equation by defining:

$P(\bar{Y})$ = prior probability of losing a match

We then rewrite our original equation in terms of the odds ratio of winning the match given input data:

$$\frac{P(Y \mid X)}{P(\bar{Y} \mid X)} = \frac{P(X \mid Y)P(Y)\frac{1}{P(X)}}{P(X \mid \bar{Y})P(\bar{Y})\frac{1}{P(X)}}$$
$$\frac{P(Y \mid X)}{P(\bar{Y} \mid X)} = \frac{P(X \mid Y)P(Y)}{P(X \mid \bar{Y})P(\bar{Y})}$$

We can then simplify this equation in terms of odds ratios:

$$O(Y \mid X) = \frac{P(X \mid Y)}{P(X \mid \bar{Y})}O(Y)$$

$O(Y \mid X)$ = odds ratio of winning a match given input data
$O(Y)$ = odds ratio of winning a match

We then take the natural log of both sides:

$$\ln{(O(Y \mid X))} = \ln{(\frac{P(X \mid Y)}{P(X \mid \bar{Y})}O(Y))}$$
$$\ln{(O(Y \mid X))} = \ln{\frac{P(X \mid Y)}{P(X \mid \bar{Y})}} + \ln{O(Y)}$$

We then have what appears to be a linear equation since $\ln O(Y)$ is a constant that does not depend on X which is our input data. This term will be treated as a constant. We then make the simplifying assumption that the natural log of the odds ratio is the linear function of independent variables that we initially created. Thus, we arrive at the equation:

$$logit(Y \mid X) = \beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0$$

We then take the inverse of the logit function and transform the odds ratio back into the posterior probability we are interested in, $P(Y \mid X)$. This is done as follows:

$$O(Y \mid X) = e^{\beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0}$$

Since the probability of winning a match conditional on some input data can be expressed in terms of its odds ratio as:

$$P(Y \mid X) = \frac{O(Y \mid X)}{1 + O(Y \mid X)}$$

We rewrite the inverse logit equation as:

$$P(Y \mid X) = \frac{e^{\beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0}}{1 + e^{\beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0}}$$

$$P(Y \mid X) = \frac{1}{1 + e^{-(\beta_1 towerKills + \beta_2 baronKills + \beta_3 deaths + \beta_4 gold\_spent\_percentage\_diff + \beta_0)}}$$

Thus, we arrive at our logistic regression model where we are able to find win probabilities given inputs into our linear function.

Independent Variable Choice & Assumptions

I chose these four independent variables:

- towerKills: The amount of enemy towers each team destroys
- baronKills: The number of times a team killed Baron Nashor
- deaths: The total death count for each team
- gold_spent_percentage_diff: The percentage difference of gold spent between a team and the opposing team

to be included in my logistic regression model because they were all statiscally significant in determining win probability and had the largest effect on win probability during model experimentation and testing. towerKills and baronKills can be explained visually in Figure 1:

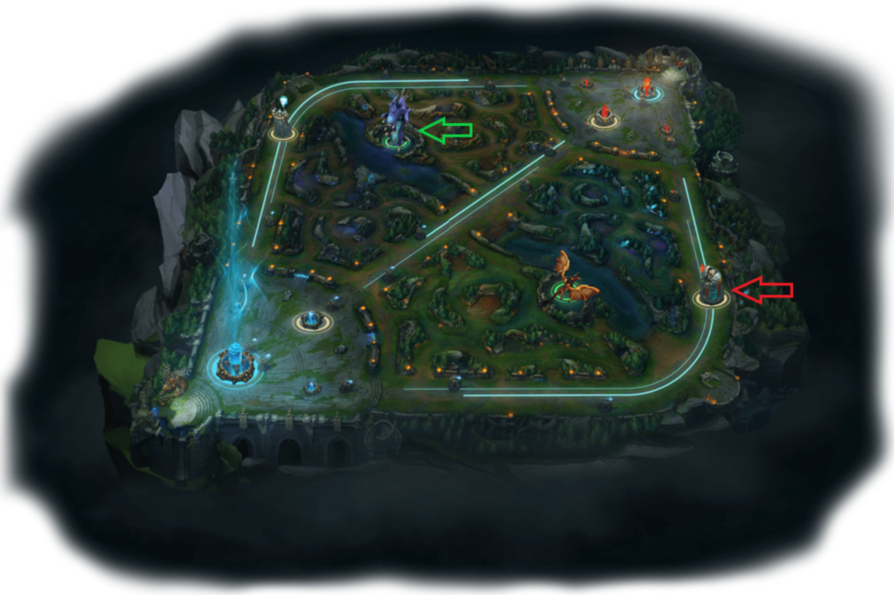


Figure 1: Map of Summoner's Rift

The green arrow in Figure 1 points to a neutral monster known as Baron Nashor. A team that manages to kill the Baron gains an advantage by becoming stronger which causes them to become more of a threat to the opposing team. The red arrow points to a structure known as a tower. Towers defend a team's territory. The more towers a team destroys, the further they can push into enemy territory to threaten to end the game by destroying the opposing team's Nexus. An important note to make about the variable, gold_spent_percentage_diff is that it is not the percentage change in gold spent. The formula for percentage difference is $\frac{\text{Team A Gold Spent - Team B Gold Spent}}{0.5(\text{Team A Gold Spent + Team B Gold Spent})}$. Percentage difference uses the average of both team's gold spending as the denominator instead of choosing either team. This method equally weighs the percent gold spent difference between the two teams. This variable will always be mirrored for opposing teams in the same match. For example, if Team A's gold spent percent difference is +15%, then Team B's gold spent percent difference will be -15%.

My claim that these variables form a linear relationship to win probability conditional on a set of inputs can be explained by observing a single variable in the linear model. If we suppose that our model only uses the variable gold_spent_percentage_diff (abbreviated as gspd) to model win probability, we can observe three distinct situations:

1. If gspd is 0%, it is reasonable to assume that an additional percent increase in gspd will be associated with some constant increase in win probability. A linear relationship is plausible in this situation.
2. If gspd is at a high value such as +90%, then the probability of winning becomes so high that an additional percent increase or decrease in gspd will not greatly affect the win probability. This is because the team with gspd of +90% is so far ahead in terms of gold and character strength that the opposing team will have nearly no chance to mount a comeback and win the game. Thus, win probability flattens out for high values of gspd.
3. If gspd is at a low value such as -90%, then the probability of winning becomes extremely low so an additional percent increase or decrease in gspd will not affect the win probability. This is similar to situation two. Thus, win probability flattens out for low values of gspd.

These three situations suggest that the win probability curve follows a sigmoid, “S”-shaped, curve. This particular curve is exactly the type of curve seen in logistic regression models. Extending this analysis to the rest of our chosen independent variables and our multivariate linear function results in the same outcome as seen in the single linear case.

A key assumption that this logistic regression model makes can be found in the linear function that is used to approximate the natural log of the odds ratio of winning a match. The linear function assumes that the independent variables are completely exogeneous, meaning that they are not affected by each other. This is quite a large assumption since it is very unlikely that all the independent variables are exogenous. It is likely that they have some influence over one another. For example, the variable, baronKills may be correlated to towerKills. This is because most teams will often use the strength boost acquired by killing

Baron as a tool to more easily destroy enemy towers. This is only one example where correlation can occur among our chosen independent variables. The endogeneity of our variables will not be a large issue if the correlation between them is weak. Table 1 displays a correlation matrix between our independent variables. The correlation between towerKills and gold_spent_percentage_diff is on the higher end at 0.703. Even though the other correlation values between the variables is not high, we should still be wary of the win probabilities generated from our logistic regression model.

Table 1: Correlation matrix of independent variables

	towerKills	baronKills	deaths	gold_spent_percentage_diff
towerKills	1.0000000	0.6056232	-0.2696844	0.7033052
baronKills	0.6056232	1.0000000	-0.0335175	0.3962897
deaths	-0.2696844	-0.0335175	1.0000000	-0.5675905
gold_spent_percentage_diff	0.7033052	0.3962897	-0.5675905	1.0000000

Data Collection & Model Results

All of the data was collected from Riot's API. Through the API, I requested match data from the top 300 highest ranked players in North America. This distinction is important because I wanted to build a model that could be accurately applied to the highest level professional teams in North America. I randomly selected one ranked solo/duo queue match from each player. I then parsed each match for potential independent variables I could use for the logistic regression model and created additional variables on my own from existing data. Each match would then give me two rows of data, one from the winning team and one from the losing team. Each row had many columns of potential variables that could be used in the model. The end result was a dataframe with 600 rows and 34 columns. The final parsed dataset used to train the model was a dataframe with 600 rows and 4 columns. The first 6 rows of this dataframe are displayed in Table 2.

Table 2: The first 6 rows of the aggregated match dataset

towerKills	baronKills	deaths	gold_spent_percentage_diff
7	0	17	0.3383997
0	0	46	-0.3383997
2	0	35	-0.2286170
9	1	16	0.2286170
7	0	16	0.2474527
1	0	34	-0.2474527

From this final dataset, I randomly sampled 300 rows to serve as the training set and set aside the remaining 300 rows to serve as the test set. I then trained a logistic regression model using the training set and applied it to the test set. The model results can be seen in Table 3.

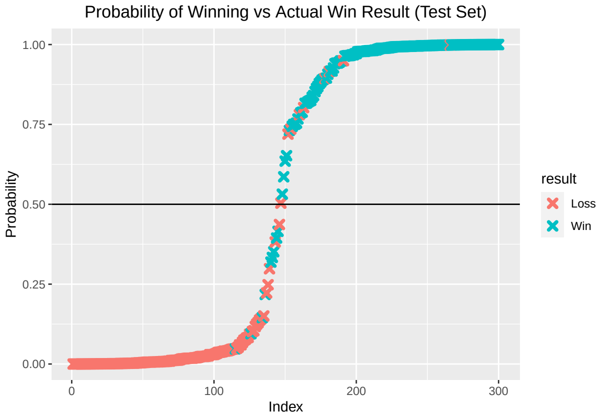
Table 3: Summary of logistic regression model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.0732250	0.8952077	1.198856	0.2305840
towerKills	0.8957712	0.1712986	5.229297	0.0000002
baronKills	1.5427027	0.5170165	2.983856	0.0028464
deaths	-0.2405705	0.0458778	-5.243718	0.0000002
gold_spent_percentage_diff	3.6704827	3.5490063	1.034228	0.3010295

Thus the finished logistic regression model can be written as:

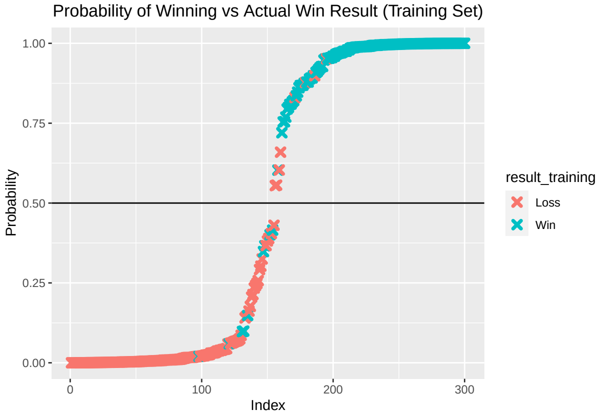
$$P(Y \mid X) = \frac{1}{1 + e^{-(0.896towerKills+1.543baronKills+-0.241deaths+3.67gold\_spent\_percentage\_diff+1.073)}}$$

The graph of predicted probabilities vs. the actual win result is displayed alongside the confusion matrix for both the test and training set in Figures 2 and 3.



	FALSE	TRUE
Loss	137	9
Win	9	145

Figure 2: Test Set



	FALSE	TRUE
Loss	146	8
Win	9	137

Figure 3: Training Set

Applying Condorcet's Model of Majority Voting

We can now generate win probabilities for different professional teams in North America using our logistic regression model. Since professional match data is not available through Riot's API, we obtain data recorded on third party websites. To generate a win probability for a professional team, we will observe a team's 10 most recent games played in North America and collect the neccessary data from each match. We then apply our model onto this dataset to obtain a win probability for each of the 10 matches. The team's win probability for a future match will then be the average of these 10 values. We repeat this process for another team of our choosing.

I will obtain the win probability for Cloud 9 (abbreviated as C9) and Team SoloMid (abbreviated as TSM). The win probability for each team after applying the above process is:

- C9 = 0.805
- TSM = 0.781

We can then use each team's win probability to simulate head to head matches and apply Condorcet's majority model to determine the probability of C9 winning the majority of the games played in a series of n games against TSM. Condorcet's model was initially used as a voting model. The simplest version of this voting model uses these parameters:

- Suppose there are an odd number of voters, $2m + 1$, where $m \in \mathbb{N}$
- There are only two outcomes on a vote, either Yes or No.
- Votes are uncorrelated between voters
- Each voter votes Yes with probability p and No with probability $1 - p$

We observe that the outcome for each vote is a Bernoulli random variable. The number of voters who vote Yes follows a binomial distribution since there are $2m + 1$ voters where each vote follows a Bernoulli distribution with probability p. The probability that the majority of voters vote Yes can be described as $\sum_{i=m+1}^{2m+1} \binom{2m+1}{i} p^i p^{2m+1-i}$. In the context of LoL, C9 amd

TSM will play a series to determine a winner. Series consist of an odd number of matches. The team who wins the majority of the matches played in the series is the winner. Thus, each match is a Bernoulli random variable where p is the probability that C9 wins and $1-p$ is the probability that TSM wins. Similarly, the number of matches C9 wins follows a binomial distribution.

Each match between C9 and TSM must be a Bernoulli random variable in order to apply Condorcet's model. We observe that this is not the case since the probabilities of C9 winning and TSM winning is greater than 1. The process for turning these probabilities into p and $1-p$ is as follows:

Let X be a Bernoulli random variable
with probability 0.805 if C9 wins
and probability 0.195 if C9 loses.

$$X = \begin{cases} 1 \text{ (C9 wins)} & \text{with probability 0.805} \\ 0 \text{ (C9 loses)} & \text{with probability 0.195} \end{cases}$$

Let Y be a Bernoulli random variable
with probability 0.781 if TSM wins
and probability 0.219 if TSM loses.

$$Y = \begin{cases} 1 \text{ (TSM wins)} & \text{with probability 0.781} \\ 0 \text{ (TSM loses)} & \text{with probability 0.219} \end{cases}$$

$$\begin{aligned} \therefore \quad & \text{P(C9 wins against TSM)} = \text{P(X = 1, Y = 0)} = 0.805 * 0.219 \\ & \text{P(TSM wins against C9)} = \text{P(X = 0, Y = 1)} = 0.195 * 0.781 \end{aligned}$$

Then, we let Z be a Bernoulli random variable with probability mass function:

$$Z = \begin{cases} 1 \text{ (C9 beats TSM)} & \text{with probability } \frac{0.805*0.219}{(0.805*0.219)+(0.195*0.781)} \\ 0 \text{ (TSM beats C9)} & \text{with probability } \frac{0.195*0.781}{(0.805*0.219)+(0.195*0.781)} \end{cases}$$

$$\begin{aligned} \therefore \quad & \text{P(C9 beats TSM)} = 0.537 \\ & \text{P(TSM beats C9)} = 0.463 \end{aligned}$$

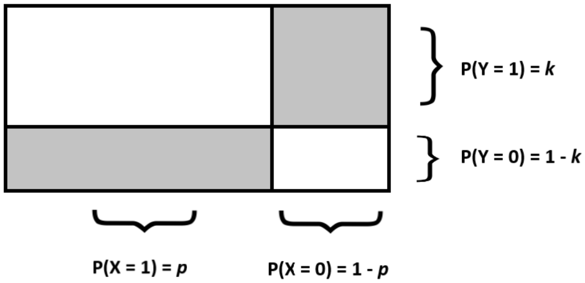


Figure 4: Intuition behind combining win probabilities

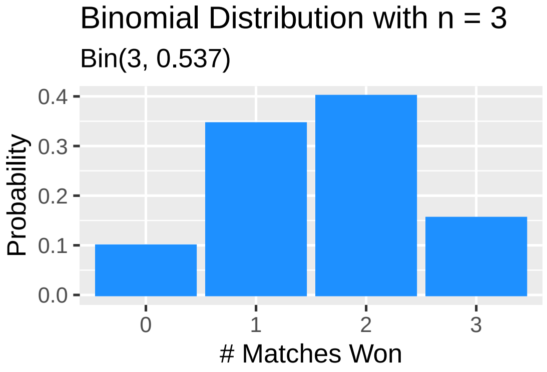
The intuition behind combining the two win probabilities is displayed in Figure 4. Let the rectangular box represent the sample space, Ω. Let the event that C9 wins a match and the event that C9 loses a match be represented by the two slices of the box separated by the vertical line. Then the respective probabilities for each event is p and $1-p$. Let the event that TSM wins a match and the event that TSM loses a match be represented by the two slices of the box separated by the horizontal line. The respective probabilities will then be k and $1-k$. Thus the probability that C9 beats TSM can be described as the probability that C9 wins and TSM loses conditioned on the fact that there is only one winner. The probability that C9 beats TSM can be described mathematically as $P(X=1, Y=0|X+Y=1)$. Similarly, the probability that TSM beats C9 can be described as $P(X=0, Y=1|X+Y=1)$. This calculation is done when we create the random variable Z as seen above.

Assumptions

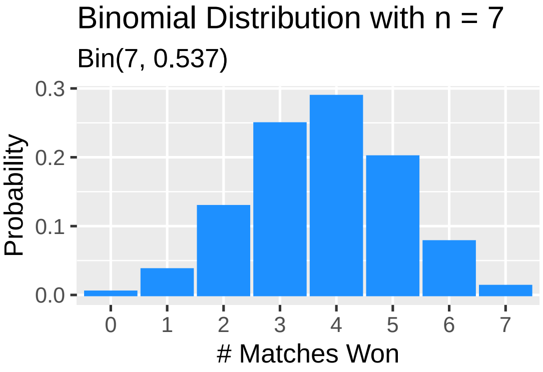
One important assumption that is made when combining the win probabilities is that the events of C9 winning and TSM winning are independent. This may not always be the case. For example, if C9 is known to lose most of their matches against TSM, the events are not independent and the combined probabilities will not be accurate. Another assumption made is that the probability that C9 beats TSM is independent and does not change throughout all the matches that are played. This assumption is similar to votes being uncorrelated between voters as proposed in the initial example introducing Condorcet's model.

Results

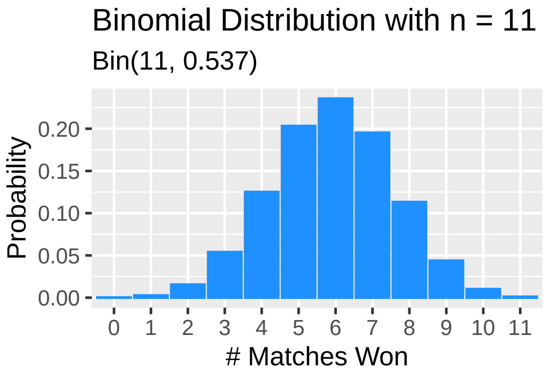
The histograms below display the binomial distribution representing the number of matches that C9 wins against TSM. The size of each binomial distribution represents a series which consists of an odd number of $2m + 1$ matches. The probability of C9 winning the series, which is the majority of the matches played, is the sum of the probabilities of C9 winning $m + 1, ..., 2m + 1$ matches. This is visually displayed with the histograms below. As the number of matches played increases, we observe that the variance decreases. Thus, the probability that C9 wins the series increases if more matches are played. This observation is a result of Condorcet's jury theorem. If the probability of C9 beating TSM is greater than $\frac{1}{2}$, then increasing the number of matches played will increase the probability that C9 wins the majority of the matches. Thus the probability that C9 wins the majority of the matches approaches 1 as the number of matches increases. Alternatively, if the probability of C9 beating TSM is less than $\frac{1}{2}$, the probability that C9 wins the majority of matches approaches 0 as the number of matches increases.


Binomial Distribution with n = 3

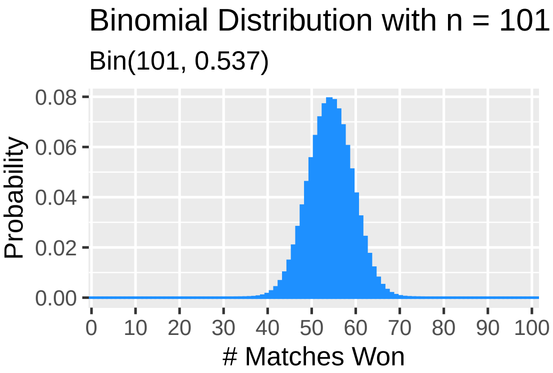
Bin(3, 0.537)


Binomial Distribution with n = 7

Bin(7, 0.537)


Binomial Distribution with n = 11

Bin(11, 0.537)


Binomial Distribution with n = 101

Bin(101, 0.537)

The empirical verification of Condorcet's jury theorem is displayed in Table 4. As the length of the series increases, the probability increases. This is because the initial probability, $P(\text{C9 beats TSM}) = 0.537 \ > \frac{1}{2}$.

Table 4: Probability table of C9 winning majority of matches for different series

Series Length	Probability
3	0.5553987
7	0.5804957
11	0.5992520
101	0.7723349

Conclusion

This project was an attempt at modeling win probability for professional North American teams with the intention of using these probabilities to determine the outcome of a series of head to head matches between two different teams. An important note to make is that the analysis done in the paper is a post-hoc analysis of the game. The logistic regression model takes in data from matches that have already occurred in order to predict outcomes of future matches. The analysis done in this paper focused on the North American region. It would be interesting to observe any potential changes to the logistic regression model with the inclusion of data from all the major regions worldwide. In the Condorcet section of the paper, data is taken from the most recent 10 matches for each team. A more sophisticated approach could include a correction factor that accounts for the strength of each opponent that a team plays. For example, if a team won their recent 10 matches against highly rated teams, their final win probability would be higher to account for this fact.

References

Bouzianis, Stephen. "Predicting the Outcome of NFL Games Using Logistic Regression." *UNH Scholars' Repository*, University of New Hampshire, 2019, scholars.unh.edu/cgi/viewcontent.cgi?article=1472&context=honors.

Hubbard, Chandler. "Esports Win Probability: A Role Specific Look into League of Legends." *Samford University Center for Sports Analytics*, Samford University, 27 May 2020, www.samford.edu/sports-analytics/fans/2020/Esports-Win-Probability-A-Role-Specific-Look-into-League-of-Legends.

Kurt, Will. "Logistic Regression from Bayes' Theorem." *Count Bayesie*, 13 June 2019, www.countbayesie.com/blog/2019/6/12/logistic-regression-from-bayes-theorem.

Quintana, Diego Angulo. "Predicting Wins in League of Legends." *RPubs*, Rstudio, 30 Aug. 2019, rpubs.com/diegolas/LogisticLoL.

Simonof, Jeffrey S. "Logistic Regression — Modeling the Probability of Success." *NYU Stern*, NYU, 2018, people.stern.nyu.edu/jsimonof/classes/2301/pdf/logistic.pdf.