Statistical Java Gaming Simulation

Timothy Liu
Lynbrook High School, San Jose, CA 95132, USA
timothys.new.email@gmail.com

Mason Chen
Milpitas Christian School. San Jose, CA 95132, USA
mason.chen.training@gmail.com

Joseph Jang
Monta Vista High School, San Jose, CA 95132, USA
jangyeongshinn@gmail.com

Abstract

Our team has designed a special game on which we apply Statistics, Probability, and Java to simulate each game move and predict the winning scenario. We applied binomial probability distribution to build a predictive model that could simulate the gaming sequence between two players. The sample size was determined based on two hypotheses: (1) playing sequence and (2) winning patterns. In this project, we identified four winning patterns and used Java to code these patterns and determine the gaming sequence and consequence based on conditional probability. The Java results were then compared to the Predictive Model to conduct objective root cause analysis for further improvement and optimization. Human behavior was also considered to study the beginner level to the more advanced level. Based on the 2-Proportions Tests, team has achieved > 95% confidence that the optimum model can accurately predict the gaming sequence and winning probability which are verified and validated by Java simulation. Team has been through a systematic Six Sigma DMAIC process, and typical Team Building Cycle (Forming, Storming, Norming, and Performing). This is a good STEM Project for teaching kids on learning and applying Statistics, Java Programming, Problem Solving, and Team Building Dynamics.

Keywords

Java, Statistics, Probability, Predictive Modeling

1. Introduction

The purpose of this project is to design, implement, and test a gaming simulation that will constitute the basis for the development of an Artificial Intelligence (AI) for use in medical research. The use of gaming simulation has been theorized by Toupo and Strogatz to predict nature's evolution [1] and by Fu and Hauert to change in social behaviors [2]. Since there is an increasing amount of medical data available, the team decided to create a simple program that would use eventually use this data to predict medical research outcomes. The concept of this paper is to utilize a simple random variable probability simulated by JAVA programming to predict the outcome of the win/lose scenario. Under certain conditions specified in the game rules, authors were able to uncover a very complicated human behavior on making uncertain decisions in order to survive in the game designed.

2. Design of the "3-chips" Game

We have designed a special 3-chips game to apply the probability in our Statistics Project. This game was designed as following:

1. There are three groups of chips with different number and color in each group as the initial game condition (e.g. 10 Red chips, 8 Yellow chips, 6 Green chips). The initial condition can be randomly assigned as long as there is NO identical number of chips in any two groups such as (X, X, Y).

2. There are two types of players (Player Type A and Player Type B) who will play each other. One player will go first, and then two players will take turns until completed the game. Player Type A is not aware of any game rules. Player Type B is aware of all four game rules.
3. During each round, the player will decide one group (could be Red, Yellow, or Green) and remove at least one chip up to all of the remaining chips from that particular group.
4. The player picked the overall last chip will be the loser of the game.

After designed this 3-chips game, team has found out the following rules to win or to lose a game:

- Rule #1: you will lose the game eventually if you will be the first player removed any group completely like (X, Y, 0) if the remaining two groups have different number ($X \neq Y$) of the remaining chips
- Rule #2: if your opponent removed any group completely such as (X, Y, 0), you will win the game if you can keep the remaining two groups with same number like (X, X, 0) except (1, 1, 0). You can keep this pair pattern until (1, 1, 0) to win the game eventually.
- Rule #3, if your opponent removed any group completely and the remaining two groups have one group with 1 chip and the other group with more than 1 chips (X, 1, 0), you will win the game if you can remove all the chips from that group to make (0, 1, 0). Your opponent will pick the last chip and lost the game.
- Rule #4, if your opponent will make two group have the same number of chips and the third group with different number like (X, X, Y) and $X \neq 1$, you can win by removing the entire Y group to make (X, X, 0) similar to Rule #2 concept.

2.1 Two Main Hypotheses

Team has brainstormed what could be the main statistical hypotheses coming out from above observations:

- Can we use basic probability to predict the winning probability among players who know the above rules or who don't know any of the rules?
- Will the playing sequence (who goes first) impact the winning probability?

2.2 Project Research

Before team would build the predictive model, team has searched the Google Internet and has found some Poker Statistics but could not directly apply the Poker statistics associated with our 3-Chips game. Team has decided to stop searching immediately. Instead, team went to search any Statistics and Probability associated with this "3-Chips" game.

2.3 Explore Basic Statistics and Probability

Team has decided to set up the initial condition as (10, 8, and 6) to build the Statistical Prediction Model. We will simulate the winning % based on 2x2 scenarios in Table 1, similar to Contingency Table [3]:

Table 1. 2x2 Matrix Move Rules

		Go First	
		A	B
Go Second	A	I	IV
	B	III	II

One Player Type A will play with another Player Type A

I. One Player Type B will play with another Player Type B
II. One Player Type B (Go First) will play with one Player Type A (Go Second)
III. One Player Type A (Go First) will play with one Player B (Go Second)

In Case I and Case II, we want to check whether the playing sequence will make any impact on the winning probability. If based on statistics, there should be no bias on winning the game by who will go first. The winning probability should be close to 50% among two players. We will use Java programming to verify this random probability at 50% winning probability.

Case III: Player A (Go First) vs. Player B (Go Second)
Initial condition is still (10, 8, and 6), and player A will pick the first move. Player A can decide which group to pick and pick chips blindly by NOT following the game rules provided earlier.

To demonstrate how Player A will pick his first move safely and likely, we can just assume that Player A will select the group with 10 chips to move on.

Player A has 10 possible choices (after first move, the remaining chips can be from 0 chip to 9 chips). Among 10 choices, Player A will have 3 choices to lose the game immediately as following.

1. (**8**, 8, 6) Rule No. 4
2. (**6**, 8, 6) Rule No. 4
3. (**0**, 8, 6) Rule No. 1

Based on the similar concept, Player A will have total 24 choices (10 + 8 + 6= 24) to pick the first move. Among 24 choices, Player A will have 6 chances (3 + 2 + 1 = 6) to lose the game according to the above four game rules.

Therefore, Player A will have 6/24= 25% probability to lose the game just after the first move. We will assume Player B won't make any mistake by following the provided four rules correctly. Based on the algorithm, we have made the following Probability Tree Diagram in Figure 2 to approximately simulate the winning probability. The tree diagram will assume the following three contingent situations when calculating the winning probability of each single move.

1. Player A will pick chips blindly, and Player B will pick the least chip(s) by following the four Game rules.
2. If Player A could fortunately survive any round and move to the next round, we would assume Player A picked the least chip(s) in that particular round to simulate the best winning scenario for Player A.
3. Both Players will not pick the chip(s) from the smallest group to advance to the next round. We assume both Players are very conservative and avoid taking the chips from the smallest group.

Based on the above scenario, Player A has 6/24= 25% chance to make a mistake and lose the game immediately. If Player A has survived the first pick (75% chance), we will assume that Player A made (9, 8, 6) after the first move. Player B will make (9, 7, 6) to follow Player A's first move in the first round.

In the second round (if Player A survived in the first round), Player A will then have 6 out of 22= 27% chance to lose the second round. Player A will advance to the third round only if Player A will make (8, 7, 6), and Player B will make (8, 5, 6) in the second round based on our best scenario.

In the third round, Player A will then have 6 out of 19= 32% chance to lose the third round. Player A will advance to the fourth round only if Player A will make (7, 5, 6) and Player B will make (7, 5, 4).

In the fourth round, player A will then have 6 out 16= 38% chance to lose the fourth round and we will stop the game simulation after four rounds. Most games should be determined and completed within four rounds.

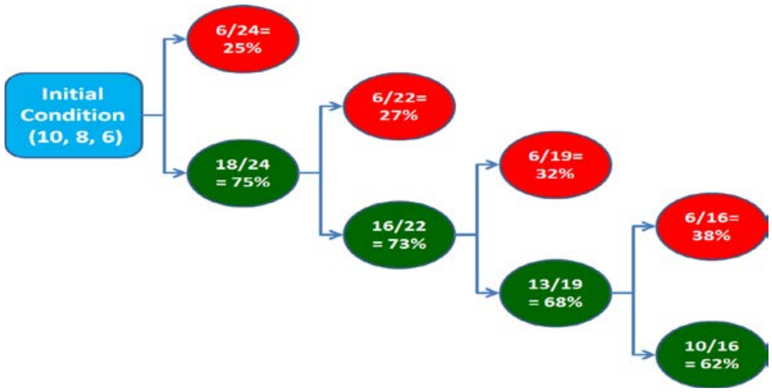

Figure 1. Decision Tree for Probability of Winning

We are assuming all the above events are "Independent" each other and which makes sense in the real case. Based on the conditional Probability P (A and B) = P(A) *P(B)
Then Player A has:

- 25% chance to lose the game immediately after the first round.
- 25% + 75%* 27%= 25% + 20.5%= 45.5% to lose the game after completed the first two rounds
- 25% + 75%* 27% + 75%* 73%* 32%= 25% + 20.5% + 17.2%= 62.7% to lose the game after completed the first three rounds
- 25% + 75%* 27% + 75%* 73%* 32% + 75%* 73%* 68%*38% = 25% + 20.5% + 17.2% + 14%= 76.7% to lose the game after completed the first four rounds
- Our statistical model has predicted, at the best scenario, Player A only has up to 23.3% to win the games over Player B if Player A will GO FIRST.

Case IV: Player B (Go First) vs. Player A (Go Second)
Initial condition is still (10, 8, and 6), and player B will pick the first move. We will duplicate the similar algorithm to predict Player A's winning probability.

Player B will make (9, 8, 6) in the first move. Then, Player A has 6/23= 26% to lose the game in the first round. Following the similar process, we have come out the 2nd Probability Tree Probability Diagram in Figure 2.

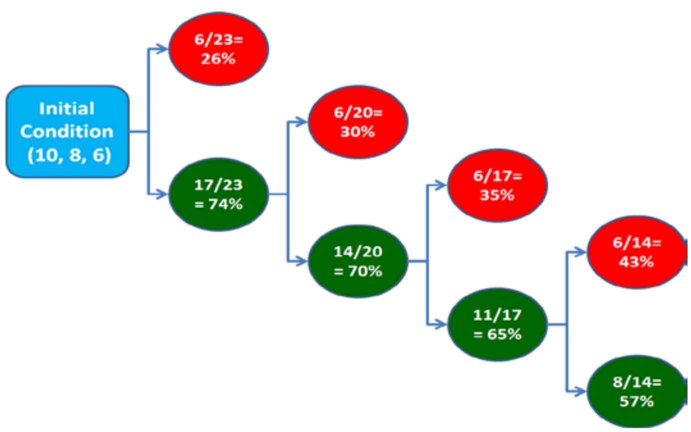

Figure 2. Probability Tree Probability Diagram

In Case IV, Player A has:

- 26% chance to lose the game immediately after the first round.
- 26% + 74%* 30%= 26% + 22.2%= 48.2% to lose the game after completed the first two rounds
- 26% + 74%* 30% + 74%* 70%* 35%= 26% + 22.2% + 18.1%= 66.3% to lose the game after completed the first three rounds

- 26% + 74%* 30% + 74%* 70%* 35% + 74%* 70%* 65%* 43%= 26% + 22.2% + 18.1% + 18%= 84.3% to lose the game after completed the first four rounds
- Our statistical model has predicted, at the best scenario, Player A only have up to 15.7% to win the games over Player B if Player B will go first.

In Summary, based on our Prediction Model:

- In both Cases I, II: either Player has 50% equal chance to win the game
- In Case III: Player A (GO First) has 23.3% chance to win the game
- In Case IV: Player A (GO Second) has 15.7% chance to win the game

Team was very excited about this statistical simulation result on predicting Player A's winning probability. But, our challenge is how to verify or approve our simulation result. We can always get two players types A and B and ask these players to play each other. Though, each game will probably take 10 minutes to complete it. We would first calculate the sample size in order to conduct our two hypothesis tests.

We will use the 2^{nd} Hypothesis to calculate our sample size.
Hypothesis Test #2: will the playing sequence impact the winning probability?
Based on the above simulation, we got 23.3% vs. 15.7% winning probability if the Player sequence is reversed.
We will use the 2-Proportion Binomial Hypothesis Test [4] to simulate the sample size. We are assuming our sample size will be large enough to overcome the 2-Proportion Z test (should verify this assumption later). Then, based on the 2-Proportion Z Statistic, and 95% 2-sided confidence level, we can use the following formula to calculate the sample size.

Assuming the same sample size for two populations:
95% 2-sided confidence interval of two-proportions
$= (P_1 - P_2) \pm Z_{\alpha/2} * [P_1 * (1-P_1)/n_1 + P_2 * (1-P_2)/n_2]$ ^0.5
= (23.3%- 15.7%) ± 1.96* [(23.3%*76.7%)/n + (15.7%*84.3%)/n] ^0.5

To reject the Null Hypothesis: Player A Winning Probability = Player B Winning Probability, the 2-proportion confidence interval should not cover Zero. Therefore, we can calculate the sample size n = 209 as indicated in Table 2.

Test for Two Proportions
Testing comparison p = baseline p (versus ≠)
Calculating power for baseline p = **0.157** $\alpha = 0.05$

Table 2. Sample size and Associated Winning Probability

Comparison	Size	Power	Actual Power
0.233	**209**	0.5	0.500418

Before we will adopt sample size= 209, we need to check two assumptions in order to use the 2-Proportions Z Approximation (two symmetric requirements):

- **np > 10**, both 209*23.3% and 209*15.7% > 10
- **nq > 10**, both 209*76.7% and 209*84.3% > 10

We have met both the Z Approximation Criteria and we can use Sample Size = 209.
Each game will take us **10 mins and 209*4** cases games will take us more than **140** hours to collect the real data. Team could not afford to take this large sample size. Fortunately, our two team members are just learning Java programming and team has decided to develop a quick Java programming to simulate this gaming simulation.

3. Design Java Programming

Java will take only 2 seconds to decide a game result, total less than 1 hour to complete all four cases. We have developed Java programming to simulate all four cases in Figures 3-5.

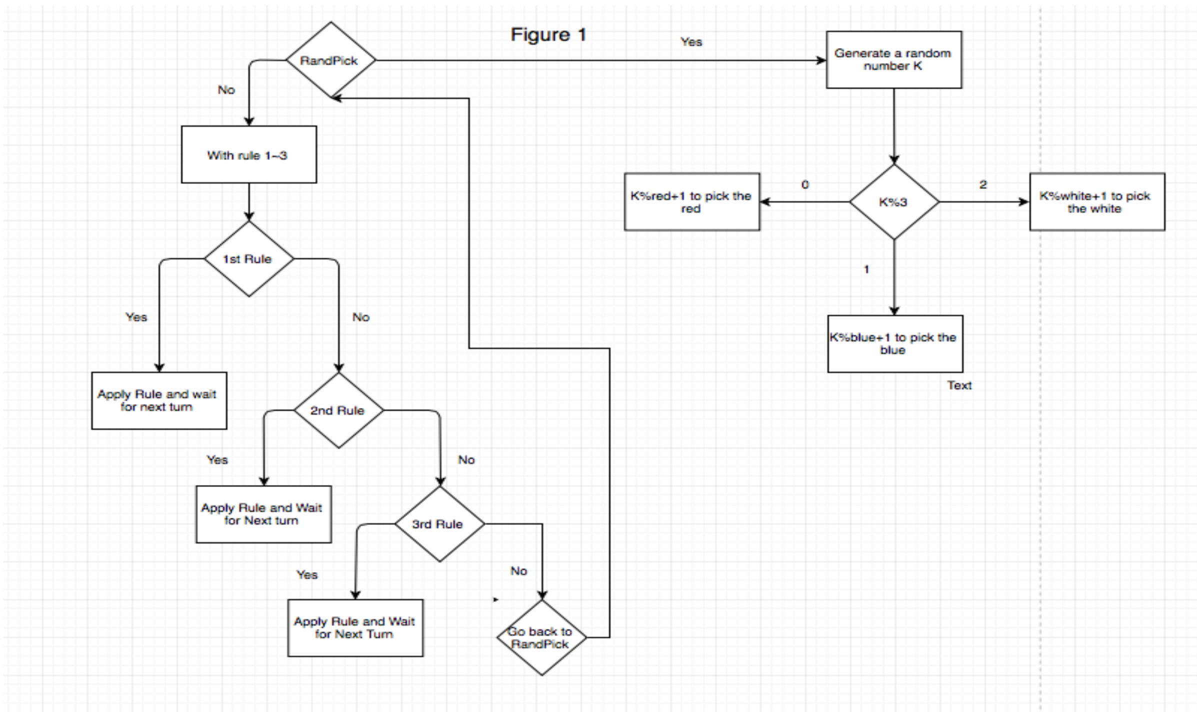

Figure 3. Java Flowchart

Before developing Java programming, our team has brainstormed the following Programming Flow Chart to lay out the Java programming modules based on the four game rules mentioned previously:

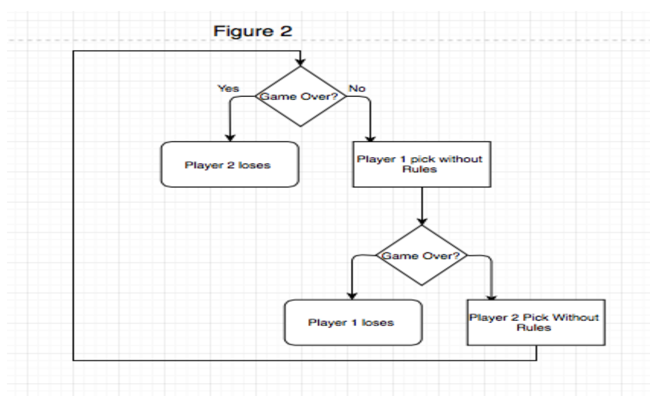

Figure 4. Java Flowchart

```
import Java.util.Random;
public class Chip{
 public int r;  public int g;   public int b;
 public void Sort(){
   int i=r;
   int j=g;
   int k=b;
   r=Math.min(Math.min(i,j),k);
   b=Math.max(Math.max(i,j),k);
   g=Math.min(Math.min(Math.max(i,j), Math.max(j,k)),
Math.max(i,k));
```

```
public boolean Rule12(){
   if(r==0)
     if(g>1 && b>g){ b=g; return true;}
     else if(g==1) { b=0; return true;}
     else if(g==0 && b>1) { b=1; return true;}
   return false;
 }
 public boolean Rule3(){
   if(r==g && r>1) { b=0; return true;}
   else if(g==b && g>1 && r>0) { r=0; return true;}
   return false;
```

```
}
public boolean GameOver(){
  if (r==0 && g==0 && b==0)
    return true;
  return false;
}
public Chip(int i, int j, int k) {
  r=Math.min(Math.min(i,j),k);
  b=Math.max(Math.max(i,j),k);
  g=Math.min(Math.min(Math.max(i,j), Math.max(j,k)),
Math.max(i,k));
}
public void RandPick(){
  Random rn = new Random();
  int pick = rn.nextInt(1000);
  if(pick%3 == 0 && r > 0){
    r-=pick%r + 1;
  }else if(pick%3 == 1 && g>0){
    g-=pick%g + 1;
  }else{
    if(list.GameOver()){
    System.out.println("Player A lose");
    break;
    }
     list.Pick();      // Player B pick with rule 1~3
    list.Sort();
}
b-=pick%b + 1;
  }
}
```

```
}
public void Pick(){
  if( Rule12() || Rule3() ) ;
  else RandPick();
}
  public static void main(String[] args) {
int x = Integer.parseInt(args[0]);
int y = Integer.parseInt(args[1]);


int z = Integer.parseInt(args[2]);
Chip list=new Chip(x,y,z);
System.out.println("Input is " + list.r + ", " + list.g + ", " +
list.b);
//Random Pick
//list.RandPick();
//Apply rules
//list.Pick();
while(true){
   list.RandPick();    // Player A random pick
   list.Sort();
   System.out.println("Player A pick " + list.r + ", " + list.g +
", " + list.b);
 System.out.println("Player B pick " + list.r + ", " + list.g + ",
" + list.b);
   if(list.GameOver()){
    System.out.println("Player B lose");
    break;
   }
  }
 }
```

Figure 5. Java Code

3. 1 Collect Java Raw Data and Result (209 samples of each case)

Next step is to conduct the hypotheses after collected JAVA raw data on four different cases:

Case I: two Type A players played each other. Among 209 samples: first Player A won 107 times and the second Player A won 102 times. We will conduct 2-sided 1-Proportion Test (not 2-Proportions Test) since all the data was from one Sample.

- We conducted a Minitab 1-Proportion Test in Table 3, and the Null Hypothesis H_0: Player A Winning Probability= 0.5 (50%). Team used Normal Approximation method to conduct 1-Proportion Z test.
- The P-value is 0.729 > 0.05, which failed to reject the Null Hypothesis. This result has indicated the playing sequence has not made significant impact on the winning probability between two Type A players.

Table 3. Hypothesis Test and CI for One Proportion

Test of $p = 0.5$ vs $p \neq 0.5$

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	107	209	0.511962	(0.444194, 0.579729)	0.35	**0.729**

Case II: two Type B players played each other. Among 209 samples: first Player B won **109** times and the second Player won **100** times. We will conduct 2-sided 1-Proportion Test (not 2-Proportions Test) since all the data was from one Sample.

- We conducted a Minitab 1-Proportion Test, and the Null Hypothesis H_0: Player B Winning Probability= 0.5 (50%). Team used Normal Approximation method to conduct 1-Proportion Z test.
- The P-value is 0.534 > 0.05, which failed to reject the Null Hypothesis. This result has indicated the playing sequence has not made significant impact on the winning probability between two Type B players.

Case III: Player A (Go First) played with Player B (Go Second). Among 209 samples: Player A only won 7 times and Play B won 202 times. Based on our Case III prediction, we would predict Player A should win 23.3%. Team has conducted 1-Proportion Test and the Null Hypothesis H_0: Player A Winning Probability= 0.233 (23.3%) in Table 4.

- P-Value is 0.000 and we should reject Null Hypothesis which has indicated our Case III prediction model is not validated through our Java simulation.

Table 4. Hypothesis Test and CI for One Proportion

Test of p = 0.233 vs p ≠ 0.233

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	7	209	0.033493	(0.009100, 0.057885)	-6.82	0.000

- However, we don't meet the Normal Approximate Criteria since Player A only won 7 times, which violates the np requirement > 10. We could not use the 1-Proportion Z test. Instead, we use the Minitab 1-Proportion Exact Test in Table 5. Exact test is based on the Binomial Distribution to calculate the Exact P-Value. P_Value is still 0.000 which indicated that we should reject Null Hypothesis which has indicated our Case III prediction is not validated through our Java simulation. We will address this issue later.

Table 5: Hypothesis Test and CI for One Proportion

Test of p = 0.233 vs p ≠ 0.233

Sample	X	N	Sample p	95% CI	P-Value
1	7	209	0.033493	(0.01357, 0.06779)	0.000

Case IV: Player A (Go Second) played with Player B (Go First). Among 209 samples: Player A only won 6 times and Play B won 203 times. Based on our Case IV prediction, we would predict Player A should win 15.7%. Team has conducted 1-Proportion Test and the Null Hypothesis H_0: Player A Winning Probability= 0.157 (15.7%) in Table 6.

- P-Value is 0.000 and we should reject Null Hypothesis which has indicated our Case III prediction is not validated through our Java simulation. We will address this issue later.

Table 6: Hypothesis Test and CI for One Proportion

Test of p = 0.157 vs p ≠ 0.157

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	6	209	0.028708	(0.006069, 0.051347)	-5.10	0.000

- However, we don't meet the Normal Approximate Criteria since Player A only won 6 times, which violates the np requirement > 10. We could not use the 1-Proportion Z test. Instead, we use the Minitab 1-Proportion Exact Test in Table 7. Exact test is based on the Binomial Distribution to calculate the Exact P-Value. P_Value is still 0.000 which indicated that we should reject Null Hypothesis which has indicated our Case III prediction is not validated through our Java simulation. We will address this issue later.

Table 7: Test and CI for One Proportion

Test of p = 0.157 vs p ≠ 0.157

Sample	X	N	Sample p	95% CI	P-Value
1	6	209	0.028708	(0.01061, 0.06144)	0.000

3.2 Analyze Java Result

Java results have supported our CASE I and CASE II non-bias result on which player would GO First. There is no significant bias observed regarding the playing sequence would impact the winning probability. However, in Case III and Case IV, our prediction model is not very reliable to predict the Java results. The biggest reason of failing the prediction is that we assumed each game will be completed within four rounds. If both players are very conservative, this assumption will be very questionable. In order to further improve the prediction capability, we will expand current four-round modeling to five-round or six-round to improve our prediction capability.

4. Improve Phase

For Case III, we have expanded predictive model from previous Four-Rounds to Five rounds in order to improve the prediction accuracy shown in Figure 6.

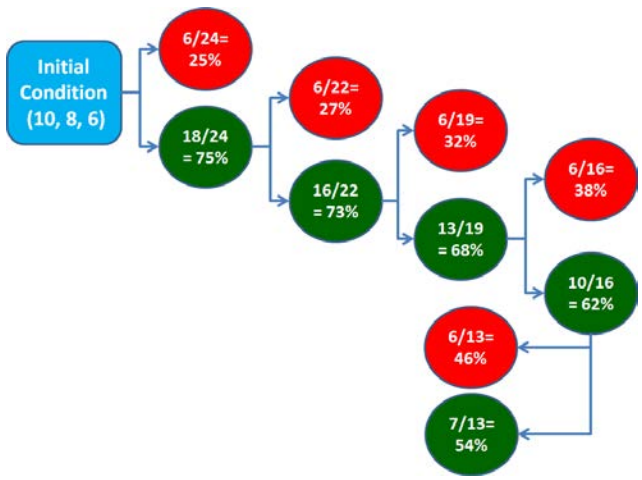

Figure 6: Probability Tree of Five Rounds

- Player A has losing probability= 25% + 75%* 27% + 75%* 73%* 32% + 75%* 73%* 68%*38% + 75%* 73%* 68%*62%*46% = 25% + 20.5% + 17.2% + 14% + 10.6%= 87.3% to lose the game after completed the first four rounds
- **Our statistical model has predicted, at the best scenario, Player A only have up to 12.7% to win the games over Player B if Player A would GO FIRST.**
- Team has conducted another 1-Proportion Z test in Table 8, and Exact Test in Table 9. Both P-Values are still 0.000 and rejected our Case III Model Prediction. Team needs to advance to the next 6-Round Model.

Table 8. Hypothesis Test and CI for One Proportion

Test of p = 0.127 vs p ≠ 0.127

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	7	209	0.033493	(0.009100, 0.057885)	-4.06	0.000

Table 9. Hypothesis Test and CI for One Proportion

Test of p = 0.127 vs p ≠ 0.127

Sample	X	N	Sample p	95% CI	P-Value
1	7	209	0.033493	(0.013570, 0.067788)	0.000

In Figure 7 Six Round Modeling, Player A will have losing probability= 25% + 75%* 27% + 75%* 73%* 32% + 75%* 73%* 68%*38% + 75%* 73%* 68%* 62%* 46% + 75%* 73%* 68%* 62%* 54%* 60% = 25% + 20.5% + 17.2% + 14% + 10.6% + 7.5%= 94.8% to lose the game after completed the first four rounds.

- Our statistical model has predicted, at the best scenario, Player A only have up to 5.2% to win the games over Player B if Player A would GO FIRST.
- Team has conducted another 1-Proportion Z test in Table 10 and Exact Test in Table 11. P-Values are 0.228 and 0.225 > 0.05 and which failed to reject our Case III Model Prediction**.**

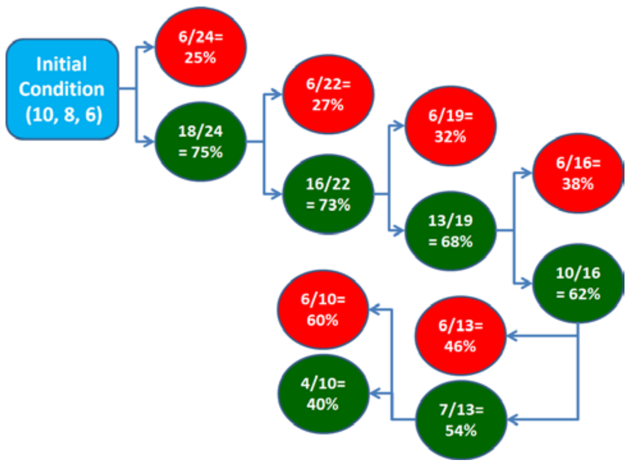

Figure 7. Probability Tree of Six Rounds

Table 10. Hypothesis Test and CI for One Proportion

Test of $p = 0.052$ vs $p \neq 0.052$

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	7	209	0.033493	(0.009100, 0.057885)	-1.21	0.228

Table 11. Hypothesis Test and CI for One Proportion

Test of $p = 0.052$ vs $p \neq 0.052$

Sample	X	N	Sample p	95% CI	P-Value
1	7	209	0.033493	(0.013570, 0.067788)	0.225

For Case IV, we have expanded predictive model from previous Four-Round to Five round shown in Table 12:

- Player A has 5.1% losing probability based on 5-Round Model shown in Table 13.
- Team has conducted 1-Proportional Z test in Figure 14a and Exact Test in Figure 14b. P-Values are 0.143 and 0.128 > 0.05. Team has failed to reject our Model IV 5-Round Predictive Modeling. Team can stop here and won't need to advance to Round 6.

Table 12. Hypothesis Test and CI for One Proportion

Test of $p = 0.051$ vs $p \neq 0.051$

Sample	X	N	Sample p	95% CI	Z-Value	P-Value
1	6	209	0.028708	(0.006069, 0.051347)	-1.46	0.143

Table 13. Hypothesis Test and CI for One Proportion

Test of $p = 0.051$ vs $p \neq 0.051$

Sample	X	N	Sample p	95% CI	P-Value
1	6	209	0.028708	(0.010607, 0.061435)	0.128

5. Conclusion

Team has successfully built a predictive model to simulate the winning probability on four Cases. There is no significant evidence showing the playing sequence would impact the winning result. This result is making sense since we are assuming all events are independent. This independency should be more accurate when we have more chips in the pool. Player B (knowing four rules) has a much higher winning probability (> 95% chance) over Play A (playing blindly). Our predictive model can accurately predict the winning probability if we can take 5 or 6 rounds. Team has conducted the sample size calculation in order to draw a statistical conclusion to verify the two hypotheses. Developing a Java programming has significantly reduced our effort to collect data to validate our predictive model.

6. Future Work Opportunities

Team has built a very basic Java model to simulate the Powerball Probability. Team could have done it better on the following future opportunities:

- Consider the Prize Model to adjust the expected value and the probability uniformity across bigger prizes
- Search the historical tickets-sold amount distribution to more accurately simulate the No-Jackpot probability
- Analyze the Roll-Over pattern on the tickets sold amount to more accurately predict the Jackpot Amount distributions
- Build a model of two Mega Balls to create an even bigger Jackpot.
- Compare Powerball to other Lotto like Mega Millions or Super Lotto

Acknowledgements

Need to acknowledge Dr. Charles Chen and Dr. Ying Huang here for the guidance and mentorship here….

References

[1] Toupo D., and Strogatz S., Nonlinear dynamics of the rock-paper-scissors game with mutations, Physical Review E, Published 11 May 2015.

[2] Maciejeweski W., Fu, F., and Hauert, C., Evolutionary Game Dynamics in Populations with Heterogenous Structures, PLOS Computational Biology, published April 24, 2014.

[3] Gokhale, D. V.; Kullback, Solomon (1978). The Information in Contingency Tables. Marcel Dekker. ISBN 0-824-76698-9

[4] Montgomery D., Runger G. (2003). Applied Statistics and Probability for Engineers. ISBN 9780471745891. 2-Proportions Test P.379-383

Biography

Timothy Liu has certified IBM SPSS Statistics Certificate, Modeler Data Analysis and Data Mining Certificates. He is wrestling club and won the 2nd place 2016 Shuai Jiao Teen open at the regional competition. He is the co-author of 3 books: “Mindy: The Tropical Trio” ISBN 9781622280001, “Mindy: Friend or Foe?” ISBN 9781537301792, and “Spellbound Story” ISBN 9781492715856.

Mason Chen is a certified IASSC Black Belt and also certified IBM SPSS Statistics Certificate, IBM Modeler Data Analysis and Data Mining Certificates. Mason is familiar with Lego Robotics/EV3, Six Sigma DMAIC, DMADOV, Lean Production, Minitab, SPSS Statistics, SPSS Modeler CRISP Data Mining, and Applied Statistics.

Joseph Jang has been learning Java. He has certified IBM SPSS Statistics Certificate. Joseph got invited and presented his JAVA Project in the American Society for Quality Statistics and Reliability Group. He was also part of a relay team and won second place at the regional competition.