PHSX815 - Project 2

Ryan Scott

March 22, 2021

1 Introduction

This project is meant to study the likelihoods associated with a particular card game based on random chance. The game utilizes a deck of N_{cards} cards, distributed among two players, played against each other. The higher valued card designates the winner. We utilize some weighted and unweighted Bernoulli distributions (the card values) which affect a categorical distribution of number of wins per game (N_{wins}). We study the effects of varying the number of cards (N_{cards}) in the deck and the number of games in each set (N_{games}). We also utilize a variable named gimme, represented by G, which allows one of the players to use cards that are guaranteed to win. All of this is studied over a fixed number of sets ($N_{sets} = 500$). Our test statistic is β (related to the power of the test, $1 - \beta$) from the log-likelihood ratio (LLR) comparing the null hypothesis, $\mathbb{H}_0$, a fair game of $G = 0$ with $\mathbb{H}_1$, a "cheated" game with nonzero G. We use a 95% confidence interval ($\alpha = .05$) for all experiments.

The goal: Study how β is affected by varying N_{games}, N_{cards}, and G values.

We first describe how the game works in **Game Mechanics**, then we take a look at the algorithms and calculations used in the study in **Algorithms and Calculations**. Next, the analysis begins when we discuss how our variables affect the test statistic in **Studying the Variables**. Finally, we conclude with a summary of results in **Conclusion**. Figures are located at the end of the document.

2 Game Mechanics

The standard game has $N_{cards} = 16$. The 16 cards are randomly dealt to two players, A and B. The two players then simultaneously show the top card of the deck and the player with the high value card records a win. After all cards are played ($^{N_{cards}}/_2$ rounds). The wins associated with player A are recorded for analysis. This constitutes one game. Multiple *games* make up one *set*.

The game is fair when $G = 0$: the cards are randomly distributed. We can modify this value, allowing player A to use unfair cards. If a normal deck of 16 cards are valued 1-16, the cards for the gimme deck are valued at 17+, guaranteed to win. When $G > 0$, the player randomly loses a number of cards equal to G (in order to keep the number of cards in their own hand constant), guaranteeing that the cheater cards are played.

3 Algorithms and Calculations

We use two python scripts to accumulate and analyze data. The first is CardgameSim.py. As it's name suggests, it is used to simulate the game and iterate over a number of games and sets from a given input. The user also has the ability to set the number of cards, gimme value, and name of the output file. When run, this script will shuffle all of the cards, randomly distribute them between two players, and return a value from 0 to $N_{cards}/2$. This is determined quasi-randomly based on how the cards were dealt and on the G value. If it returns 0, player A did not win any games; if 8, player A won 8 games, etc. When G is nonzero, player A receives G cards that are guaranteed to win. This suggests the returned win value for cheated games must be G or greater. Our distribution of wins shifts from [0,8] to [2,8], if $N_{cards} = 16$ and $G = 2$.

In our null hypothesis, we have $G = 0$, meaning that the game is fair. Because the cards are distributed randomly, the probability of winning a round in a game is $p_0 = .5$. When we have nonzero G values, this probability changes and the distribution is skewed to the right. The probability is then given by $p_1 = \frac{2G}{N_{cards}} + .5(1 - \frac{2G}{N_{cards}})$. The first expression gives the number of games that will win based on G, the second expression gives a standard $p = .5$ probability that the rest of the games will be won.

The number of hands won per game is listed in the designated text file as a single value. We then separate each game with a comma (,) and each set is a new line of data. The text file is then composed of N_{games} rows by N_{sets} columns.

Additionally, this script will also save the user designated variables in a text file as "rules_userinput.txt". That file contains four lines of data that indicate the values for N_{cards}, N_{games}, N_{sets}, and G, respectively. This is recorded for the analysis script: CardgameAnalysis.py.

We use CardgameAnalysis to read the data from two text files, organize it, and return plots of interest. This script first gets all of the text file data (iterated games and the associated rules) and converts it into data that we can manipulate and study. The script has four plots available, 3 unique: the Frequency plot, the Density plot, and the Log-Likelihood Ratio (LLR) plot. The fourth plot returns a side-by-side of the Density and LLR plots - this is default.

The first plot of interest is the simplest: the frequency plot. This is created by taking all of the number of games won and plotting them in a histogram. The y-axis reports the associated frequency for any integer value of the x-axis, which reports the number of games won. More about the interpretation of data is given below, but we end up with a roughly normal curve.

The second plot gives density. This will appear nearly identical to the first, except that we are using the calculated density of the plot, where the sum of all data is equal to 1.0 (area under curve equals 1.0).

Third, we see our LLR. This is calculated as $\lambda = \log(\mathcal{L}_1/\mathcal{L}_0)$, where $\mathcal{L}_0$ and $\mathcal{L}_1$ are the likelihoods for the null and alternative hypotheses, respectively. These are each calculated as $\mathcal{L}_x(\alpha) = \mathcal{L}(\alpha|x) \propto P(\alpha|x)$, where α are the parameters set for the hypothesis and x is the data. We first use all of the data to establish probabilities of each hypothesis, and then evaluate the data for each set given by those probabilities. This experiment does not have a simple expected distribution of data, so it is all determined experimentally. Once the probabilities are established for each game in a set, we multiply those probabilities to get our first likelihood $\mathcal{L}$. Then we can look at the same data and evaluate a second likelihood assuming *the other probability distribution is true.* Dividing these and taking the log gives us our LLR.

We use the likelihoods because they are an important part of developing the uniformly most powerful test statistic, as indicated by the Neyman Pearson Lemma for testing simple (point) hypotheses. We can determine a confidence interval we want to use which gives a particular α value. In our case, we use 95% confidence, so $\alpha = 1 - .95 = .05$. This means that we are only willing to accept the alternative hypothesis over our null hypothesis if the data lands in the farthest 5% of data.

In our script, we normalize the data between 0 and 1 and look for where the data passes the 95% confidence level, we can record the index of that data point and use that to plot a vertical line at $\lambda = \lambda_{crit}$. That data point is then given to our alternative hypothesis, where we evaluate how much of the data falls prior to that λ_{crit} value. We do this in the reverse way that we used α by finding the index of the alternative hypothesis data corresponding to λ_{crit}, then normalizing the data, and determining our β value from that indexed location. This β is used to then calculate the power of our test, $power = 1 - \beta$. Our plot does not record the power, only the β value. As β varies between 0 and 1, we see that when β approaches 0 our test becomes stronger, meaning that it is "easier" to determine which hypothesis we are sampling from. As β approaches 1 we see that our distributions of data become less discernible. If $\beta = 1$, then all of our alternative hypothesis lies within the 95% confidence range.

4 Studying the Variable: N_{games}

First, we look at what happens to our *beta* value as the number of games varies. We hold all other variables constant:

$N_{cards} = 16$ $\quad$ $N_{sets} = 500$ $\quad$ $G_0 = 0$ $\quad$ $G_1 = 2$

We vary N_{games} as 1, 2, 5, 10, 25, 50, and 75 games per set and analyze results. These are demonstrated in Figures 1-7 in the appendix. These each show two plots: on the left is the probability density of the number of wins per game over all 500 sets; the right shows the log-likelihood ratio vs probability. The vertical line represents the critical value, $lambda_{crit}$ described in **Algorithms and Calculations** given by α.

First, we see that the density plot is not affected by the number of games played. This is expected: we wouldn't anticipate more games changing the probability of winning, we only expect to see more wins and more losses.

If we look at the LLR plot, things look a little different. At a low number of games, we see large β

values, meaning that we cannot easily distinguish between the two hypotheses. As we increase the number of games per set, we see that the β value reduces until at 50 games per set and 75 games per set we have $\beta = 0.0$, and we can easily deduce which hypothesis we are testing from.

5 Studying the Variable: N_{cards}

Next, we look at what happens when we modify N_{cards}, again keeping all other variables constant. This is performed similar to the previous case of N_{games}, except that we hold N_{games} at 25 and vary N_{cards} from 12, 16 and 20. When we use more or less cards than this, our analysis script begins to break down. For this data, refer to Figures 8-10 in the appendix.

This data shows that at a low value of $N_{cards} = 12$, our β value is higher than when we use more cards. Interestingly, the β value for $N_{cards} = 16$ and $N_{cards} = 20$ are not easily distinguishable. Unfortunately, larger values were not able to be tested because the analysis script would encounter errors.

This is all as expected though. More cards means a wider density plot. This is because the maximum number of wins per game is increasing. Also, when we reduce the proportion of G to N_{cards}, our data begins to look more identical because the percentage of guaranteed wins is reduced. This is seen and examined in the next section where we study G.

6 Studying the Variable: G

When we look at varying G values, we again hold all other variables constant. This time, we set $N_{cards} = 16$ and $N_{games} = 10$. Figures 11-13 in the appendix show G ranging from 1-3. We now see differences in both Density and LLR plots. Reducing G correlates to a tighter overlap of our Density plots, the null and alternative hypotheses become less discernable, as indicated in the LLR plots where $\beta \propto G$

Again, this is as expected. Increasing the number of cards that are guaranteed to win will make the data more discernable; the probability of winning a cheated game is much higher than in a fair game.

7 Conclusion

In summary, we have seen that our cardgame based on random chance with the opportunity for cheating can be distinguished over multiple trials using the β statistic from the log-likelikehood ratio. We can graph the density of data and see real changes to the distributions of probabilities over hundreds of games and we can analyze the data with likelihoods. As the number of games increases, the number of cards decreases, and the G value increases, we see an decrease in our β value, which reveals the power of the test.

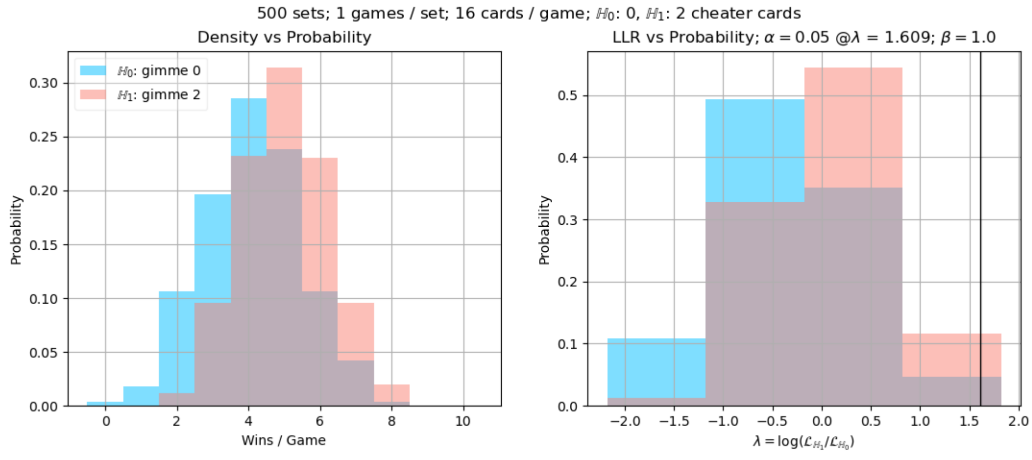

Figure 1: $N_{games} = 1$

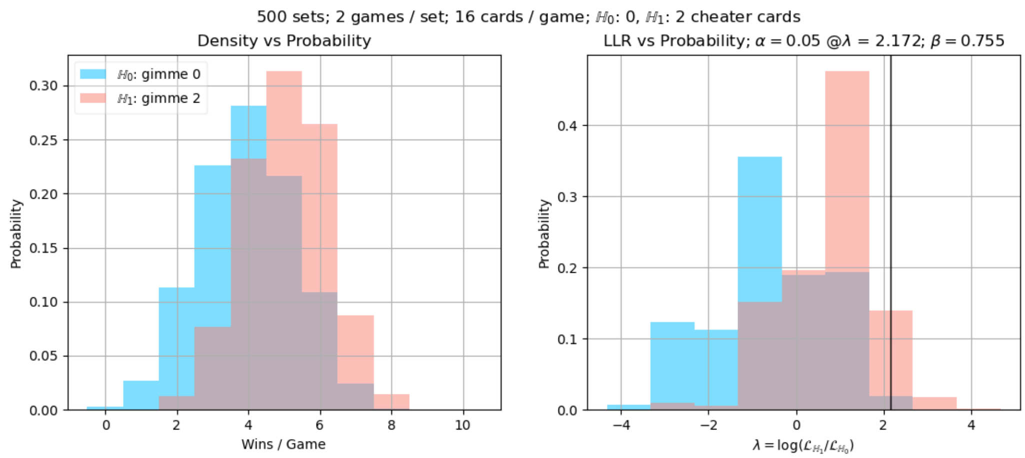

Figure 2: $N_{games} = 2$

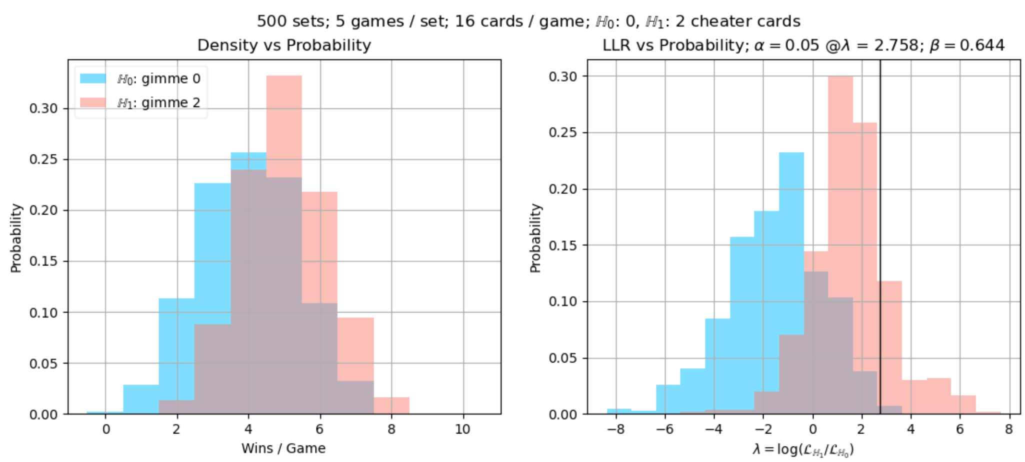

Figure 3: $N_{games} = 5$

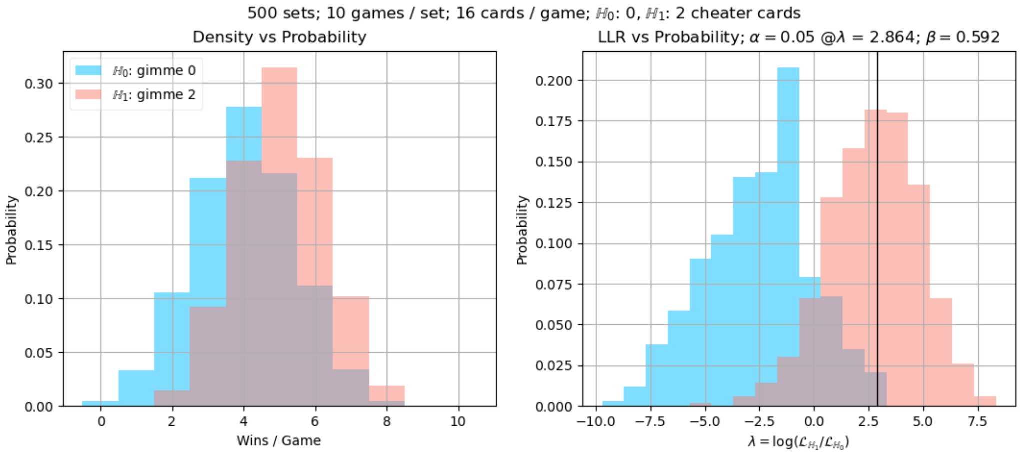

Figure 4: $N_{games} = 10$

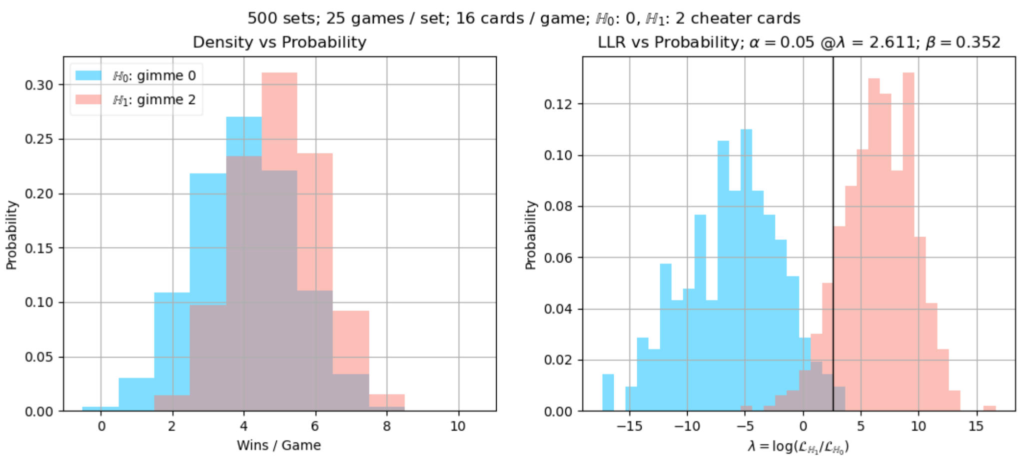

Figure 5: $N_{games} = 25$

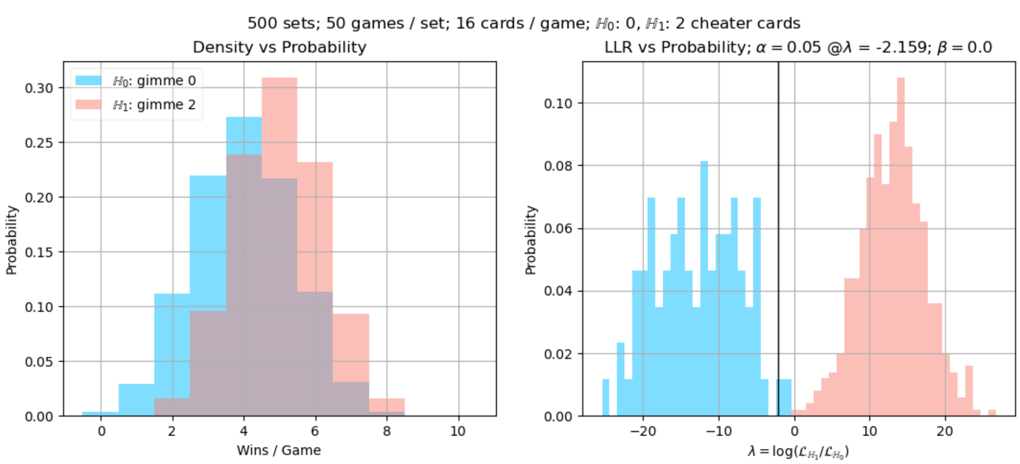

Figure 6: $N_{games} = 50$

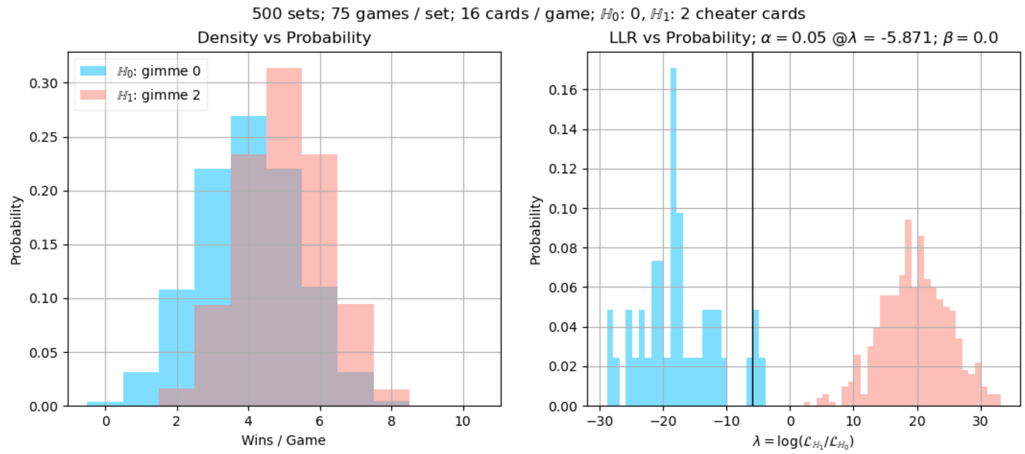

Figure 7: $N_{games} = 75$

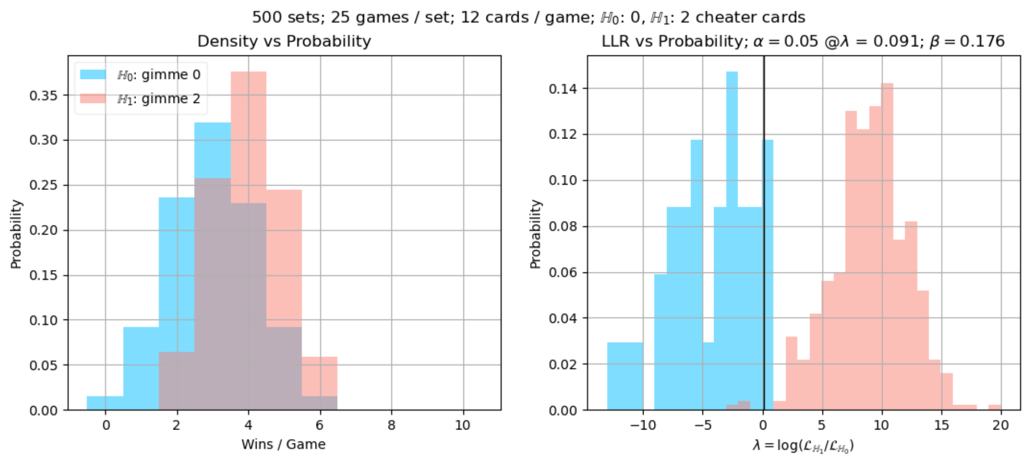

Figure 8: $N_{cards} = 12$

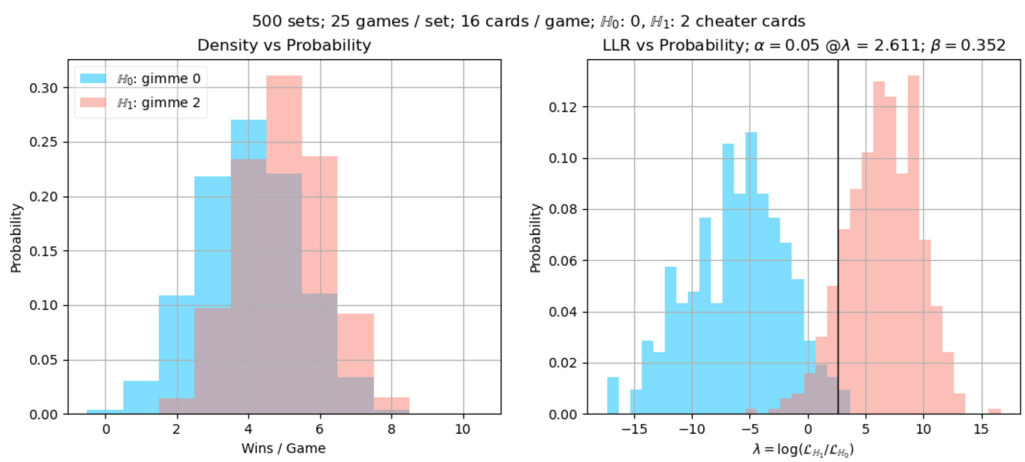

Figure 9: $N_{cards} = 16$

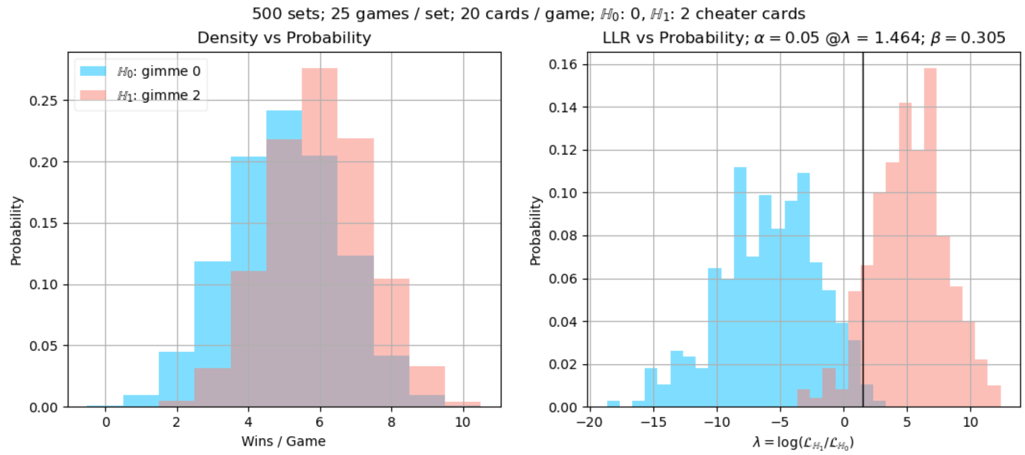

Figure 10: $N_{cards} = 20$

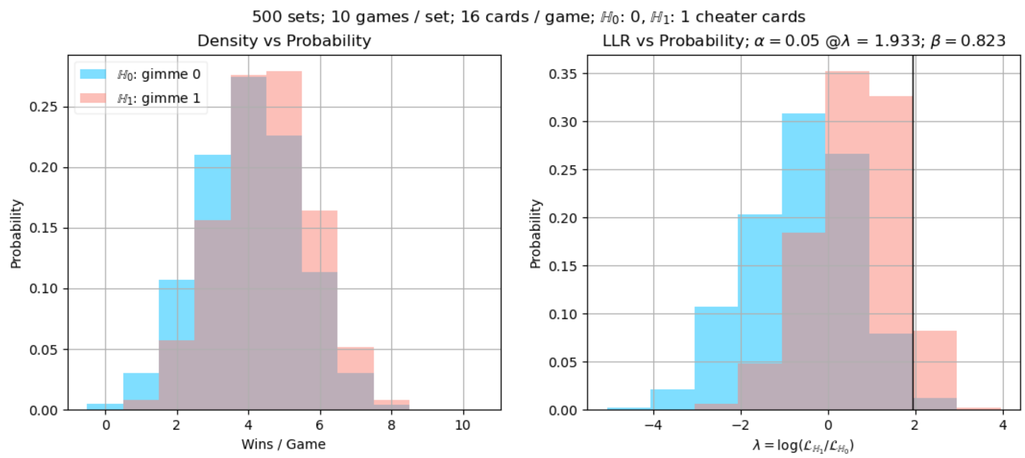

Figure 11: $G = 1$

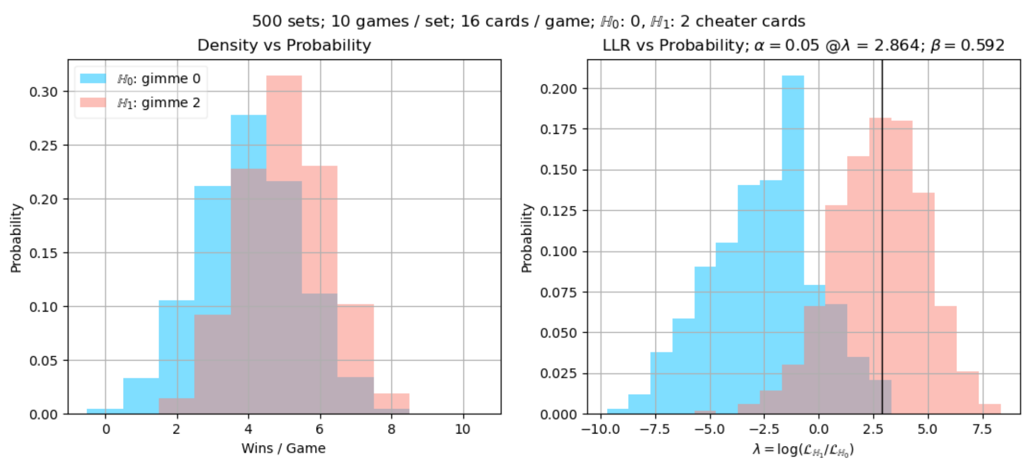

Figure 12: $G = 2$

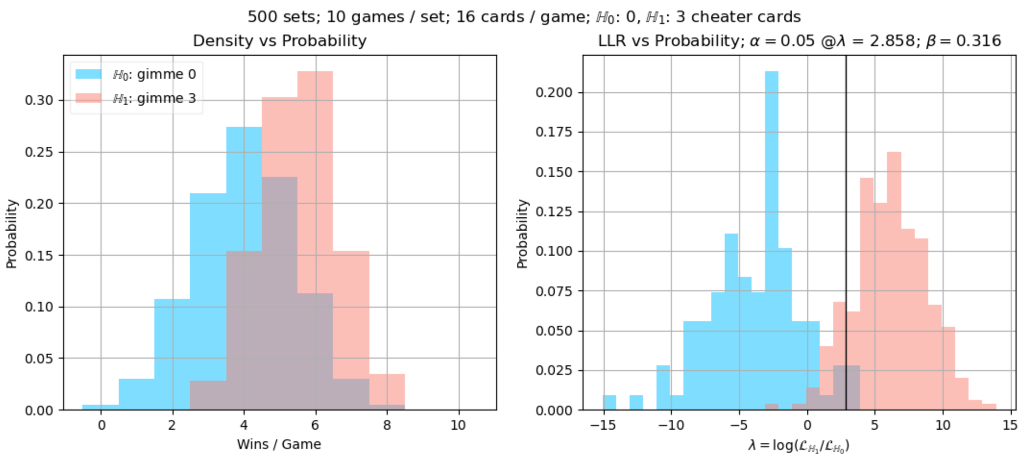
500 sets; 10 games / set; 16 cards / game; $\mathbb{H}_0$: 0, $\mathbb{H}_1$: 3 cheater cards
Density vs Probability
$\mathbb{H}_0$: gimme 0
$\mathbb{H}_1$: gimme 3
Probability
Wins / Game
LLR vs Probability; $\alpha = 0.05$ @λ = 2.858; $\beta = 0.316$
Probability
$\lambda = \log(\mathcal{L}_{H_1}/\mathcal{L}_{H_0})$

Figure 13: $G = 3$