Simulation data driven study of Skill in a game

Ansuman Banerjee, Swagatam Das, Subhamoy Maitra, Diganta Mukherjee
Indian Statistical Institute, Kolkata

Executive Summary

In this document, we report the salient outcomes of the study related to the project "Simulation Data-Driven Study of Skill in a Game" offered by Games24x7 Private Limited and executed at the Indian Statistical Institute from January 2024 to February 2024. Specifically, the key objective is to **examine** – (a) the **skill** element in a **new variant of Ludo**, designed by Games24x7, in a scientific manner; and (b) **without** any human **game play data** since the game is planned to be offered for play basis the establishment of skill element. The new variant of Ludo employs three dice rolls and gives players the choice to opt from up to three independent dice values (depending on their cyclic turns) and thus requires more informed decision making from a player in comparison to a single dice roll in traditional Ludo which does not provide any choice to a player.

To this effect, in this study, we perform theoretical and simulation data driven statistical analysis to evaluate presence of skill. We specifically try to address the following questions – (i) is there a stark distinction in the outcomes across different strategies that can be employed for playing the game?; (ii) is there an existence of winning formula (e.g., Nash equilibrium) from a set of strategies clearly showcasing the preponderance of skill requirement for sustained success in the game?; (iii) can a set of strategies leverage any existence of first or second movers' advantage in the game for a superior outcome; (iv) is there a progression of positive outcome with increasing learning investment in the game?; and (v) having the option to choose from three dice values in the new variant lead to superior outcome compared to the single dice traditional Ludo?[1]

For the simulation study, we employ bots (automated software applications executing repetitive tasks over a network) with different strategy implementation and make game play simulations among these bots. Specifically we employ two classes of bots – (a) intuitive rule-driven bots; and (b) learning based algorithmic strategies using Monte Carlo Tree Search (MCTS) – both with different degrees of decision making objectives. Finally, certain Game theoretic explanations are also presented. Nash equilibrium is found both theoretically and via the simulations for the set of strategies evaluated and simulation results further show upto ∼49.5% of skill premium (over and above the equi-probable 50% that would happen in a game of pure chance) for certain strategies over others. Some key observations from the theoretical and simulation driven empirical study are outlined below.

[1]This is not to say that the traditional version of Ludo is not a game of skill. Only that skilled players can use the additional information in the three dice variant smartly to get a superior outcome compared to the single dice one, where information availability is less.

- The presence of diverse strategies, each with its own level of complexity, consistently outperforming others while maneuvering through potential first or **second** mover advantages.
- The clear advantage of victory-seeking strategies over defeat-seeking ones, highlighting that chance alone cannot propel a defeat-seeking strategy to win randomly.
- The observable improvement in performance with increased learning investment in the game.
- Emergence of alternative strategies as winner for longer versions of the game, further highlighting the role of dynamic optimisation (at least intuitively, as in rule based play) in such settings.
- Moreover, it's worth noting that the three-dice iteration of Ludo enhances the role of skill compared to the traditional single-dice version. This is due to the requirement for more comprehensive decision-making in dice selection, which mitigates the impact of chance beyond the user's control, inherent in relying on a single dice value.
- A clearly higher information premium available in algorithmic play compared to rule based play. This is a very interesting and logical observation as "algorithmic" players are putting more effort in the computation of possibilities given the available information, and hence should be considered more skilled.

These observations in aggregate strongly indicate that in the two-player three-dice variant of the Ludo game, skill as well as progression of skill has a prominent role to play in improving performance.

1 Introduction

In the civilized world, games serve as both entertainment and cognitive exercises, fostering strategic thinking and social interaction. Irrespective of online or offline mode, they offer a diverse spectrum of experiences, engaging players in skillful pursuits and immersive narratives. Today, games are also a medium of business, drawing players from all corners of the world and impacting the economy. Ludo is a popular board game for 2-4 players. It originated in India and was inspired by the ancient game of Pachisi. The British popularized it globally in the early 20$^{\text{th}}$ century.

In this work, we try to identify the skill component of a specific version of the Ludo game. We like to identify the skill through different methods scientifically.

- We first explain certain game theoretic mechanisms to identify the skill component. In fact, we note a Second Mover's advantage in this specific game, which reduces with a larger number of moves. That itself talks about the skill component, as in soccer, a captain chooses the end based on the direction of the wind, or in cricket, the captain chooses either batting or bowling based on weather or pitch conditions after the toss is won.
- Next, we design certain simulations with different skill levels and show that the hypothesis that the skilled ones are winning compared to the less skilled, unskilled, or random players is clearly supported by extensive experiments.

1.1 Organization & Contributions

In Section 2, we present the background material. We initiate an informal discussion on why it is critical to establish objective metrics for regulation in the online gaming industry. This is followed by an exposition of Ludo games in general (Section 2.1) and consequently of the specific variant that we study here (Section 2.2). Section 3 describes the setting of the problem of skill vs chance, and in this regard, certain theoretical angles from game theoretic viewpoints are explored, which we later substantiate by data. Points related to Nash equilibrium are presented in Section 3.1 and then the critical observation explaining the second mover's advantage is inspected in Section 3.2. The detailed skill inclusion mechanisms in bots are described in Section 4 for 16 turns, and the experimental data are collected and analyzed. Statistical interpretation of the data clearly suggests a strong presence of the skill component in this variation of Ludo. Consequently, Section 5.5 studies the 20 and 24 turns version, and we finally recommend the use of 20 turns in the game. Finally, Section 6.2 checks the statistical robustness of our methodology. Section 7 concludes the paper.

2 Background

Much has changed in the online gaming sector in India over the past few years, especially so in the last year itself, where the government has undertaken key measures to provide policy and regulatory clarity for this sunrise industry. The tech fraternity has also given this industry their vote of confidence as was evident through our recent study where amongst those surveyed, over 84% technology practitioners believed that India could lead the global gaming industry.[2] To ensure that the industry realizes its full potential, there is a need for rationalization of regulation based on scientific and objective metrics. Such metrics can help to scientifically identify various gaming genres and provide suitable regulatory frameworks that protect both business and consumer interest while meeting the good governance objectives.

The use of scientific standards and data-backed decision-making in the formulation of regulatory policy is a well-established approach that has widened regulatory oversight over legitimate businesses while allowing consumers to make informed decisions.

A prime example of this can be seen in the food and beverage industry, where FSSAI has mandated the provision of nutritional information, allowing consumers to choose products suitable for their dietary requirements while creating greater awareness for healthy eating choices without restricting any product categories from being available in the market.

Rational and scientific benchmarking also allows businesses to remain viable and competitive, driving innovation and investments into creating more valuable products and services for consumers in a free market environment.

India's online gaming market still lacks transparency to certain extent due to the absence of a regulatory framework. The prime confusion is related to divergent perceptions in distinguishing the games of skill and games of chance. In essence, skill-based games are constitutionally protected businesses that fall under the purview of the Centre; while games of chance are classified as betting

[2] This is part of the findings of Mukherjee and Maitra, 2023.

and gambling, falling under the state regulations. Despite significant clarity at the Centre, various state-level regulations continue to adversely impact the skill-based online gaming sector. For example, one may refer to the recent Madras High Court judgment (November 2023), which has overturned a move by the state government to ban skill-based games like Rummy and Poker under the purview of its Prohibition of Online Gambling & Regulation of Online Games Act.

Some argue that the line between games of skill and games of chance is not always clear. In line with such arguments, it has been observed in the recent past that skill-based gaming operators have been issued demands for retrospective taxes at (higher) rates commensurate with the games of chance. As a corollary, this may decelerate the growth trajectory of the skill gaming sector.

Hence, what distinguishes these two types and what factors should be foremost when discerning between a game primarily reliant on skill versus one driven by chance? In essence, proficiency, experience, and structured gameplay strategies serve as pivotal indicators of a 'game of skill'. In such instances, a player's performance hinges on their mastery and comprehension of the game's mechanics. Players are tasked with leveraging their abilities and insights into the game's intricacies to outmaneuver their adversaries.

Additionally, factors like practice and experience hold considerable sway; in a skill-based game, an individual lacking experience is more prone to losses compared to someone who has dedicated time and effort to refining their abilities. Conversely, in games of chance, no such predispositions exist, and any participant holds an equal likelihood of winning or losing. These considerations lend themselves to scientific scrutiny, amenable to assessment through empirical experimentation, statistical analysis, and mathematical frameworks. Thus, the development of quantifiable metrics, guided by scientific acumen, becomes imperative to accurately gauge where a game resides on the continuum from chance to skill, particularly when the determination is not binary.

In the present scenario, it is imperative to have a clear and transparent methodology for determining whether a game is based on skill or chance. Such a methodology should be fair, standardised, and based on objective criteria. Presently, with varying and subjective understanding of these concepts, the industry faces challenges dealing with regulatory authorities, often having to resort to legal recourse to demonstrate that their skill-based game offerings fall squarely within the purview of the law. Once there are clearly defined objective metrics to determine the preponderance of skill, gaming companies can focus on innovation in designing games that emphasise skill, strategy, and decision-making. These elements form the foundation of games considered to be skill-based. By doing so, gaming companies can efficiently align their products with the legal framework and ensure that they are compliant with the regulations of the country.

The context is now set. Before proceeding further, let us now present the game concerned. We will first explain the game in general and then concentrate on the specific version that we consider here.

2.1 Generic Ludo

In the game of ludo, each player rolls a die and moves their four tokens around the board, trying to get them to the destination squares. The tokens are usually of four different colours: *red*, *green*, *yellow* and *blue*. One token is considered to be active only after a 6 is rolled, and the player can start moving the token after that. One can capture others' tokens, which will lead the captured token to be considered inactive again. All four tokens are initially inactive. The first person to get all their tokens to the destination square, which is 56 squares away from the start, wins.

Once activated, tokens start from their respective coloured squares adjacent to the 6x6 corner squares, move around the board and aim to reach the finish, which is the purple square adjacent to their coloured row of squares in the 8^{th} row. Thus, (7,2) is the starting square for red tokens and (9,14) is the starting square for blue tokens. Again (8,7) is the destination square for the red tokens and (8,9) is the destination square for the blue tokens.

Now, let us consider a game between two ludo players, playing with the colours **red** and **blue**. The game starts off with alternate dice rolls by the red and blue players. When a 6 appears in roll of either player, one of their tokens in considered to become active and moves to the beginning square. In the next turn, if a 6 appears, either the second token can be activated or the first token can be moved 6 squares; otherwise, the first token moves a number of squares equal to the roll, in it's designated path. In any of the consequent turns, if 6 appears, an inactive token may be activated as long as there are such tokens. If a red token reaches the same position as blue token (except their respective starting squares) after the roll, the blue token is 'captured' and considered inactive, and vice-versa. No token can be captured on the starting squares of either player or one of it's safe squares (the row of squares with the same colour on the 8^{th} row.)

As long as there are zero or one active tokens, there is only one choice of play available to both players. However, choices and complexities arise when there are multiple active tokens.

Extra move rule:

In ludo, a player gets an extra 'dice roll' every time they either (a) roll a six, (b) cut another player's token or (c) get a token home.

- In both the formats, in the event a player simultaneously rolls a six and either (1) cuts another player's token or (2) gets a token home, they will only get one extra turn.
- If a player rolls 3 sixes consecutively, the third six rolled is treated as null and void. Hence, if a player rolls three sixes in a single turn, the third six will not count and only two sixes will count for that turn. Upon rolling third six, the player will simply pass the turn to the next player.

2.2 The New Variant of the Game

Now, let us consider a new board game, played on a board similar to a ludo board, but with three independent die and a fixed number of turns.

In the first turn of the game, the first moving player (for example, red) rolls the three die. He can choose any of the three rolls and move any of his tokens by a number of squares equal to that roll. It is to be noted that all tokens are considered active at the beginning of the game, with no requirement for 6 to start. Next, the second moving player (say, blue), chooses one of the die and moves any of their tokens. The third roll, by default, is given to red.

In the next turn, blue rolls the three die and chooses any of those to move his tokens. Red then chooses one of the other two and the last one is given to blue. This continues till the total number of turns, considered to be either 16, 20 or 24, is reached (we consider results for all of these options). Thus, the total number of moves made by either player (excluding extra moves), is 24, 30 or 36 respectively.

The players still get an extra turn on rolling a six, capturing a token, or getting a token home. However, that extra turn is not considered to be a separate turn. When two or more tokens of same colour are on a square, they form a safe zone and cannot be captured by opponent tokens reaching that square.

Rules of promotion and capturing are similar as in generic ludo, but the difference lies in the win criteria, which is based on a point system unlike generic ludo. A player is awarded 1 point for each square traveled by a token. Further, an extra 56 points are awarded for getting the token to "home". If a token is captured, it's points accrued are removed. At the end of the game, the player with more number of points wins the game.

3 Some Theoretical Considerations

Before embarking on a detailed empirical analysis, let us discuss what the game structure tells us on paper in terms of expected outcome(s). For the sake of simplicity, we will assume the expected point 3.5 ($=\frac{1+2+3+4+5+6}{6}$) in each throw and a Nash equilibrium[3] consideration for strategy selection by the players. As playing according to a strategy is in itself an act of skill, a favourable outcome with strategic playing supports the skill base of any game, that involves uncertainty (nature's move) over chance. It is pertinent to mention that we are considering only a limited number of simple and intuitive strategies to understand the role of skill (here strategic behaviour) in winning this game. There could be many other, more complex strategies possible. Here an initiating discussion is presented, and more comprehensive statistical results with different strategies will be explained in the following sections.

3.1 Nash Equilibrium on Expected Path

First note the following for the 24 move Ludo game:

(a) Expected moves & points:

- 24 moves
- +4 moves as one 6 is expected in six moves
- $+\frac{2}{3}$ moves as a double six is expected in thirty-six moves,
- each getting 3.5 on average for $24 + 4 = 28$ moves, except for the double 6, where it is 2.5 (as the third six will be spoiled, in this case the average is $\frac{1+2+3+4+5}{6} = 2.5$).

[3]Nash equilibrium is a very intuitive and the most popular notion of equilibrium in strategic games. For detailed discussion, refer to Dutta (1999). It suffices to say that a pair of Nash equilibrium strategies for a two player game is one where no player can benefit by deviating unilaterally.

- That is, expected number of total points $= 28 \times 3.5 + \frac{2}{3} \times 2.5 = 99.67$

(b) Pieces have symmetric chances to capture each other (hence $\frac{1}{2}$ each). Thus, the average progress will be half the remaining path if there are opposing pieces moving in the same region.

The strategies are defined in terms of pairs of players: {R1, R2}, {R3, R4} and {B1, B2}, {B3, B4} as it is easy to show that moving in pairs with a little staggering (small gap) is preferable in obtaining protection to moving alone. It can be noted that the first 12 moves (50% of the game) will roughly consist of reaching the opposition's starting point. After this, the following strategies are considered for the remaining 12 moves:

1. Promotion priority (PP): The leading two pieces R1, R2 keep moving forward and tries to get one of them (w.l.g. R1) to finish (moving 56 steps + getting 56 bonus points).

2. Safe progress priority (S): The other two pieces R3, R4 start moving and tries to reach the same point as R1, R2, so no attempt at finishing.

3. Mix of the two strategies above (M): Move R1 and R3 with the objective of reaching finish for R1 and progress for R3.

We consider expected outcomes when each possible pair of strategies play against each other. This is calculated assuming uniform probability of capture. The payoffs are calculated as below.

- Strategy (PP) vs. (PP): $(99.67 + 56 + 3.5, 99.67 + 56 + 3.5)$. This is total expected number of points from movements plus the 56 bonus points for promoting one coin. While this may have a second order effect of getting another six in the extra move and then getting some more points. As these come with very small probability and it will not change our results qualitatively, we ignore this for the present discussion.

- Strategy (S) vs. (S): $(99.67, 99.67)$ (in this case, there is no bonus as promotion does not occur).

- Strategy (PP) vs. (S)
  Payoff for (PP): $\frac{1}{4}(99.67 + 56 + 7) + \frac{1}{2}(37 + 12.5 + 3.5) + \frac{1}{4}(12 + 12) = 73$.
  The above calculation is based on four possibilities where the first player captures (or does not capture) times the second player captures (or does not capture). Assuming equi-probability, these occur with $\frac{1}{4}$, $(\frac{1}{4} + \frac{1}{4} =)\frac{1}{2}$ and $\frac{1}{4}$ probability respectively. When both sides do not capture, then there is expected movement (99.67) plus promotion bonus plus the gain from extra moves won for the capture. In the adverse case, with both sides capturing, the token is able to achieve a net movement of half the length of his home stretch only. In the mixed case, the progress is full home stretch plus half enemy territory for one token, half of home stretch for the second one plus one bonus move for capture.
  Payoff for (S): $50 + \frac{1}{4}(12 + 12) + \frac{1}{2}(37 + 3.5) + \frac{1}{4}(50 + 7) = 90.5$
  The calculation use the same logic as above.

- Strategy (M) vs. (M): $\frac{1}{4}(56 + 56 + 19) + \frac{1}{4}(12.5 + 12.5) + \frac{1}{4}(56 + 56 + 9.5) + \frac{1}{4}(12.5 + 25) = 103$
  Now, as the promotional possibility is pursued, in the best - no capture - case we get 56 plus

bonus 56 and the remaining expected progress. This happens with probability $\frac{1}{4}$. The worst case of both capture - again with probability $\frac{1}{4}$ - results in scores of 12.5 expected for each of the two tokens used. The mixed case of one promotion and one capture gets 56 plus bonus 56 and half of the remaining expected progress. The other mixed case with only the advance token getting captured is similarly computed.

- Strategy (M) vs. (S)
  Payoff for (M): $25 + \frac{3}{4}(12.5 + 25) + \frac{1}{4}(30 + 56 + 19 + 3.5) = 80.25$
  Again, the probability of capture is attributed through the possible combinations. Now (M) suffers a higher risk of being captured (in 3 out of 4 possible configurations, hence $\frac{3}{4}$) compared to the (S) player.
  Payoff for (S): $50 + \frac{3}{4}(50 + 3.5) + \frac{1}{4}(12) = 93$

- Strategy (M) vs. (PP)
  Payoff for (M): $25 + 30 + 56 + \frac{3}{4}(10 + 3.5) + \frac{1}{4}(19) = 126$
  Payoff for (PP): $\frac{3}{4}(99.67 + 56) + \frac{1}{4}(12) = 120$
  Again, similar asymmetry of capture probability is present between (M) and (PP). We omit these details as the logic is similar to the earlier cases.

We summarise these and the corresponding outcome (payoff = 1(W) or 0(L)) in the tables below:

points	(PP)	(S)	(M)
(PP)	(159.17, 159.17)	(73, 90.5)	(120, 126)
(S)	(90.5, 73)	(99.67, 99.67)	(93, 80.25)
(M)	(126, 120)	(80.25, 93)	(103, 103)

Table 1: Points on Expected path for 16-turn game

Payoff (W/L)	(PP)	(S)	(M)
(PP)	(1/2, 1/2)	(0, 1)	(0, 1)
(S)	(1, 0)	**(1/2, 1/2)**	(1, 0)
(M)	(1, 0)	(0, 1)	(1/2, 1/2)

Table 2: Payoff on Expected path for 16-turn game

Remark 1. *With limited number of moves, in this hypothesized game play with a limited set of strategies, a safe play strategy like S turns out to preferred. In the sense of a Nash equilibrium, unilateral deviation being non-profitable. We do observe that ($S \times S$) is the only NE in this version. This is not to say that this will be the only combination observed in actual game play. Rather, we can say that if both the players get around to playing S, they will have no individual incentive to change.*

We will see later that this result morphs into other choices as we increase the number of turns. As choosing a Nash Equilibrium is an activity requiring strategic consideration, this automatically requires skilled maneuvering on behalf of the players. The statistical evidence in favour of this is presented in several subsections of section 5.

The strategies considered in the empirical analysis will be refinements of these, subject to various possibility of capture. Thus, when we analyse the game-play strategies in simulation, we will keep track of the distribution of points achieved.

While we understand that the complexities of various game play paths are too high to be fully addressed in this short report, we do try to visualise these in terms of some rather simplified statistics like win percentage, average points gained and its variability. While winning is the ultimate goal, the points gained even in a loss has important signals about the nature of game play. A high risk strategy may lose against a more safe strategy often but still score points at a higher rate (losing closely and winning hands down, but less often).This will then show up in the variability measure (here SD). The skill of a player then would show up in being able to choose the right (combination of) strategy given the version of the game and may as well depend on the state of play. We explore these issues in a limited way below.

3.2 The Second Mover's Advantage

Before analysing the game outcomes for alternative choice of strategies, we analyse another related point theoretically. Let us consider the situation of two players using a strategy similar to (PP), i.e, they move their first active token, and make captures when possible. Now, captures are possible after any token has covered more than 26 squares and eventually crossed the opponent's starting square. To find the turn from where captures are likely to start, we compute the expected value in one move of a fair die, considering the standard rules of ludo regarding extra turns and assuming no captures.

The probability distribution of X, the value obtained in the move, is given by (note that X can only take the values $1, 2, \ldots, 17$, excluding the possibility of captures):

x	Pr(X=x)
1, 2, 3, 4, 5	$\frac{1}{6}$
7, 8, 9, 10, 11	$\frac{1}{36}$
12, 13, 14, 15, 16, 17	$\frac{1}{256}$
Total	1

Thus, the expected value of X is given by $E(X) = \sum_{x=1}^{17} xP(X = x) = 4.153$.

Further, the captures begin after a token has traveled more than 26 squares, and we can assume the first capture is only possible when the token has traveled between 27 to 32 squares, which is within the range of a dice roll from the opponent's starting square, which gives us an average of 29.5 squares traveled before the first capture.

Considering the expected value of X to be the average number of squares traveled by the first token in a move, the average number of moves before the first capture is as follows:

$$\frac{\text{average no. of squares traveled before first capture}}{\text{average no. of squares traveled in a move}} == \frac{29.5}{4.153} = 7.11.$$

Since the number of turns is always an integer, it is approximated to the nearest integer, i.e., 7.

Now, 7 moves are equivalent to 2 complete turns in the game and one move by the first player. So, it is the second player who is expected to get the opportunity of choosing when the first chance of capture arrives, and has a $1 - (\frac{5}{6})^2 = \frac{11}{36}$ chance of making a capture. However, the first player has only a $\frac{25}{256}$ chance of making a capture in his next move, assuming the second player's active token does not move in the last move(which is usually not the case, so the probability is even less).

Hence, being significantly more likely to make the first capture, the second player enjoys an advantage in games involving players with the ability to make captures and moving one token at a time. Similar logic can be applied to prove second player's advantage even when one player moves in pairs (as in the "responsible pair" strategy).

Remark 2. This kind of advantage supports our skill over chance argument as such advantage is clearly documented for classical skill games like Chess (White vs. Black)[4], where the first mover can influence the selection of opening strategy for the Black by choosing a particular opening. *In Cricket and Tennis (Bat/Bowl[5] or Serve/Receive) by dictating who will get first use of a pitch or being better able to use the direction of the sunlight and wind. Here, in the version of Ludo, this is modelled through the first active engagement when pieces from both sides start sharing the same zone of the board. The player who initiates this engagement has some advantage. The second mover has a higher probability of initiating this kind of engagement. We refer to subsection 5.2 to present the supporting statistical data.*

4 Specification of Players' Strategies for Empirical Evaluation

For the empirical evaluation, we do not consider (M) directly as it is not a winner. Instead we create two sets of three (i.e., a total of six) different alternatives that represent different decision capabilities while playing the game. These sets are: (i) **Set I** — *intuitive rule based* strategies, similar to that mentioned in the previous section; and (ii) **Set G** — *algorithmic* strategies, using either random or sophisticated self-learning based game-play using Monte-Carlo-Tree-Search (MCTS). In this section, we describe these strategies in detail.

4.1 Set I (Intuitive rule based strategies)

We use three alternatives where one is similar to (PP) (called "aggressive"), a second similar to (S) (called "responsible pair") and a relatively unsophisticated algorithm (called "naive"). We detail below.

[4]Sonas (2002) documents a 4% (3%) advantage for White in classical (Rapid) chess games.

[5]Sood and Willis documents an average of 2.8% advantage due to winning the toss

4.1.1 The "Naive" Player (N)

The naive player follows a very simple algorithm for movement of tokens. As long as the first token is active, the token is moved by a number of squares equal to the die roll in it's designated path; unless the move is illegal because it crosses it's destination square with that move.

If the first token is inactive or it's movement is illegal, the next active token is moved in a similar way. If the first token is captured at any point, it is still given priority in movement compared to the other tokens. That is, the capture of a token does not affect it's movement algorithm when it's activated in the future.

All tokens are considered inactive when they reach their destination, and the player wins if all his tokens reach the destination before the opponent's.

4.1.2 The "Aggressive" Player (A)

The aggressive player follows an algorithm similar to the unskilled player's , but with a few more choices, which usually work in the aggressive player's favour.

The main difference lies in the choice of movement of an active token for the aggressive player. If there is the possibility of capturing an opponent token with a move, the aggressive player prioritizes that capture (for example, say an opponent token is 4 squares away from the third player token, and the roll is 4. Then the aggressive player will prioritize moving the third token over the other actives and capture the opponent token.)

Further, if it is observed that one of the player tokens can reach the 'safe' squares (the row of same coloured squares in the 8^{th} row, where the opponent tokens can't capture it) after the turn, it is given priority and moved off to the safe square. However, if a token can be promoted, i.e, moved to it's destination square, it is given the highest priority, since it will award at least 57 points to the player. If none of the above movements are possible, the aggressive player moves the first active token, choosing the highest roll available. Thus, the action performed by the aggressive player in any specific turn (if possible), in decreasing order of priority is-

- Promoting a token
- Capturing an opponent token
- Moving a token to one of the safe squares close to promotion
- Moving the first active token with the highest roll available

4.1.3 The "Responsible Pair" Player (RP)

The player following the "Responsible pair" strategy has a considerably more complicated algorithm, initially aiming to get all of his tokens to the opponent's starting square. The first two tokens start moving in alternate turns, and once either has reached 27, one of the two remaining tokens start moving in their place. After all tokens have reached 27, the first two tokens start moving in alternate turns.

After crossing the opponent's starting square, all tokens prioritize captures in their movements. However, before a token has crossed the opponent's starting square, it cannot cross that square to make a capture.

When an opponent token is within a distance of six (the maximum dice roll) of one of the last two tokens of the player, that token is given priority in movement, in an attempt to capture the opponent token.

When one of the opponent's tokens have entered the safe squares close to the destination and are ready to promote, the "responsible pair" player prioritizes the movement of their highest value token, aiming to promote it. All other tokens are only used for captures and moving to safe squares if possible. If that token gets captured, the next highest value token is given priority,

At any move, if no skilled movement is possible, the responsible pair player chooses the highest roll available.

Thus, the actions that can be performed by the "responsible pair" player, in decreasing order of priority, are-

- Promoting a token
- Capturing an opponent token
- Moving a token to a safe square
- Moving the highest point token when the opponent's token is close to promotion
- Chasing an opponent token with either of it's last two tokens
- Moving the tokens in alternate turns till none of them have reached 27, provided none of them reach 27 with the highest feasible roll
- Moving the tokens in alternate turns after all of them have reached 27 with the highest feasible roll

In our empirical evaluation that follows, three variants of the game, depending on the number of moves allowed, are considered, consisting of 16, 20, and 24 total turns, respectively. We simulated the games 10000 times and the results have been presented in the following.

4.2 Set G (Learning based algorithmic strategies)

In addition to the intuitive rule driven strategies, described above, we also design three algorithmic strategies of different decision making capabilities including self-learning capabilities in the couple of them with different objectives.

- **Defeat-seeking (DS) strategy** bot that is programmed to consistently aim for defeat.
- **Random (R) strategy** bot that takes random action from the set of possible moves at each state.

- **Limited-information (LI) strategy** bot playing in a manner akin to **traditional Ludo**, where it *only considers the first dice and ignores the other two dice*, but selects the most strategically advantageous action aimed for victory based on the first dice alone.

- **Full-information (FI) strategy** bot for the 3-dice Ludo is designed to attempt victory.

4.3 Full-information (FI) Strategy Bot

This bot generates decision trees corresponding to each action at a game state. In the 3-dice ludo, this will mean to 12 possible actions (max of 3 dices multiplied by the 4 movable pawns). However, depending on the game state the action space may be less than 12 (e.g., only 2 dice values to select from as a second mover and 3 pawns remaining to be moved, will lead to 6 possible actions at that state). Each branch (possible action) is then taken, and the game is simulated to its conclusion, akin to Monte Carlo Tree Search (MCTS) **?**, repeatedly (100 iterations, which is configurable) to determine the win rate for each branch. The action with the highest win rate (based on the posterior simulation) is selected for movement. Algs. 1, 2, and 3 depict the entire process followed in the bot. Following is the description of the steps in the algorithm.

Algorithm 1 LudoNode Class

function LudoNode($state, parent$)
  $self.state \leftarrow state$ ▷ Current state of the game
  $self.parent \leftarrow parent$ ▷ Parent node
  $self.children \leftarrow []$ ▷ List of child nodes
  $self.visits \leftarrow 0$ ▷ Number of times node has been visited
  $self.value \leftarrow 0$ ▷ Accumulated value of the node
end function

1. Initialization (Alg. 1):
   - Create a class LudoNode to represent nodes in the search tree.
   - Each node contains the current state of the game, a reference to its parent, a list of children, the number of visits (visits), and the accumulated value (value).
   - Implement the *mcts_ludo* function as the entry point, taking the root node and a simulation budget.
2. Selection (Alg. 2):
   - The selection function traverses the tree from the root to a leaf node using the UCT (Upper Confidence Bound for Trees) formula until an unexplored node or a terminal state is reached.
   - It uses the *select_child* function to choose the child node with the maximum UCT value. For this purpose multiple iterations are run to generate necessary statistics for the UCT. The maximum number of iterations, represented by *budget*, is a hyper-parameter that can represent how deep the learning needs to be. The learning is expected to be better with higher budget. In Section 5.4, we vary this parameter to show that in the Ludo variant

Algorithm 2 Monte Carlo Tree Search for Ludo

```
function MCTS_LUDO(root, budget)
    for iter in 1 to budget do
        node ← SELECTION(root)
        result ← ROLLOUT(node)
        BACKPROPAGATE(node, result)
    end for
    return BEST_CHILD(root).state          ▷ Return the state of the best child as the move to make
end function
function SELECTION(node)
    while node is fully expanded and not a terminal state do
        node ← SELECT_CHILD(node)
    end while
    return node
end function
function SELECT_CHILD(node)
    if node.children are not fully expanded then
        return EXPAND(node)
    else
        return ARGMAX(child_of_node{ child.value/child.visits + C · sqrt(log(node.visits)/child.visits) })
                                           ▷ Upper Confidence Bound for Trees (UCT) Formula
    end if
end function
function EXPAND(node)
    unexpanded_moves ← GET_UNEXPANDED_MOVES(node)
    random_move ← RANDOMLY_CHOOSE(unexpanded_moves)
    new_state ← SIMULATE_MOVE(node.state, random_move)
    new_node ← LudoNode(new_state, parent = node)
    node.children.append(new_node)
    return new_node
end function
function ROLLOUT(node)
    current_state ← node.state
    while not GAME_OVER(current_state) do
        legal_moves ← GET_LEGAL_MOVES(current_state)
        random_move ← RANDOMLY_CHOOSE(legal_moves)
        current_state ← SIMULATE_MOVE(current_state, random_move)
    end while
    return EVALUATE_RESULT(current_state)
end function
function BACKPROPAGATE(node, result)
    while node is not null do
        node.visits += 1
        node.value += result
        node ← node.parent
    end while
end function
function BEST_CHILD(node)
    return ARGMAX(child_of_node{child.value/child.visits})
end function
```

Algorithm 3 Utility Functions Specific to Game Rules

function GET_UNEXPANDED_MOVES(*node*) ▷ Return a list of legal moves that haven't been expanded yet
end function
function GET_LEGAL_MOVES(*state*) ▷ Return a list of legal moves in the current state
end function
function SIMULATE_MOVE(*state*, *move*) ▷ Simulate the effect of making a move in the current state
end function
function GAME_OVER(*state*) ▷ Check if the game is over in the current state
end function
function EVALUATE_RESULT(*state*) ▷ Evaluate the result of the game in the current state
end function

the learning can indeed increase with the budget, showcasing more skill and hence better outcomes can be attained with more budget.

3. Expansion (Alg. 2):

    - The expand function is called when a node is selected for expansion.
    - It selects an unexpanded move, simulates the move to create a new state, and adds a new child node to the current node.

4. Simulation or Rollout (Alg. 2):

    - The rollout function performs a random rollout from a given node until a terminal state is reached.
    - It randomly selects legal moves at each step and simulates the effects of those moves until the end of the game.
    - The result of the simulation is then evaluated based on the win rate of the rollout for a given selection (i.e., the value of each node is a reflection of the win rate given a rollout from that node).

5. Backpropagation (Alg. 2):

    - The backpropagate function updates the visits and value of each node in the selected path based on the result of the simulation.
    - It propagates the information up to the root of the tree.

6. Best Child Selection (Alg. 2):

    - The *best_child* function selects the child node with the highest average value (value/visits) as the best move.

7. Utility Functions Specific to Ludo (Alg. 3):

    - Implement utility functions such as *get_unexpanded_moves*, *get_legal_moves*, *simulate_move*, *game_over*, and *evaluate_result* based on the rules of the Ludo game.

- These functions define the game mechanics, legal moves, and how to evaluate the result of a game state.

8. Output (Alg. 2):

   - The *mcts_ludo* function returns the state of the best child as the recommended move.

9. Implementation Considerations:

   - Fine-tune parameters like exploration-exploitation trade-off (C) and simulation budget based on experimentation and the specific characteristics of the Ludo game.

This pseudocode serves as a template, and the utility functions need to be implemented based on the specific rules and mechanics of the Ludo game you are working with. MCTS is powerful for exploring and exploiting a large decision space without the need for domain-specific knowledge. The trade-off between exploration and exploitation, often managed by the UCT formula, is a key component of its success.

Random (R) Strategy Bot

The random player chooses any one of the possible moves with equal probability at every move. It chooses one of the four tokens and one of the available rolls independently and moves that token by the number of squares equal to the roll, provided the move is legal. If the move is illegal,another token and roll are drawn. This bot employs a random selection method for both the dice roll (from 1 to 3 possible rolls) and the subsequent action. For example, given dice rolls [2,5,_] and randomly choosing dice 5, with pawn positions at [54,2,3,3], the bot randomly selects a move from the movable pawns (in this case, pawns 2, 3, and 4).

Limited-information (LI) Strategy Bot

This bot begins with the first available dice from the common pool. The bot then generates decision trees corresponding to each movable pawn. Each branch (pawn) is then advanced, and the game is simulated to its conclusion, akin to Monte Carlo Tree Search (MCTS), repeatedly (100 times, which is configurable) to determine the win rate for each branch. The pawn with the highest win rate is selected for movement. The utility functions outlined in Alg. 3 is different in this bot compared to Skill 1 bot because the possible legal moves would be different at each as mentioned earlier. The rest of the Alg. 0 and 1 typically remains similar.

Defeat-seeking (DS) Strategy Bot

This bot mirrors the approach of Skill Level 1 but inversely, opting for the dice and pawn combination that results in the lowest win rate. In other words, in the UCT formula in Alg. 2 is changed from *argmax* to *argmin* causing selection of child with lowest value or win rate.

5 Empirical Results

Given the different strategies described in the previous section, we now provide the empirical results from the detailed simulation study performed using these strategies. We implemented the strategies in R 4.3.2 and performed the simulation in RStudio 1.1.456 using Anaconda Navigator with parallel execution performed using doParallel package. Each game play between two strategies of sets G and I was simulated $10K$ times and game play between dominant strategies of both sets was simulated $1K$ times. This provided enough sample space on which statistical analysis was performed to arrive at the conclusions. For each pair of strategies, the first and second mover is swapped and the simulation is repeated again. For the bots in set G, we used a maximum *budget* (as in Alg. 2) of learning iterations to be 100, which was arrived at empirically based on cost and time incurred for the simulation. The average time taken to perform a simulation between two strategies for 100 iterations of learning with $1K$ simulations, parallelized across 31 cores, was $\sim$539 minutes. For bots in set I, the time taken is much less, approximately $\sim$46 seconds, since these are rule driven and do not include learning iterations at every step of decision making.

The findings are presented in the following manner. In Section 5.1, we delineate a comparison among various strategy bots engaged in gameplay, aimed at discerning the skill premium across them. This analysis includes identifying the Nash equilibrium derived from simulation results. Section 5.2 illuminates the advantage enjoyed by second movers within the array of strategies, elucidating how different approaches exploit or mitigate this advantage. Exploring the variance in outcomes between considering a single die versus three dice values, as in traditional Ludo versus the new variant, is the focus of Section 5.3, shedding light on the notion of skill premium. The progression of favorable outcomes within learning-based strategies in set G, relative to increasing learning iterations (i.e., varying *budget* in Alg. 2), is detailed in Section 5.4. Finally, in Section 5.5, we extend the maximum number of allowable moves in the game to observe any resultant changes in outcomes relative to game length.

5.1 Comparison across strategies

We perform the comparison among the various strategies within sets I and G, separately and then compare the dominant strategies from both the sets as well.

5.1.1 Comparison within set I

Table 3 summarizes the results for strategies in set I. Table 4 further showcases the game-theoretic pay-offs within the strategies of set I. The payoffs are computed based on the win/loss percentage of the strategies. Some key findings from these results are as follows.

- Outcome observations
  - RP and A clearly outperforms N both in terms of win% as well as the mean points achieved. In a later section 6.1, we substantiate all the performance comparison statements statis-

Player 1 Strategy	Player 2 Strategy	Player 1 Win Percentage	Player 1 points		Player 2 points	
			Mean	SD	Mean	SD
Naive	Naive	49.83	159.7	41.01	157.2	42.39
Naive	Aggressive	0.22	75	53.16	209.6	48.65
Naive	Responsible Pair	13.6	82.2	54.1	135.1	39.51
Aggressive	Naive	97.07	211.5	48.67	74.7	52.83
Aggressive	Aggressive	48.12	120.6	59.06	121.9	58.57
Aggressive	Responsible Pair	44.42	114.5	71.78	109.3	44.37
Responsible Pair	Naive	83.6	133.6	38.77	86.6	54.09
Responsible Pair	Aggressive	49.88	105.3	43.59	117.7	69.59
Responsible Pair	Responsible Pair	49.68	99.1	16.02	99.3	16.31

Table 3: Summary Statistics for various strategies of set I

Payoff (W/L)	(N)	(A)	(RP)
(N)	(1/2, 1/2)	(0, 1)	(0, 1)
(A)	(1, 0)	(1/2, 1/2)	(0, 1)
(RP)	(1, 0)	($1/2^+$, $1/2^-$)	**(1/2, 1/2)**

Table 4: Nash Equilibrium in winning

tically, in terms of both win percentage as well as average points earned. For ease of exposition, we do not insert these details in the descriptive discussion in this section.

- The performance between RP and A is interesting. While the win% if A as player 1 is lower than RP as player 2, mean points and standard deviation for A is higher. This indicates and we also observed that when A wins, it wins big but the % of times it wins is less. When A and RP swap position as player 1 and 2, the win rate evens out but the average points scored in the end by A remains larger.
- Our observations regarding A and RP further gets supported by the extent of wins of A with N compared to that of RP with N. Clearly, A wins bigger and with higher rate.
- Interestingly, A, even though winning bigger and better against N when compared to RP against N, is not guaranteed to be the best strategy to follow because its performance against RP is not guaranteed. This shows the clear existence of skill in decision making as a player where their and opponents' strategies can play a role in the outcome. However, there are some strategies, e.g., N, which can get easily defeated showcasing differentiated skill gradiants across strategies.
- On average (over order of move), the skill premium for (A) is 48.4%, for (RP) it is about 34.5% (over and above the equi-probable 50% that would happen in a game of pure chance).

- Payoff Observations and Nash Equilibrium from the strategies
  - (RP $\times$ RP) payoff matches with the theoretical result (S $\times$ S)
  - (N $\times$ N) payoff matches with (aggressive + no capture) payoff

- (A × A) payoff is less than the theoretical prescription (for PP × PP), closer to a mixed strategy like (M).
- In line with the theory, (RP, RP) is the only Nash Equilibrium. So, overall, RP is the winner among the strategies in set I.
- Note that even though the strategy pair (A, A) yields higher expected points for both players, it is not a NE as unilateral deviation is found to be profitable. Whereas, the actual NE (RP, RP) yields a lower payoff for both players. This harks back to the classical discussion of efficiency versus equilibrium in the game theory literature.

Remark 3. *In the above discussion we observed high skill premium for certain strategies as well as found a Nash Equilibrium within the strategy set considered. The tension between efficiency and equilibrium behaviour is also highlighted. Once again, at the cost of reiterating, we like to point out that this is once more a discussion that is closely intertwined with skill considerations.*

5.1.2 Comparison within set G

Table 5 summarizes the results for strategies in set I. These results are based on a learning budget (as described in Section 4 and beginning of Section 5). Key findings from these results are as follows.

- FI and LI clearly outperforms R and DS as one would expect since FI and LI both are victory seeking while DS is defeat seeking and R is random action.
- FI also outperforms LI. Unlike in set I, in set G, the win% between any two strategies is also correlating with mean point value difference achieved by the strategies. Hence, it can be concluded that FI is clearly the dominant strategy in set G as both player 1 and 2.
- FI deterministically wins against DS both as first and second player. This shows not only clear skill gradation but also establishes the skill attribute of the game where a player will certainly win if it optimizes towards the win while the opponent tries to lose. If the game had any predominance of chance, this could not have been ascertained because chance would come between for some games.
- To determine the skill premium for the strategies in set G, notice that when we square off LI with R, the advantage is 27.5% on average (over first and second move). As we upgrade to FI, the advantage premium is now 49.5%. Thus, the advantage of skill is very obvious. Also, FI against LI enjoys 34% skill advantage, confirming the overwhelming skill premium.

5.1.3 Results for dominant strategies of both sets

In this set of findings, a distinct advantage is evident for the I class of strategies. However, does this necessarily discredit the role of skill? We argue otherwise. This is because the learning iterations (budget) in FI are limited to just 100, while the potential outcomes in the subsequent dice roll (three dice) amount to $6^3 = 216$. Consequently, FI inadequately explores the vast search space, resulting

Player 1 Strategy	Player 2 Strategy	Player 1 Win Percentage	Player 1 points		Player 2 points	
			Mean	SD	Mean	SD
Defeat-seeking	Defeat-seeking	50.29	95.17	17.6	95.19	19.45
Defeat-seeking	Random	1.69	75.08	15.79	111.24	18.98
Defeat-seeking	Full-information	0	70.57	15.77	117.52	20.91
Defeat-seeking	Limited-information	0.2	73.17	15.93	113.34	21.23
Random	Defeat-seeking	98.08	111.26	18.07	75.35	15.55
Random	Random	47.59	89.93	19.48	92.26	17.55
Random	Full-information	0.93	76.66	15.78	110.17	19.22
Random	Limited-information	19.68	78.74	19.55	99.75	23.35
Full-information	Defeat-seeking	100	114.78	16.12	74	13.76
Full-information	Random	99.09	108.68	17.57	79.09	14.63
Full-information	Full-information	44.6	87.06	25.87	91.37	27.44
Full-information	Limited-information	81.93	101.77	23.09	84.03	27.33
Limited-information	Defeat-seeking	99.68	110.75	17.1	76.05	14.76
Limited-information	Random	75.1	97.8	22.6	81.88	19.51
Limited-information	Full-information	13.97	80.99	25.98	104.43	21.48
Limited-information	Limited-information	48	91.86	26.4	94.01	25.14

Table 5: Summary Statistics among strategies of set G

Player 1 Strategy	Player 2 Strategy	Player 1 Win Percentage	Player 1 points		Player 2 points	
			Mean	SD	Mean	SD
Aggressive	Full-information	55.7	117.19	83.29	65.87	23.55
Full-information	Aggressive	38.1	63.6	21.51	122.35	73.23
Responsible Pair	Full-information	50.05	98.97	18.07	93.14	25.56
Full-information	Responsible Pair	40.6	90.32	23.86	101.62	20.31

Table 6: Summary Statistics among dominant strategies of sets G and I

in insufficient skill acquisition. The failure of FI, therefore, serves as evidence in support of intuitive rule-based strategies. As elaborated in Section 5.4, the performance of FI improves progressively with a higher budget of iterations. With a more thorough exploration of the search space (even at a significant computational cost) or by enhancing the algorithm's learning efficiency, we anticipate that the advantage attributed to the I class could diminish, and perhaps even reverse.

Remark 4. *To conclude the discussion of this subsection 5.1, it is germane to mention that the more intelligent strategy wins a higher proportion of games against a strategy with lesser/no intelligence or with defeat-seeking intention, emphasising the role of skill in this game. The labelling of "more" or "less" is of course subject to perception but that in itself is an informed notion. To be precise, this argument is valid for the results available in subsections 5.1.1 and 5.1.2. In particular, an extreme manifestation could be observed in Table 5, where the "Full Information" is playing against the "Defeat Seeking" one irrespective of who is the first mover.*

5.2 Second movers' advantage

We first like to refer Remark 2 in this regard, where we have pointed out the Second movers' advantage. From the Tables 3, 5, 6, it is evident that for any pair of strategies the win rate and also the end points are consistently higher as player 2 compoared to that of player 1. This shows that for the sets of strategies across I and G, second movers' advantage is prevalent. Key observations in this regard are as follows.

- Second mover's advantage exists within set I (refer Table 3), it is 3% for the dominant strategy pair (RP, A).
- Second mover's advantage exists within set G (refer Table 5), it is 2% for the dominant strategy FI against LI. It is not very apparent against RP as the win percentage is already around 99%.
- Second movers' advantage is also evident from Table 6 as well between A, RP, and FI. It is 3% between A and FI while being 5% between RP and FI. Thus, apparently, the more conservative strategy RP leverages the second movers' advantage better.
- One may note that the second movers' advantage is related to the strategies that we have discussed. We may very well find a particular strategy pair such that the second movers' advantage is not exploited.

Player 1 Strategy	**Player 2 Strategy**	**Player 1 Win Percentage**	**Player 1 points**		**Player 2 points**	
			Mean	**SD**	**Mean**	**SD**
Naive	Naive	49.82	157.5	43.22	156.5	43.57
Naive	Aggressive	23.19	112.6	60.84	166.5	48.88
Naive	Responsible Pair	22.87	93.76	59.1	126.8	48.1
Aggressive	Naive	73.46	167.8	47.17	115	59.33
Aggressive	Aggressive	45.12	118.7	64.43	124.2	62.79
Aggressive	Responsible Pair	36.59	110.4	63.96	119.2	49.14
Responsible Pair	Naive	73.81	122.5	44.26	96.71	59.31
Responsible Pair	Aggressive	58.06	113.5	47.52	113.7	63.26
Responsible Pair	Responsible Pair	49.59	98.3	16.7	98.65	16.87

Table 7: Summary Statistics for strategies in I, considering one dice only

5.3 Comparison with one dice instead of three dices

For the sake of comparison, we also simulate a version of the bot which considers one dice instead of three dices. Looking at the table 7 and comparing with the results of table 3 we see a clear improvement on the skill perspective when three dices are considered. We claim that the single dice consideration is akin to conventional Ludo, and that three dice consideration increases the skill quotient of the game. Juxtaposed with the skill premium % improvement, we find very strong support for our claim. For example, consider the play between R and A in Set I under the 3 dice (Table 3) and 1 dice (Table 7) versions. The R vs. A advantage (5.6% as second mover and 0 as first mover) increases to 13% and 8% respectively. This margin of roughly 8% is clearly due to informational advantage that is leveraged by a strategic player. This is further evidenced by a comparison of second mover's advantage between R and A for the two versions considered. While for the three dice version, it is approximately 3.5%, for the single dice case it reduces to 2.25%.

In set G, the comparison between LI and FI further showcases the advantage of choosing from three dices w.r.t. consideration of a single dice only. It is important to reiterate here that FI clearly outperforms LI both as first and second mover and both from the perspectives of win % and end points. To summarize FI wins ∼82% as first mover and ∼86% as second mover resulting in an average win % of ∼84%. This clearly shows a skill premium of 34% of 3-dice driven decision making to that of single dice information. It is interesting to note that limiting information (showing only 1 dice at a time rather than 3) results in a more severe disadvantage in the algorithmic play (FI vs. LI) compared to rule based play (set I) where, as mentioned above, the information premium is around 8%.

Remark 5. *The above discussion reemphasises the role of information usage, and hence the role of skill, in this game. When the players have more information they are able to exploit it meaningfully to better performance. As the three dice version requires dynamic optimisation skills on the part of the players, it is a step higher on the skill ladder than the one dice variation.*

5.4 Skill progression with the extent of MCTS learning (computational budget)

In this section, we consider three variations of the FI Strategy bot of set G. As mentioned previously, for all our other simulations we have used the budget of iterations in Alg. 2 to be 100, i.e., FI bot used 100 iterations of self-play in each move to choose the best action based on UCT with the highest win rate. Here, we consider the original FI bot with 100 iterations, referred to as FI-100, along with FI-50 and FI-10 bots, which use the same algorithm as the FI bot to choose the action in each move, but with 50 and 10 iterations of self play per move, respectively. We have capped the budget to 100 iterations because of the cost and time constraints, as mentioned in the beginning of Section 5.

Player 1 Strategy	Player 2 Strategy	Player 1 Win Percentage	Player 1 points		Player 2 points	
			Mean	SD	Mean	SD
FI 100	FI 10	78.1	103.81	23.22	84.06	20.4
FI 10	FI 100	12.6	79.72	20.35	106.17	23.86
FI 100	FI 50	55.1	93.39	25.83	88.99	26.42
FI 50	FI 100	31.8	85.33	25.36	98.06	25.49
FI 50	FI 10	71.4	99.88	25.97	82.75	20.85
FI 10	FI 50	18.7	83.26	21.82	102.06	24.16

Table 8: Summary Statistics for FI bot with different budget (learning iteration)

As shown in Table 8, it is noted that with increase in learning budget, the outcome become superior, i.e., FI-100 > FI-50 > FI-10, both in terms of win % and mean points. To be specific, the skill premium of FI-100 is ∼33% and ∼12% over FI-10 and FI-50, respectively; while FI-50 has a skill premium of ∼26% over FI-10. This clearly shows a progression of superior outcome when the learning is better.[6]

Remark 6. *It is to be noted that the learning rate here may not be completely correlated to experiential learning in humans where there are evolution of learning over experience. Such an evolution may have infinite possibilities and hard to capture in a machine driven simulation (and can only be captured through real-user data analysis). However,* ***the variation of learning rate in FI and the subsequent progression of superior outcomes do substantiate the scope for learning in the game to perform better – thus pointing to the predominance of skill as well as progression of skill.*** *Design for experiments related to actual human game playing is already developed and the analysis in this regard will be presented once the data is available.*

5.5 Impact of number of moves

In this section, we examine if no. of maximum moves in the game has any effect on the outcomes. For this purpose, we have considered two game configurations with 30 and 36 moves. For the sake of brevity, we only showcase the results corresponding to Section 5.1 for these two additional game configurations, since that would suffice to showcase any impact of maximum allowable moves in the game. Now we present the results for the 30 move version.

5.5.1 Results in 30 move game

Table 9 summarizes the results for strategies in set I. Table 10 further showcases the game-theoretic pay-offs within the strategies of set I. The payoffs are computed based on the win/loss percentage of the strategies. Some key findings from these results are as follows.

- Second mover's advantage exists, it is about 1.5% for (RP, A). Reduced from 24 move version.
- On average (over order of move), the skill premium for (A) is 30%, for (RP) it is about 26.5% (above the random 50%).
- Interestingly, we now see that both (RP, RP) and (RP, A) are Nash Equilibria. So a longer format supports aggression more.
- Thus, again overall RP is the winner. But now the second mover may also choose A.

In contrast to the 24 move game, again the (A) strategy wins over the FI strategy (from set G) but now the (RP) strategy is beaten by FI. Thus, in this longer format, MCTS learning starts to perform better. But it should be noted that the average points earned by strategy (RP) is very close to that of FI even when it loses more often as the first mover. As second mover, strategy (RP) has a slightly higher winning percentage but earns 8 more points on average. This leads us to conclude that

[6] A minor curiosum: The win (learning?) rate progression is not linear in number of iterations of self play per move, from 10 to 50 to 100. Considering the superior bot as the second player, that would have resulted in a win rate of $81.7 + 0.68 \cdot 18.7 \sim 94\%$ for FI-100 against FI-10, but we see 87.4%. Thus, it is actually concave. So is there a learning saturation? It might be worth exploring (at considerable computational cost!) in future.

Player 1 Strategy	**Player 2 Strategy**	**Player 1 Win Percentage**	**Player 1 points**		**Player 2 points**	
			Mean	**SD**	**Mean**	**SD**
Naive	Naive	49.68	210.2	50.31	209.1	51.66
Naive	Aggressive	1.3	99.4	61.64	269.9	50.9
Naive	Responsible Pair	13.01	125.1	61.73	194.4	53.91
Aggressive	Naive	98.19	268.4	52.32	100.1	60.98
Aggressive	Aggressive	48.69	159.6	69.49	160.8	69.9
Aggressive	Responsible Pair	50.06	162.4	78.72	148	52.93
Responsible Pair	Naive	84.97	191.2	52.7	130.4	61.53
Responsible Pair	Aggressive	47.09	143.4	50.61	164	76.29
Responsible Pair	Responsible Pair	48.49	128.3	28.7	129.4	29.27

Table 9: Summary Statistics for various strategies in set I for 30-move game

Payoff (W/L)	(N)	(A)	(RP)
(N)	(1/2, 1/2)	(0, 1)	(0, 1)
(A)	(1, 0)	(1/2, 1/2)	(0, 1)
(RP)	(1, 0)	**(1/2, 1/2)**	**(1/2, 1/2)**

Table 10: Nash Equilibrium of Set I strategies for 30 move game

there are high-risk/high-gain versions of a game strategy compared with more conservative versions. A strategy may win spectacularly sometime but lose narrowly a higher proportion of time. This shows up in both the mean and SD (measure of fluctuation) of points being higher but winning percentage being lower. If the focus is primarily on winning percentage, such strategies may not be equilibrium choices despite having a higher average point count.

Player 1 Strategy	**Player 2 Strategy**	**Player 1 Win Percentage**	**Player 1 points**		**Player 2 points**	
			Mean	**SD**	**Mean**	**SD**
Aggressive	Full-information	69.6	167.11	92.72	75.1	30.94
Full-information	Aggressive	26.7	72.72	32.42	167.71	87.6
Responsible Pair	Full-information	40.2	126.65	40.99	129.26	44
Full-information	Responsible Pair	49.5	124.16	43.42	132.12	40.78

Table 11: Summary Statistics of 30 move games among dominant strategies of sets G and I

5.5.2 Results in 36 move game

Table 12 summarizes the results for strategies in set I. Table 13 further showcases the game-theoretic pay-offs within the strategies of set I. The payoffs are computed based on the win/loss percentage of the strategies. Some key findings from these results are as follows.

- Second mover's advantage still exists, and now it is roughly 2.5%. So it is slightly higher than the 30 move format. This is slightly surprising as one could reasonably argue theoretically that second mover's advantage (as theoretically established in 3.2) is significant when the number of moves is small (when the chances of promoting several pieces is very low, so one extra capture makes a significant difference). It gets mitigated when we consider longer games and there is scope for several capture/promotion. But we do not observe this very clearly in the empirical results. So change of strategy, as discussed below, may have a role to play here.

- So on average (over order of move), the skill premium for (A) is 32.5%, for (RP) it is about 25.5% (above the random 50%).

- Most interestingly, we now see that the primacy of the strategy (RP) is finally lost and the advantage of playing aggressively (A) comes to the fore. Now, (RP, RP) is no longer a Nash Equilibrium. In fact, (RP, A) is now the only Nash equilibrium. So this even longer format supports aggression more and actually does not support responsible play alone. Thus, now the first (second) player will choose RP (A).

Player 1 Strategy	**Player 2 Strategy**	**Player 1 Win Percentage**	**Player 1 points**		**Player 2 points**	
			Mean	**SD**	**Mean**	**SD**
Naive	Naive	49.49	259.3	53.49	257.8	54.2
Naive	Aggressive	0.95	128.5	68.96	331.3	55.15
Naive	Responsible Pair	13.25	168.8	65.66	262.3	65.99
Aggressive	Naive	98.93	330.2	56.23	126	68.43
Aggressive	Aggressive	49.31	196.6	77.5	196.9	77.49
Aggressive	Responsible Pair	58.99	213	84.53	181.8	55.93
Responsible Pair	Naive	84.1	260.1	67.04	173.8	65.27
Responsible Pair	Aggressive	35.89	177.3	53.83	216.7	82.24
Responsible Pair	Responsible Pair	48.77	189.9	57.99	191.4	57.15

Table 12: Summary Statistics for various strategies of set I for 36 move game

Payoff (W/L)	(N)	(A)	(RP)
(N)	(1/2, 1/2)	(0, 1)	(0, 1)
(A)	(1, 0)	$(1/2^{-}, 1/2+)$	(0, 1)
(RP)	(1, 0)	$\mathbf{(1/2^{-}, 1/2^{+})}$	$(1/2^{-}, 1/2^{+})$

Table 13: Nash Equilibrium in winning for 24-turn game

In continuation to the 20-turn game, again the (A) strategy win over the FI strategy but now the (RP) strategy is clearly beaten by FI. Thus, in this longest format, MCTS improves further. Again it should be noted that the average points earned by strategy (RP) is very close to that of FI even when it loses more often as the first mover. As second mover, strategy (RP) has a slightly lower winning percentage but earns 8 more points on average.

Remark 7. *While across different number of moves considered, the presence of skill premium and Nash Equilibrium considerations are evident; in the above comparative discussion of outcomes in games with different number of moves, we see that there is more variety in terms of equilibrium outcome when we consider 30 moves (20 turns with the 3 dice). We see such shifts in strategy in Chess also, when*

Player 1 Strategy	Player 2 Strategy	Player 1 Win Percentage	Player 1 points		Player 2 points	
			Mean	SD	Mean	SD
Aggressive	Full-information	75.4	218.98	106.69	90.17	48.7
Full-information	Aggressive	19.9	86.24	48.37	224.44	103.81
Responsible Pair	Full-information	38.1	175.36	59.44	177.09	59.22
Full-information	Responsible Pair	51.7	172.04	58.45	180.81	59.41

Table 14: Summary Statistics of 36 move games among dominant strategies of sets G and I

we move from the classical (longer) version to the Rapid version. In light of this, we may recommend the 30 moves game as one worthy of a deeper investigation.

Our examination of strategies and their outcomes unveils several nuances worthy of consideration. Firstly, if a player begins prioritizing point optimization, the once reliable safe strategy may become less secure, as it increases the likelihood of tokens entering the opponent's home zone. Furthermore, by refining the aggressive strategy to include a simple condition of waiting for a 6, which has a 42% probability in three dice rolls, before moving into the opponent's zone, the likelihood of capturing tokens rises to 7%. This minor adjustment can tip the scales in favor of the aggressive approach over the conservative one. Notably, occurrences of the number 6 are not particularly rare. Consequently, tailoring both conservative and aggressive strategies based on contextual game factors can confer an advantage to one over the other. Thus, is it appropriate to solely compare a limited set of strategies? Should we not delve into the integration of strategies within specific contexts? We refrain from such detailed discussions, as our primary objective remains the delineation of skill versus chance dominance. Bots and strategies serve as mere mechanisms, with countless potential applications and adaptations.

5.6 Summary of Empirical Findings

Our analysis in the above, as summarised in the **Remarks** above, reveals a notable trend: which strategies yield better performance in a game depends on the maximum allowable number of moves. The optimal strategy for a shorter version need not remain optimal in a longer one and vice versa. This underscores the overarching dominance of skill across various game formats due to the following findings.

- The presence of diverse strategies, each with its own level of complexity, consistently outperforming others while maneuvering through potential first or **second** mover advantages.
- The clear advantage of victory-seeking strategies over defeat-seeking ones, highlighting that chance alone cannot propel a defeat-seeking strategy to win randomly.
- The observable improvement in performance with increased learning investment in the game.
- Emergence of alternative strategies (A *vis-a-vis* RP from set I) as winner for longer versions of the game, further highlighting the role of dynamic optimisation (at least intuitively, as in rule based play) in such settings.

- Moreover, it's worth noting that the three-dice iteration of Ludo enhances the role of skill compared to the traditional single-dice version. This is due to the requirement for more comprehensive decision-making in dice selection, which mitigates the impact of chance beyond the user's control, inherent in relying on a single dice value.
- A clearly higher information premium available in algorithmic play (set G) compared to rule based play (set I). This is a very interesting and logical observation as "algorithmic" players are putting more effort in the computation of possibilities given the available information, and hence should be considered more skilled.

The **Remarks** we have made above, related to specific parts of the analysis, are covered in the above points in a comprehensive manner.

6 Robustness Checks

In this section we present several statistical robustness checks, both parametric as well as non-parametric, for our analysis.

6.1 Statistical Significance Check for all the Comparative Performance Statements

Tests of significance are performed based on the summary statistics of the simulated games between the dominant strategies of sets G and I for 16, 20 and 24 turn games. First, the null hypothesis on win percentage $H_0 : p = 0.5$ is tested against the alternative hypothesis $H_0 : p \neq 0.5$, where p denotes the probability of the first player winning the game. Given the large sample size (1000), the distribution of p is considered to be asymptotically normal, and two-tailed test for normal population is performed at 5% level of significance.

P1 Strategy	P2 Strategy	P1 Win %	p-value	Decision
Aggressive	Full Information	55.7	0.0002	Reject H_0
Full Information	Aggressive	38.1	0	Reject H_0
Responsible Pair	Full Information	50.05	0.98	Accept H_0
Full Information	Responsible Pair	40.6	0	Reject H_0

Table 15: Test for Proportions for dominant strategies of both sets for 16 turns

For test of average points earned, the null hypothesis $H_0 : \mu_1 = \mu_2$ is tested against the alternate hypothesis $H_1 : \mu_1 \neq \mu_2$ at 5% level of significance, where μ_1 and μ_2 are the mean points scored by the first and second player in a game respectively. Since sample size is large, paired sample t-test is performed. Since the sample correlation is unknown, but known to be negative for a competitive game, all tests are done for three alternative values of the sample correlation, viz. -1, -0.5 and 0, in order to span all the possibilities.

P1 Strategy	P2 Strategy	P1 Win %	p-value	Decision
Aggressive	Full Information	69.6	0	Reject H_0
Full Information	Aggressive	26.7	0	Reject H_0
Responsible Pair	Full Information	40.2	0	Reject H_0
Full Information	Responsible Pair	49.5	0.75	Accept H_0

Table 16: Test for Proportions for dominant strategies of both sets for 20 turns

P1 Strategy	P2 Strategy	P1 Win %	p-value	Decision
Aggressive	Full Information	75.4	0	Reject H_0
Full Information	Aggressive	19.9	0	Reject H_0
Responsible Pair	Full Information	38.1	0	Reject H_0
Full Information	Responsible Pair	51.7	0.28	Accept H_0

Table 17: Test for Proportions for dominant strategies of both sets for 24 turns

Coming to the results, we discuss the two sets (proportion and average) in the order of number of moves allowed. First note that the order of standard error for proportion is $\sqrt{p(1-p)/n} \sim \sqrt{\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{1000}} \sim 0.016$. Thus, in percentage terms, the critical difference assuming normal distribution is 3.2%. Thus, any difference more than 3.2% is significant. Even if we consider a significance level of 0.001%, the critical difference would be 4.8%. Thus, in tables 15, 16 and 17 all margins are significant even at 0.001% level except for (FI, R) for 20 and 24 turns.

For the average point results when we consider the undominated strategies i.e. {A, R, FI} (refer tables 6, 11, 14), the standard error is at most $106/\sqrt{1000} \sim 3.35$. Thus the 5% level results in tables 18, 19 and 20. Even at 0.001% level, the critical difference would be ~ 10. Hence, the differences that additionally become insignificant are (R, FI) in table 18 and (FI, R) in 19 and 20.

It may be noted that the choice of ρ does not affect the outcome of any of the tests. Thus, even if we calculate the exact standard error adjusting for correlation, the results would remain unchanged.

Scrolling above to the results pertaining to all the strategies and different number of moves (tables 3, 5, 6, 9, 11, 12 and 14), all the results can similarly be easily tested for significance. It is to be noted that all the within group (G or I) results are based on $10K$ simulations, so the standard errors are even more tight ($1/\sqrt{10}$ of earlier values). So the critical difference margin for proportion will be roughly 1.6% (instead of 4.8%) and that for average point will be ~ 3. For example, in Table 3, the second mover's advantage in the (A, A) result is now significant in win percentage but not in average points. All the other results may be assessed thus. As these are routine analyses, we skip the details.

P1 Strategy	P2 Strategy	P1 Mean pts	P2 Mean pts	p ($\rho = 0$)	Decision	p ($\rho = -0.5$)	Decision	p ($\rho = -1$)	Decision
Aggressive	Full Information	117.19	65.87	0	Reject H_0	0	Reject H_0	0	Reject H_0
Full Information	Aggressive	63.6	122.35	0	Reject H_0	0	Reject H_0	0	Reject H_0
Responsible Pair	Full Information	98.97	93.14	0	Reject H_0	0	Reject H_0	0.0002	Reject H_0
Full Information	Responsible Pair	90.32	101.62	0	Reject H_0	0	Reject H_0	0	Reject H_0

Table 18: Test for difference of mean points for dominant strategies of both sets for 16 turns

P1 Strategy	P2 Strategy	P1 Mean pts	P2 Mean pts	p ($\rho = 0$)	Decision	p ($\rho = -0.5$)	Decision	p ($\rho = -1$)	Decision
Aggressive	Full Information	167.11	75.1	0	Reject H_0	0	Reject H_0	0	Reject H_0
Full Information	Aggressive	72.72	167.71	0	Reject H_0	0	Reject H_0	0	Reject H_0
Responsible Pair	Full Information	126.65	129.26	0.17	Accept H_0	0.26	Accept H_0	0.33	Accept H_0
Full Information	Responsible Pair	124.16	132.12	0	Reject H_0	0.0005	Reject H_0	0.003	Reject H_0

Table 19: Test for difference of mean points for dominant strategies of both sets for 20 turns

P1 Strategy	P2 Strategy	P1 Mean pts	P2 Mean pts	p ($\rho = 0$)	Decision	p ($\rho = -0.5$)	Decision	p ($\rho = -1$)	Decision
Aggressive	Full Information	218.98	90.17	0	Reject H_0	0	Reject H_0	0	Reject H_0
Full Information	Aggressive	86.24	224.44	0	Reject H_0	0	Reject H_0	0	Reject H_0
Responsible Pair	Full Information	175.36	177.09	0.51	Accept H_0	0.59	Accept H_0	0.64	Accept H_0
Full Information	Responsible Pair	172.04	180.81	0.0009	Reject H_0	0.0066	Reject H_0	0.019	Reject H_0

Table 20: Test for difference of mean points for dominant strategies of both sets for 24 turns

6.2 Statistical Robustness Check

Before we conclude our discussion, we do a standard robustness check for our empirical results. We use the simulated game play data (10000 games) involving the four strategies {Naive (N) , Aggressive (A), Random Action (Ran), Responsible Pair (RP)}, resulting in 16 combinations. To understand the probability distributional properties of the outcome data we do 1000 bootstrap replications of the data and tabulate the descriptive statistics in Table 21. The results are quite tight in terms of central tendency and dispersion. We have also checked (using the standard Kolmogorov-Smirnoff non-parametric test statistic) goodness of fit with the Normal distribution. The test accepts all 16 combinations (details available on request). Thus, we can safely say that all our probability estimate related comments made above, assuming Normal distribution, are valid.

7 Concluding Remarks

We provided theoretical as well as empirical analyses of the 3-dice, 2-player variant of the Ludo Game with a fixed number of moves (= 24, 30 or 36) above. We have identified a clear second mover's advantage, which is understandable in terms of the rules of the game. It is pertinent in this context to note that many skill games will have a similar kind of advantage. In particular, it is well known that chess has a white pawn (first mover) advantage. Thus, observing a second mover's advantage that does not fade is actually evidence in favor of the importance of skill *vis-a-vis* chance in this Ludo game. We evaluated the game with a set of intuitive rule-driven strategies as well as learning based algorithms.

We observe that certain strategies with their own levels of complexities consistently outperform others while maneuvering through the second mover advantages. We see this phenomena across all versions of the game (24, 30 or 36 moves). However, we identified a clear shift from a conservative strategy (Responsible pair) to more aggression (Aggressive) as the number of moves increased. It is also logically consistent as, for a shorter format, the possibility of promotion is minimal. So, the players are better off avoiding capture. When the games are longer, there are more and repeated (even after capture and hence restart) opportunities for promotion. Thus, the players profitably shift to a more aggressive play, and hence (RP), which is the only Nash equilibrium strategy in the shortest format, no longer remains the only equilibrium choice and eventually makes way for (A) for the second player. Thus, in the interest of a more varied and exciting array of outcomes, we recommend the 30-move game among the three versions considered. This length creates an optimal mix of caution and aggression in the game-play and would attract a richer set of real players.

We have checked the statistical significance (or otherwise) of all the comparative results under the assumption of Normality (a very safe one given the large number of simulations). These statements are often sustained at a level of significance as low as 0.001%! The results are also demonstrated to be consistent even without the Normality assumption through an extensive resampling exercise.

Moreover, learning-based algorithms like MCTS with limited simulation were considered, which showed improvement in results with the duration of the game. This further emphasizes the role of skill and learning in this game. We also noted that there is a clear advantage of victory-seeking strategies over defeat-seeking ones, highlighting that chance alone cannot propel a defeat-seeking strategy to win randomly. Moreover, it's worth noting that Ludo's three-dice iteration enhances the role of skill compared to the traditional single-dice version. This is due to the requirement for more comprehensive decision-making in dice selection, which mitigates the impact of chance beyond the user's control, inherent in relying on a single dice value. To confirm the robustness of our empirical results and the Normality assumption, we did a bootstrap resampling, and the results confirm our assumptions.

These observations in aggregate strongly indicate that in this game, skill as well as progression of skill have a prominent role to play in improving performance.

7.1 Future Scope

In future work, we also plan to analyse a general class of Ludo games with K dice ($K = 3$ or 5), four players and fixed number of moves (alternatives to be evaluated: 24/25 or 30 or 35/36). We will also consider a more general variant with a flexible starting point for the tokens.

It would be interesting to see the application of advanced techniques like Q-learning, a reinforcement learning algorithm that enables computer bots to simulate skill in playing Ludo against humans in this context. By iteratively engaging in game-play, a Q-agent learns optimal strategies, develops a nuanced understanding of game dynamics, and adapts to various scenarios. These Ludo-playing Q-agents can be deployed on commercial gaming platforms to address the skill versus chance debate by providing standardized benchmarks for player performance assessment, thereby enhancing the overall gaming experience and fostering a more engaging environment for players. We'd like to reserve this as a potential future avenue of research.

We have designed an experiment to obtain actual game-playing data from human participants. This will also be added in the next version of the report to present how skill can be sharpened to certain extent with more number of games for actual humans.

References

1. Diganta Mukherjee & Subhamoy Maitra (2023), Unveiling the potential and scope of the Online Skill Gaming Industry: Study with technology students and professionals, report submitted to E-Gaming Federation.

2. Prajit K. Dutta (1999), Strategies and Games: Theory and Practice, The MIT Press.

3. Jeff Sonas (2002), The Sonas Rating Formula — Better Than Elo?. ChessBase.com. 22 October 2002.

4. Gaurav Sood and Derek Willis (2016), How much does the toss really matter?, https://www.espncricinfo.com/blogs/content/story/997931.html

5. Getty, Daniel, Hao Li, Masayuki Yano, Charles Gao, and A. E. Hosoi (2018), "Luck and the Law: Quantifying Chance in Fantasy Sports and Other Contests." SIAM Review 60, no. 4 (January

2018): 869–887. © 2018 SIAM.

6. S. D. Levitt and T. J. Miles (2014), The role of skill versus luck in poker: Evidence from the World Series of Poker, J. Sports Econom., 15, pp. 31–44, https://doi.org/10.1177/1527002512449471.

7. Diganta Mukherjee, Subhamoy Maitra, and Swagatam Das (2023), Role of Skill in the Game of Online Rummy: A Statistical Analysis, March 2023, DOI:10.13140/RG.2.2.12826.11203

8. Adriaan Kalwij and Kris De Jaegher (2023), The age-performance relationship for a cognitive-intensive task: Empirical evidence from chess grandmasters, Sports Economics Review, Volume 2, June 2023, 100010

Descriptives	U-U	U-A	U-Ran	U-R	A-U	A-A	A-Ran	A-R
Mean	4980.577	2103.797	8821.365	1587.512	7532.634	4433.382	9565.087	2756.873
Std. Error	1.6285	1.2749	1.0044	1.1244	1.3879	1.5961	.6436	1.3708
95% CI, LB	4977.381	2101.295	8819.394	1585.306	7529.910	4430.249	9563.824	2754.183
95% CI, UB	4983.772	2106.299	8823.335	1589.718	7535.358	4436.514	9566.350	2759.563
Median	4980.750	2103.500	8821.250	1588.750	7532.500	4432.500	9566.000	2756.500
Std. Deviation	51.4970	40.3170	31.7620	35.5569	43.8901	50.4718	20.3529	43.3491
Minimum	4808.5	1956.0	8712.0	1446.0	7386.5	4279.5	9505.0	2609.07
Maximum	5202.0	2234.0	8908.5	1716.0	7667.0	4578.5	9624.5	2888.5
Contd.	Ran-U	Ran-A	Ran-Ran	Ran-R	R-U	R-A	R-Ran	R-R
Mean	1208.944	327.385	4998.397	3427.692	7515.792	5481.672	6244.956	4961.646
Std. Error	.9892	.5483	1.5850	1.4674	1.3569	1.5382	1.5540	1.5166
95% CI, LB	1207.003	326.309	4995.287	3424.812	7513.129	5478.653	6241.906	4958.670
95% CI, UB	1210.885	328.461	5001.507	3430.571	7518.455	5484.691	6248.005	4964.622
Median	1209.250	327.000	4998.000	3428.500	7515.250	5481.500	6245.000	4960.500
Std. Deviation	31.2798	17.3389	50.1212	46.4031	42.9102	48.6426	49.1420	47.9577
Minimum	1108.0	269.5	4855.5	3282.0	7364.0	5314.0	6112.5	4813.0
Maximum	1312.5	379.5	5163.5	3577.5	7667.5	5642.0	6401.0	5116.0

Table 21: Resampled results, 1000 replications