My title...

Jordan R. Love

Department of Mathematical Sciences
Montana State University

May 3, 2019

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Jordan R. Love

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	Andrew B. Hoegh Writing Project Advisor
Date	Mark C. Greenwood Writing Projects Coordinator

Contents

Abstract

Abstract goes here.

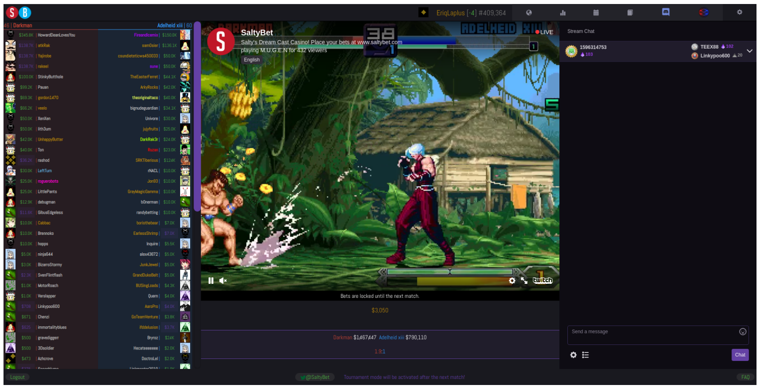
April 7, 2019

1 Background & Motivation

1.1 What is SaltyBet?

SaltyBet is an online, 24/7 nonstop, A.I. driven street fighter game where viewers are given fake “Salty Bucks” to bet on their favorite characters. [?] SaltyBet is hosted on Twitch, a popular video game streaming website where users can stream and commentate while they play. SaltyBet was one of many to creatively use the stream to foster more viewer engagement. Most notably, the popular ”Twitch Plays Pokemon” stream cites as an inspirational source for a non-standard format. [?] While an official launch date for SaltyBet is not currently known, it has been running since at least April 26th, 2013 based on the companion twitter account. [?] Since then, the twitch stream has maintained a fairly consistent average of 400 users over the past year as reported by SullyGnome, a third-party twitch statistics. [?] A typical screen during a match is shown in Figure 1.

Shortly after its inception, SaltyBet added a premium feature which al-



EriqLaplus [-4] #409,364

46 | Darkman		Adelheid xiii | 60	
$345.8K	HowardDeanLovesYou	Fireandicemix	$150.0K
$138.7K	atkRak	samOsier	$136.1K
$138.7K	Yajirobe	csundietaticwa450033	$50.0K
$138.7K	rakeel	sune	$50.0K
$100.0K	StinkyButthole	TheEasterFerret	$44.1K
$99.2K	Pauan	ArkyRocks	$42.0K
$69.3K	gordon1470	theoriginaltaco	$40.0K
$66.2K	vealo	bignudeguardian	$34.1K
$50.0K	XenXen	Univore	$30.0K
$50.0K	lith3um	jujyfruits	$25.0K
$42.0K	UnhappyButter	DarkRak3r	$24.0K
$40.0K	Ton	Ruzan	$23.0K
$36.2K	rashod	SRKTiberious	$12.4K
$30.0K	LeftTurn	rNACL	$10.0K
$25.0K	roguerobots	Jon93	$10.0K
$25.0K	LittlePants	GrayMagicGamma	$10.0K
$12.9K	debugman	b0nerman	$10.0K
$11.6K	GibusEdgeless	randybetting	$10.0K
$10.0K	Cabbac	boristhebear	$7.0K
$10.0K	Brennoko	EarlessShrimp	$7.0K
$10.0K	hopps	Inquire	$5.5K
$5.0K	ninja644	alex43672	$5.0K
$3.0K	BizarroStormy	JunkJewel	$5.0K
$2.3K	SvenFlintflash	GrandDukeBelt	$5.0K
$1.0K	MotorRoach	BUSingLoads	$4.3K
$1.0K	Verslapper	Quern	$4.0K
$708	Linkypoo600	AeroPro	$4.0K
$671	Chenzi	GoTeamVenture	$3.8K
$625	immortalityblues	ifddelusion	$3.7K
$500	gravediggerr	Bryroz	$2.4K
$500	3Dsoldier	Hecateeeeeee	$2.0K
$473	Azhcrove	DoctroLel	$2.0K

SaltyBet

Salty's Dream Cast Casino! Place your bets at www.saltybet.com

playing M.U.G.E.N for 432 viewers

Bets are locked until the next match.

$3,050

Darkman $1,467,447 Adelheid xiii $790,110

1.9:1

Stream Chat

1596314753 103

TEEX88 102

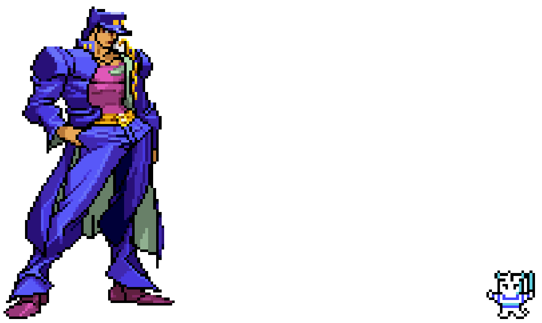
Linkypoo600 20

Tournament mode will be activated after the next match!

Figure 1: Typical screen at SaltyBet.com of a match in progress – The left vertical bar shows the users who have bet on the current match alongside their bet amount, the center vertical bar contains the match in progress and match odds, the right vertical bar shows the chat where users communicate.

lowed users to access all previous match data. This spawned the creation of many viewers opting to scrape and then use this data to build bots which automatically bet on characters. [?, ?] These bots consist of many different types of algorithms. One of the more popular bots applies a genetic algorithm to determine which character within a match will win. [?] Among the bots programmed to predict winners, none found apply a probabilistic approach to modeling characters latent ranking or incorporate immutable features of each character.

Each character consists of several features or traits which define the performance of the character. SaltyBet operates off of the MUGEN engine which was developed by "elecbyte" in early 2002. [?] This engine clones the basic features of classic Street Fighter series of games originally developed by Capcom. The engine also comes with specifications in order to create customer characters. Since then, a large number of characters have been created by users to use and compete within the MUGEN engine. SaltyBet itself employs 5,777 unique characters during its matches. Each character must consist of a set of images which define the characters movement and moveset. A moveset describes all of the actions and motions a character can make to attack another character. The image used to define a character defines its hitbox: the area on the screen where a character can receive damage. Several example characters are shown in Figure 2. Notice specifically the large discrepancy in hitbox size between many of the characters. Each character is also equipped with an AI script which defines how the character will attack and respond to other attacks. Finally, there are numeric values describing the attack strength, health, and meter (a measure of accessiblity to highly performant attacks) associated with each character. Within the SaltyBet community, there is a large amount of anecdotal evidence indicating hitbox advantage can be highly predictive of match outcomes. [**?**]



(a) 69 x 117 pixel hitbox

(b) 18 x 18 pixel hitbox

1.2 Bayesian Methods

The goal of this paper is to develop a probabilistic framework for evaluating the latent strength of characters and include additional information about the characters to improve predictive accuracy. Specifically, a focus will be given to identifying hitbox disparities between characters through this model to determine if this is a substantial predictor and at what level of the disparity does it arise. To do this, we will perform a brief review of Paired-Comaprison (PC) Models and extensions for modeling additional information.

It is of interest in this project to employ a Bayesian framework for a number of reasons. Bayesian methods often have large computational cost depending on this complexity of the model, but there are also many advantages. The advantages relevant to this project are 1. a natural estimation of variance, 2. a natural recursive formula for online estimation, and 3. a natural integration into a Markov Decision Process.

1.2.1 A Natural Estimation of Variance

While it is beyond the scope of this project to detail the differences between Frequentist and Bayesian methods, we will discuss specifically the advantage of estimation of variability. One of the primary differences between Bayesian and Frequentist methods is that the parameter under estimation is assumed to be random as opposed to fixed. This admits the workflow of a Bayesian modeling problem to be interpretted, through use of Bayes' rule, in terms of probability distribtuions throughout. While not always trivial to obtain the resulting posterior distribution of a parameter within a complex model, all information surrounding the parameter is contained within the samples of the posterior distribution. This includes variability which is simply a transformation of the samples of the posterior distribution.

Within this and similar contexts, the interest is not only in predicting a point estimate describing which characteror sports team is estimated to be most likely to win but also quantifying our uncertainty around the estimate. This is not unlike financial markets where we are not only concered with the overall performance but also the potential risk for large swings within the data. Since one of the ideal applications of this model would be to form a "trading strategy", Bayesian modeling allows access to all necessary information to make an informed decision.

1.2.2 A Natural Recursive Formula

Another advantage of Bayesian modeling is the natural recursive formula which can be formed by using the posteriors estimated at time $t-1$ as the prior distributions at time t. Many other statistical and machine learning techniques do allow this recursive estimation procedure to be implemented easily. Most notably, Recursive Least Squares, an estimation technique found commonly in Signal Processing possesses a similar framework. [CITE] However, as with many other estimation techniques, Bayesian methods typically encompass a larger class of estimation procedures with specific prior choices being commonly known in literature (Regularized Least Squares, LASSO)

[CITE]. This is no different for Recursive Least Squares with D.S.G. Pollock describing the relationship from a bayesian perspective. [CITE]

1.2.3 Natural Integration with Markov Decision Processes

Markov Decision Processes (MDPs) are a formulation of the problem of the optimally moving around a space given some set of actions and known rewards. This optimization problem is often intractable to specify aprior due to the unknown rewards associated with each action and lack of observability of the entire system. This leads to a class of MDPs known as Partially Observable Markov Decision Processes (POMDPs) where the estimation of the current state of the system and optimal policy are estimated simultaneously. POMDPs are notoriously difficult to solve optimally, but the framework allows for many numerical approximation methods to be applied. This framework can be seen in many contexts such as Air Traffic Control, Surveliance, and Robotics. [].

Since MDPs and POMDPs are both built first mathematically on the theory of probability, Bayesian modeling provides a seamless integration into these methods for two reasons. These two methods have already been described in previous advantages. The first is that we have a natural interpretation of variability in the language of probability constructed directly into our estimation procecure, the second is that the goal of both MDPs and POMDPs is typically not to make a single decision but to make multiple decisions over time. This allows the natural recursiveness of bayesian methods again to seamlessly integrate with this method. For this reason, Bayesian methods are often the de-facto estimation procedures when dealing with these problems.

While implementing a POMDP is beyond the scope of this project, this framework was an important deciding factor in what model to choose to use as a natural extension from estimating the latent strength of characters and predicting outcomes is to operatialize and optimize both which bets to make and how much to bet based on the information available.

TODO: Need lots of citations.

1.3 Data Collection

For this project, historic matches were scraped from the SaltyBet website through the premium functionality provided. There were a total of TODO: X matches between Y characters. The scraping script was written using Python and the Selenium module [X], the code appendix contains code for scraping each match and character. Due to the restrictions placed on the number of server calls each user can make to the SaltyBet server per hour, scraping for the entire dataset took approximately TODO: Z days. Once scraped, the data were formatted and into a PostgreSQL database.

Within saltybet, characters are divided into five distinct tiers: X, S, A, B, and P. These tiers are assigned based on the performance of each character previously. Characters are promoted or demoted based on their performance directly after a match. There are three distinct types of matches: Matchmaking, Tournament, and Exhibition. The matchmaking mode algorithmically chooses players to match up against each other where the odds are approximately equal of each character winning. Tournament mode is a random set of 16 characters from a specific tier who fight each other in a single-elimination tournament. Finally, exhibition mode is a set of viewer-requested matches which also allows teams of characters to compete. Exhibition mode games are typically chosen by viewers to force edge-case behavior of the characters. Many times, viewer requested matches result in a server crash due to intense computational loads.

Characters were web-scraped individually and stored with the following information: Author, Life, Meter, Hitbox Width, and Hitbox Height. The scraping process for characters was similar to the process for scraping match data and the code appendix contains a modified script for character data scraping. Of all the characters scraped, TODO: X had no image associated with their profile. For the purpose of this project, we excluded these characters. The number of matches affected by this removal is TODO: Z. Figure 2 showed an example of differing hitboxes, but

2 Models

2.1 Thurstone-Molster Model

The goal of Paired-Comaprison (PC) Models are to quantify the probability of a subject choosing one option over another or, in sports, one team winning over another. Each PC model has some formulation of the latent ”ranking” either of strength of team or underlying preference. The general form of a PC model has the form of equation 1. In this case, $f(x)$ represents a transformation of the structure of the latent strength.

$$P(A > B) = f(x)$$

One of the first models to address these issues was the Thurstone-Molster model. In this model, the underlying strength or preference is modeled as a normal distribution. Thurstone and Mollster derived several simplifications of this model to ease computation. One of the most computational simple choices is Case V of the Thurstone and Mollster Model. In this model, variances are assumed to be equal and and the strength of each character is assumed to be independent of one another. Use of the normal distribution allows for us to derive a theoretically simple equation for the paired comparison.

$$A \sim N(\mu_a, \sigma_a), B \sim N(\mu_b, \sigma_b)$$

$$P(A > B) = P(A - B > 0) = A - B \sim N(\mu_a - \mu_b, \sigma_a + \sigma_b) = \int_0^\infty$$

Equation 5 shows that in order to compute the probability of interest with these underlying assumptions, we must compute the ”error” function. This function is well known in probability and statistics, but no closed form solution exists. As this method was developed during a time when computational intensive operations were more expensive, an alternative was required for efficient use a paired-comparison model. For those interested, a Gibbs Sampling approach to the Thurstone-Molster model is described in Yao. [CITE]

2.2 Bradley-Terry Model

2.2.1 Original Bradley-Terry Model

The next development within Paired-Comparison Models was the Bradley-Terry model developed by TODO:. This model changed the underlying strength or preference distribution to be distributed as a Gumbel or Extreme Value Distribution. Using this assumption, a concise result is obtained through an similar process for the Thurstone-Mollster model. Using the CDF method to compute the difference between the two Gumbel distributions:

$$P(A > B) = P(A - B > 0) = A - B \sim Logistic(\mu_a - \mu_b, \beta) = TODO:$$

This reduces to a convient logistic function which can be evaluated more quickly than the required error function evaluation of the Thurstone-Mollster model. While these models make differing assumptions regarding the underlying strength or preference distribution, the results are neglible [TODO:

Reference from Glickman]. Figure 3 shows a comparison of shapes of the CDFs. The logistic distibution has slightly longer tails than the normal distribution and is more sloped at zero than the normal distribution. For the purposes of this paper, we will focus on the Bradley-Terry formulation of the Paired-Comparison model.

Often, a simplification is made to equation 2 in order to make the probability more interpretable. If we assume that the score function $e^{\mu_a} = \lambda_a$, then we can re-write the probability statement in a more succinct form. Equation 3 can now be interpretted as the fraction of strength competitor A consists of compared to the estimated combined strengths of competitors A and B.

$$P(A > B) = \frac{\lambda_a}{\lambda_a + \lambda_b}$$

2.2.2 "Home-Field Advantage" Model

Agresti in his 1990 Categorical Data Analysis included a modification to the original Bradley-terry model to include a "home-field advantage" effect. This effect is well-known in many sports as playing a role in the probability that a team will win. The effect works be introducing a constant term θ which acts as multiplier which captures the effect of the "home-field advantage". The resulting probability statement of interest is shown in equation 4. The multiplier is attached to the estimated latent strength of whichever team is currently playing at home and can be reversed easily depending on which direction probability we are interested in.

$$f(x) = \begin{cases} x^2 & : x < 0 \\ x^3 & : x \geq 0 \end{cases}$$

Within our project, we will use this model to estimate hitbox advantages. Specifically, we provide a hitbox advantage indicator to the character with the lower hitbox height of the two characters currently playing. Within this model, there is no simple way to account for hitboxes of the same height. Another potential issue is that we may not observe a hitbox advantages for small differences in hitbox heights. To account for the previous two issues, we introduce a third model which has been developed before.

2.2.3 “Conditional Home-Field Advantage”

The model we are considering here only provides an advantage multiplier when necessary. This has many applications both in the current project and within sports. Within sports, we may consider many games which are played within at a neutral site. In this case, we would expect neither team to have a “home-field advantage”. However, we would like predict the outcome of these games using data where some games were played at a non-neutral site. Within the context of this project, we can only assigned a hitbox advantage indicator if the difference in hitbox height is beyond a certain threshold. The resulting new probability statement is a combination of equation 3 and 4 where equation 3 is used if no advantage is present and equation 4 is used if an advantage is present.

2.3 Model Estimation

2.3.1 Maximum Likelihood Estimation

The original algorithm for estimating probabilities of paired comparisons was developed before Thurstone formulated a model in full. Zermelo in 1929 developed and proved an algorithm which converged to a unique set of parameter estimates given certain conditions were met. These conditions are discussed more fully in section X (TODO:) as an analysis of the comparison graph. The algorithm described by Zermelo is described below:

TODO: Algorithm Box

TODO: Discuss Algorithm Intuitively and results and connection to MLE

2.3.2 Minorization-Maximization Algorithms

While the algorithm described by Zermelo handles the most basic case, it was not extended to consider covariate models or "home-field advantage" models. In this case, a more general theory of estimation algorithms surrounding Bradley-Terry models were developed. Lange,Hunter and Yang (2000) showed the algorithm developed by Zermelo is a specific case of a more general class of algorithms known as Minorization-Maximization (MM) Algorithms. One well known special case extending from this class of algorithms is the Expectation-Maximization (EM) Algorithm. Heiser (1995) TODO: describes in detail the connection between the MM and EM algorithms.

Since the algorithm proved by Zermelo is a special case of MM algorithms, the estimation procedure only changes in notation between the original and latest literature. Instead, we provide the MM algorithm estimation procedure for the "home-field advantage" model discussed previously and note the key differences.

TODO: Algorithm Box for MM Algorithm for "home-field advantage"

TODO: Discuss key differences

2.3.3 Bayesian Estimation Methods

While a review of classical statistical methods have been review previously, the goal of this project is to develop a Bayesian view of Bradley-Terry models with the intent for these models to be used in conjunction with the mathematics of Markov Decision Processes. For this, we refer to the work of Caron and Doucet TODO: add reference. Their paper "Efficient Bayesian Inference for GeneralizedBradley-Terry Models" describes the construction of Gibbs samplers for both the typical comparison and "home-field advantage" models through the reformulation of the likelihood function via different latent variables.

In the original case, the latent variables of interest are the latent strength distributions which are denoted as λ_i in the above discussion. Instead of introducing this model, Caron and Doucet instead introduce the latent variable $Z_{i,j} = min(Y_{kj}, Y_{ki})$ where each Y_kj, Y_{ki} is a realization from the underlying strength distribution of competitors indexed by i and j, respectively. This allows the difference between two competitors to be captured in the new latent variable $Z_{i,j}$ as opposed to seperate latent variables λ_i and λ_j. However, by properties of the exponential distribution, we have that:

$$Z_{i,j}\ Gamma(n_{ij}, \lambda_i + \lambda_j)$$

TODO:

3 Comparison Graph Connectivity

3.1 Comparison Graph Definition

One key assumption when using Bradley-Terry models for paired comparisons is the data is in such a format the comparisons of interest can be estimated. In many cases when a paired-comparison model is employed, all items being compared have been compared to each other at least once and in each comparison at least one has been chosen over the other. If we consider professional sports as an example, each team typically plays most other teams at least once and some teams multiple times. This leads to many connections among a relatively small number of teams. The probability of the comparison graph being appropriate for modeling is high. However, within many college sports the number of teams is far larger than the number of matches any one team will play during a season. This leads to a much lower probability of the comparison graph of all college teams being connected. One efficient way to perform both diagnostic check snad validate these assumptions is through graph theory.

A first principles view of graph theory is beyond the scope of this paper, but the author refers interested readers to Chartrand [**?**]. A graph is defined as a collection of nodes and edges. Nodes represent atomic items such as sports teams or choices where edges may represent matches between teams or preferences between choices. A graph can be an undirected or directed graph. In the case of an undirected graph, a single edge is placed between two nodes indicating a general relationship. Within an undirected graph, consider moving between two nodes whenever any edge connects them. A directed graph adds an additional layer for direction. For the purposes of this paper, a directed edge from between two nodes indicates the outcome of a match where the direction will extend from the winning node to the defeated node. A directed graph restricts movement between nodes to movements allowed by the directed edges.

Formally, we will define a graph as an object consisting of two sets: nodes and edges. We will index the nodes as $1 \leq i < j \leq k$, where k represents the number of nodes in the graph. We will refer to edges as $e_{i,j}$, where this represents the directed edge from i to j unless specified we are working in an undirected setting in which case $e_{i,j}$ and $e_{j.i}$ are interchangeable. We will define a **comparison graph** as the directed graph consisting of all characters as nodes and all matches played between each character as directed edges. Figure X shows a sample comparison graph.

3.2 Condition of Strong Connectivity

The original constraint for paired-comparison data to be suitable for analysis such that all comparisons can be estimated was formulated by Ford in 1957 [?]. In graph theoretic terms, the condition can be equivalently stated as the comparison graph constructed from the data must be **strongly connected**. This condition necessitates that within a directed graph any node can be reached by starting at any other node in the graph and moving along directed edges. This is a stronger statement than simply saying a undirected graph be connected as the "one-way streets" formed by directed edges may not necessarily form a strongly connected graph.

In order to evaluate the feasibility of this constraint within a dataset of paired-comparisons, we can use well known algorithms to determine the connectivity of the comparison graph. Tarjan's Algorithm by Robert Tarjan is an efficient algorithm which runs in linear time and determines the number of strongly connected components within a graph [?]. This algorithm can serve two purposes within the context of paired-comparison models. If the interest is in determining if the comparison graph is strongly connected, then the desired output from this algorithm is that the graph is constructed from one strongly connected component. However, in the event there exist more than one strongly-connected component within the graph (implying the entire graph is not strongly connected), the algorithm will return labels corresponding to a disjoint set of nodes which are strongly connected themselves. In this case, a paired-comparison model can be fit to each of the strongly connected components identified, but comparisons across the strongly connected components will be unavailable. An example of a strongly connected graph can be seen in Figure X while an example of a graph with multiple disjoint strongly connected components can be seen in Figure Ya.

3.3 Condition of Weak Connectivity

While strong connectivity of the comparison graph allows comparisons without modification, Yan (2014) discusses how using a singular pertubation method can allow for comparisons to be made when the graph is only **weakly connected.** Weakly connected graphs are directed graphs whereby if all directed edges are converted to undirected edges, the graph is connected or all nodes can be reached from any other node. Figure Yb shows an example of this situation in detail. In Figure Ya, we see the original directed graph with node A having only directed edges away from it. This can be known in other graph theoretic literature as a ”source.” However, if we remove the direction from each of the edges, we see in Figure Yb that the undirected version of this graph is connected. One intuitive reason for why weakly connected graphs are not sufficient without modification is that one team is undefeated and without an additional measure of ”strength of win”, the magnitude of how much the latent strength of this team differs from the next best team is unknown.

The modification discussed by Yan is by adding a penalized term to the likelihood in such a way that a ”pseudo-loss” to at least one other team is added to the comparison graph in order for the graph to become strongly connected. [?] This term can be re-interpreted as a specific prior distribution to allow for the penalized term to exist within the likelihood.

3.4 Simulation of Graph Connectivity

In the case of the data we are analyzing, we assume that two characters are randomly chosen to participate in a match. This is an assumption of the underlying structure of the data generating mechanism which is unavailable for us to view. Unfortunately, the problem of detecting "densely" connected clusters within a graph is computational challenging [CITE]. Instead, we will assume that of the 9,662 characters present, two characters are randomly chosen and then compared within a match. For our simulation, we will allow the order in which the characters are chosen to correspond to the direction of the edge between each character. While this does not account for the underlying strength of a character, it provides a general enough framework for an idea of connectivity within a specific graph size.

The simulation begins by generating 9,662 nodes and then randomly generating edges between them. The number of edges generated varies from 10,000 to 1,250,000 in steps of 500 with 50 simulated comparison graphs being generated at each step. We then apply Tarjan's Algorithm to determine the number of strongly connected components within each graph. In parallel to this experiment, we also measure the number of weakly connected components within each graph and iterate across the same graph properties as the previous experiment. The results of this simulation are shown in Figure Z and Table A below.

4 Data Analysis

4.1 Data Cleaning

With the major components modeling, estimation, and diagnostic of this project discussed in detail, we begin to discuss the analysis of the SaltyBet data collected. First, we note some data cleaning steps. As discussed previously, SaltyBet consists of three phases: Matchmaking, Exhibition, and Tournament. Within the Exhibition mode teams of two characters are allowed to be requested by viewers. Since the data does not specify the makeup of each team, we remove matches which contain any team of characters. This accounts for only XYZ% of the total number of matches. In addition, while ties are both permitted by SaltyBet and literature exists to incorporate this into the Bradley-Terry models, we remove them from this dataset for a number of reasons. First, ties are very unlikely within SaltyBet consisting of only XYZ% of the data (XYZ matches). Second, if a tie does occur, it is typically due to either server failure or some unique attribute of the match up of characters (consider two identical characters being matched).

4.2 Exploratory Data Analysis

After data cleaning, our data consists of a total of XYZ matches from a total of XYZ characters. Of all matches, XYZ characters have no losses with XYZ characters have no wins. The mean and median number of matches from each character is XYZ and XYZ, respectively. In addition, only XYZ% have ever faced the same opponent twice. Of the characters who do face a repeat opponent, XYZ% had at least one of the two matches requested in an exhibition match indicating that viewer intervention is the primary cause of repeat matches.

For hitbox data, the median hitbox height was XYZ pixels while the median hitbox width was XYZ pixels. Figure Z1 shows a density plot of each characters hitbox height and width. Of the XYZ characters within the cleaned dataset, XYZ% fall within 10 pixels of the median height and width.

5 References - Left in for... reference

References

[1] Ella Kaye and David Firth. *BradleyTerryScalable: Fits the Bradley-Terry Model to Potentially Large and Sparse Networks of Comparison Data*. R package version 0.1.0. 2017. URL: `https://CRAN.R-project.org/package=BradleyTerryScalable`.

6 Appendix-R Code

```
All that sweet sweet R code.

Verbatim would also work here I think.
```