Math 381 A
Sheng Yu

Homework #6 Writeup: Simulation on Two-Person Dice Game

I. Introduction:

We have two players A and B play a dice game. The rule is very simple: two players alternately roll a die, and they can choose to either add the roll to their current score or subtract it from the other player's score. Player's score will never below zero, so player's score will set to zero at the beginning of the game or whenever their current score is below zero. Since A and B both love the number 3, they decide to set the limit as 33. The first player reaches or exceeds the limit 33 wins the game. In here, we will simulate the game and try to implement different strategies to check if they can potentially increase the win ratio. Also, we will be applying the Central Limit Theorem to our results to see if the game illustrates the normal distribution.

II. Simulation:

We will start with the most basic strategy for the simulation: player will always add the roll to their current score. This is also the most intuitive strategy since the goal to win the game is to get to the limit 33 first so we want to add as many rolls to the score as possible. We will simulate the game with this strategy for 1000 runs of 100000 simulated games. For each game, we decide who is first by flip the coin, so A and B have equal opportunity to start the game first. In MATLAB, we use random number generator to randomly generate 0 or 1 which simulates the process of flip the coin. Head refers 0 for A to start first, and tail refers 1 for B to start first. I will use MATLAB for the simulation.

The code we have is:

```
% Set up

run = 1000;
game = 100000;
limit = 33;

% To record win ratio of A and B for each run

winRatio_A = zeros(1, run);
winRatio_B = zeros(1, run);

% To see whoes turn it is

A = 0;  % Head
B = 1;  % Tail

for i = 1:run
    num_A_win = 0;
    num_B_win = 0;
    for j = 1:game
        score_A = 0;
        score_B = 0;
        % randomly decide first hand is A or B (flip a coin)
        player = round(rand);
        while score_A < limit && score_B < limit
            % random roll
            roll = randi(6);
            % A's turn
            if player == A
                score_A = score_A + roll;
            end
            % B's turn
            if player == B
                score_B = score_B + roll;
            end
            % Switch player
            player = 1 - player;
            % See who wins
            if score_A >= limit
                num_A_win = num_A_win + 1;
                break;
            end
            if score_B >= limit
                num_B_win = num_B_win + 1;
                break;
            end
        end
    end
    winRatio_A(1, i) = (num_A_win / game);
    winRatio_B(1, i) = (num_B_win / game);
end
```

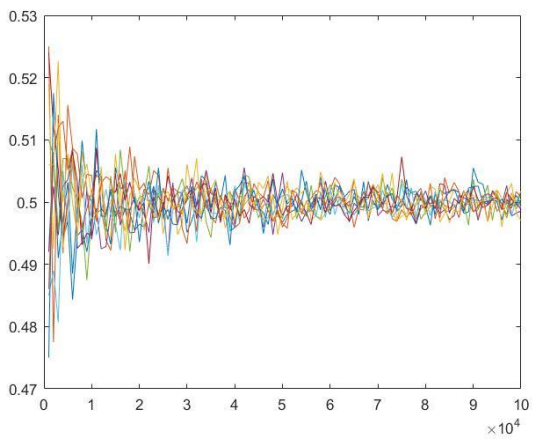
Before we proceed, we want to see if the 100000 simulations per run is sufficient or near sufficient for convergence. To see that, we will have ten runs and plot them in a same plot.

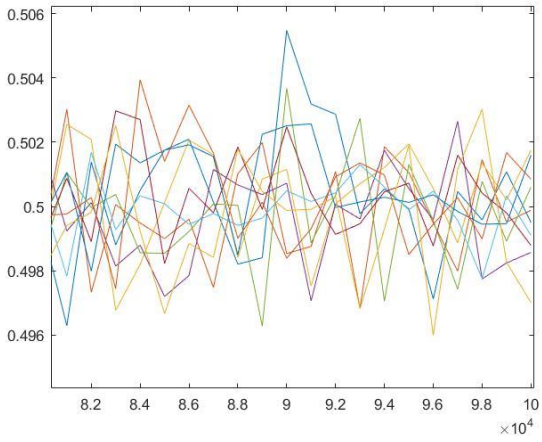
In MATLAB, we can do the following modification to make such plot:

```
% Set up
run = 10;
game = 1000:1000:100000;
limit = 33;

% To record win ratio of A and B for each run, and the average
% number of round to end the game for each run.
winRatio_A = zeros(run, length(game));
winRatio_B = zeros(run, length(game));

% To see whoes turn it is
A = 0;  % Head
B = 1;  % Tail
for i = 1:10
    for k = 1:length(game)
        num_A_win = 0;
        num_B_win = 0;
        round_to_win = zeros(1, game(k));
        for j = 1:game(k)
            score_A = 0;
            score_B = 0;
            % randomly decide first hand is A or B (flip a coin)
            player = round(rand);
            while score_A < limit && score_B < limit
                % random roll
                roll = randi(6);
                % A's turn
                if player == A
                    score_A = score_A + roll;
                end
                % B's turn
                if player == B
                    score_B = score_B + roll;
                end
                % Switch player
                player = 1 - player;
                % See who wins
                if score_A >= limit
                    num_A_win = num_A_win + 1;
                    break;
                end
                if score_B >= limit
                    num_B_win = num_B_win + 1;
                    break;
                end
            end
        end
        winRatio_A(i, k) = (num_A_win / game(k));
        winRatio_B(i, k) = (num_B_win / game(k));
    end
end
for i = 1:run
    tempA = winRatio_A(i, :);
    plot(game, tempA);
    hold on
end
for i = 1:run
    tempB = winRatio_B(i, :);
    plot(game, tempB);
    hold on
end
```

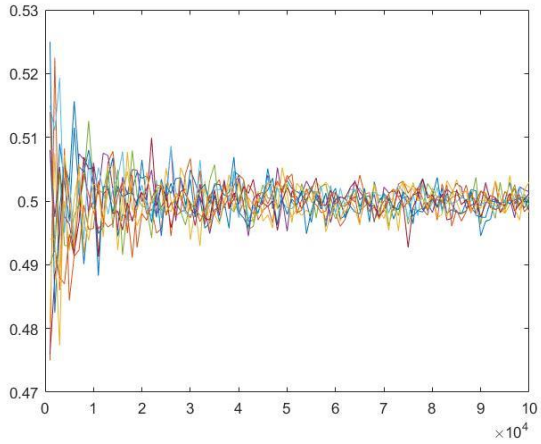
We start with simulation 1000 and all the way up to 100000 to see if 100000 is sufficient in here.

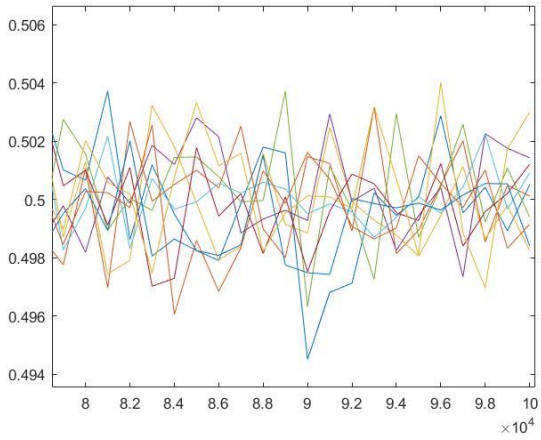
We can get the following plot for player A:





And for player B:





For this case, since both players are using the same strategy, the result is not very enlightening as the probability of winning the game would be 50%. In here, we just want to demonstrate how to draw such plots and to prove the 100000 runs is sufficient. In the next section where we use different strategies on player A and B, we will do this again.

III. Other Strategies:

We will refer the default strategy: players always add the roll to their current score as **Strategy 0.**

Convention Strategy:

If A and B's current score is within 6 from the limit 33, then they are much likely to win the game with next roll, so A and B would check this case and will only subtract each other's score under such condition. We will add this convention strategy with all the strategies below.

Strategy I:

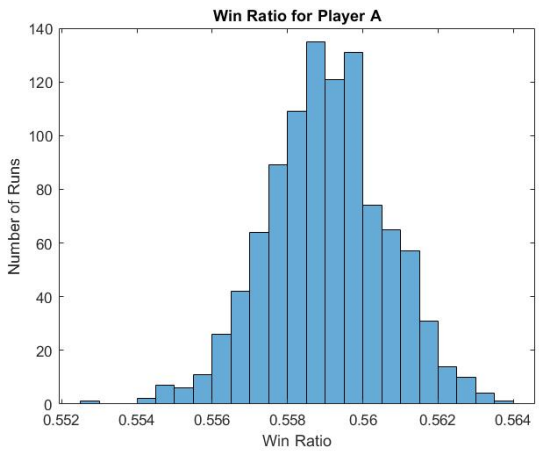
This strategy is identical as **Strategy 0** except whenever B's score is more than 5 more than A's score, A will only subtract the roll to B's score. Player B would use **Strategy** 0 as a comparison. The only difference in the code is the way we design the case when it is A or B's turn in the while loop, everything else is the same:

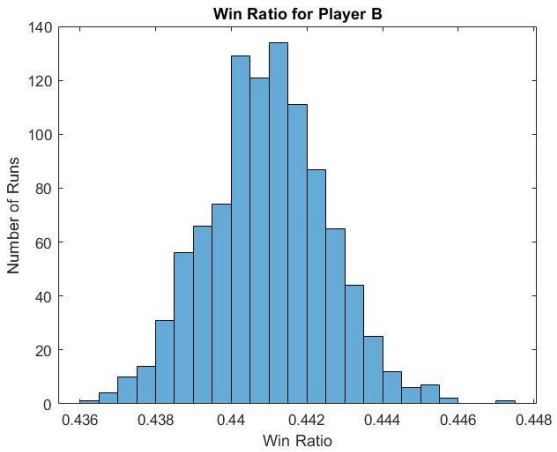
```
% A's turn
if player == A
    if limit - score_B <= 6
        score_B = score_B - roll;
    else
        if score_B - score_A >= 5
            score_B = score_B - roll;
            if score_B < 0
                score_B = 0;
            end
        else
            score_A = score_A + roll;
        end
    end
end
% B's turn
if player == B
    if limit - score_A <= 6
        score_A = score_A - roll;
    else
        score_B = score_B + roll;
    end
end
```

With the same code as before, we can generate the following histogram, because for the **Strategy 0**, the histogram peaks with win ratio around 0.50, so we will include this one as an unconventional example. We did 1000 runs with 100000 simulated games per run. We can draw the histograms with the following code:

```
figure;
histogram(winRatio_A);
title("Win Ratio for Player A");
xlabel("Win Ratio"); ylabel("Number of Runs");
figure
histogram(winRatio_B);
title("Win Ratio for Player B");
xlabel("Win Ratio"); ylabel("Number of Runs");
```

It gives us:

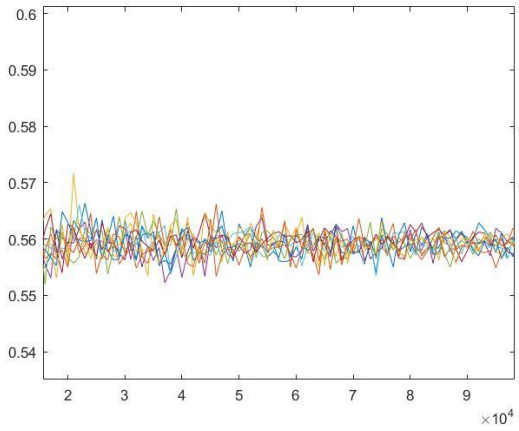


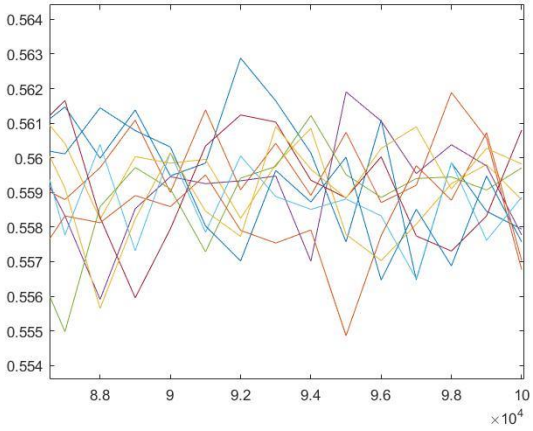


With the histograms above, we see that the win ratio for player A and B both have a peak value, and the histograms are symmetric about their peak. They look very like the shape of a normal distribution. With the Central Limit Theorem, it suggests that the distribution of these values is very nearly normal.

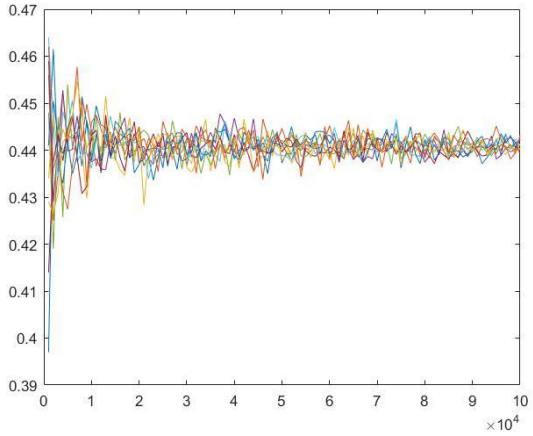
Before we proceed, we want to prove that 100000 games per run is sufficient as the result would converge. In section II, we introduced the way to draw plots to demonstrate convergence, follow that code, we can get:

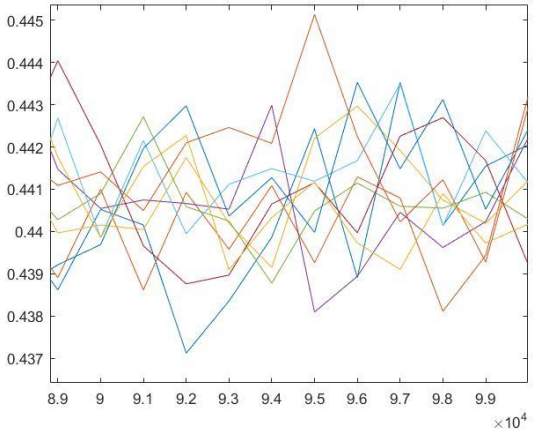
For player A:





For player B:





We can see a clear convergence from the plots above. So, for the rest simulations, we will keep using 100000 games per run.

Now, to calculate the mean $\bar{x}$ and the sample standard deviation s:

```
% Calculate mean and sample standard deviation

mean_winRatioA = mean(winRatio_A);    % 0.559321500000000
mean_winRatioB = mean(winRatio_B);    % 0.440678500000000
sd_winRatioA = std(winRatio_A);       % 0.001674480094196
sd_winRatioB = std(winRatio_B);       % 0.001674480094196
```

For a true normal distribution, 99.73% of the values of the population will be within 3 standard deviations of the mean. We can calculate such confidence interval with the following code:

```
% Confidence Interval: 99.73%

upper_P_winRatioA = mean_winRatioA + 3 * sd_winRatioA;  % 0.564344940282588
lower_P_winRatioA = mean_winRatioA - 3 * sd_winRatioA;  % 0.554298059717412
upper_P_winRatioB = mean_winRatioB + 3 * sd_winRatioB;  % 0.445701940282588
lower_P_winRatioB = mean_winRatioB - 3 * sd_winRatioB;  % 0.435655059717412
```

Let p_A and p_B denotes the true probability lies between $\bar{x} - 3s$ and $\bar{x} + 3s$. For a cleaner writeup demonstration, we will only include four decimal places, we get:

$$0.5543 < p_A < 0.5643$$

$$0.4357 < p_B < 0.4457$$

We can see that both from the confidence interval and the histogram, A's win ratio increased by playing the game with this new strategy. But for the same idea, what happens to the change of win ratio if we change the difference threshold 5 to a higher or lower number?

We can make a table, and we will simulate 100 runs instead of 1000 runs for time consideration:

Difference Threshold	99.73% Confidence Interval for A	99.73% Confidence Interval for B
1	$0.3033 < p_A < 0.3123$	$0.6877 < p_B < 0.6967$
2	$0.5492 < p_A < 0.5591$	$0.4409 < p_B < 0.4508$
3	$0.5738 < p_A < 0.5833$	$0.4167 < p_B < 0.4262$
4	$0.5680 < p_A < 0.5781$	$0.4219 < p_B < 0.4320$
5	$0.5543 < p_A < 0.5643$	$0.4357 < p_B < 0.4457$
6	$0.5394 < p_A < 0.5476$	$0.4524 < p_B < 0.4606$
7	$0.5259 < p_A < 0.5347$	$0.4653 < p_B < 0.4741$
8	$0.5161 < p_A < 0.5252$	$0.4748 < p_B < 0.4839$
9	$0.5082 < p_A < 0.5180$	$0.4820 < p_B < 0.4918$
10	$0.5033 < p_A < 0.5124$	$0.4876 < p_B < 0.4967$

Table-1: A with Strategy I with different difference threshold; B with Strategy 0

We can see from the table that for the difference threshold 3, it seems to be the more effective for improving the probability of winning the game comparing with other difference thresholds. However, we cannot be certain since the confidence intervals overlap to each other for different thresholds. But for now, the table makes sense because if we set the difference threshold too small such as 1, then it happens too frequently that most of the time A would focus on subtracting B's score instead of adding to A's own score. And if we set the difference threshold too big, then this part of the algorithm would rarely kick in so that the strategies are essentially the same for player A and B.

Strategy II:

Now, we will implement a new strategy for player A. We will have player B stays the same. For player A, **Strategy II** will have A to reach a certain goal first, then follows **Strategy I** with difference threshold 3. Again, we include the **Convention Strategy** for both players. We modify the code in the while loop, since B's behavior is the same, we only modify A:

```
% A's turn
if player == A
    if score_A < goal       % goal is defined outside the loop
        score_A = score_A + roll;
    else
        if limit - score_B <= 6
            score_B = score_B - roll;
        else
            if score_B - score_A >= 3
                score_B = score_B - roll;
                if score_B < 0
                    score_B = 0;
                end
            else
                score_A = score_A + roll;
            end
        end
    end
end
```

We can generate the result in a table:

Goal	99.73% Confidence Interval for A	99.73% Confidence Interval for B
1	$0.5810 < p_A < 0.5903$	$0.4097 < p_B < 0.4190$
2	$0.5805 < p_A < 0.5896$	$0.4104 < p_B < 0.4195$
3	$0.5791 < p_A < 0.5882$	$0.4118 < p_B < 0.4209$
4	$0.5768 < p_A < 0.5865$	$0.4135 < p_B < 0.4232$
5	$0.5741 < p_A < 0.5835$	$0.4165 < p_B < 0.4259$
6	$0.5684 < p_A < 0.5780$	$0.4220 < p_B < 0.4316$
7	$0.5698 < p_A < 0.5757$	$0.4243 < p_B < 0.4302$
8	$0.5666 < p_A < 0.5722$	$0.4278 < p_B < 0.4334$
9	$0.5621 < p_A < 0.5726$	$0.4274 < p_B < 0.4379$
10	$0.5588 < p_A < 0.5674$	$0.4326 < p_B < 0.4412$
12	$0.5491 < p_A < 0.5604$	$0.4396 < p_B < 0.4509$
16	$0.5284 < p_A < 0.5377$	$0.4623 < p_B < 0.4716$
20	$0.4986 < p_A < 0.5084$	$0.4916 < p_B < 0.5014$
24	$0.4677 < p_A < 0.4761$	$0.5239 < p_B < 0.5323$

Table-2: A with Strategy II with different goal values; B with Strategy 0

We see from the table; **Strategy II** is overall more effective than **Strategy** 0 for player to win the game. Let us investigate more with **Strategy II** comparing to **Strategy I**. We will have player B plays the game with **Strategy I** with difference threshold 3, and player A stays unchanged:

Goal	99.73% Confidence Interval for A	99.73% Confidence Interval for B
1	$0.5035 < p_A < 0.5140$	$0.4860 < p_B < 0.4965$
2	$0.5067 < p_A < 0.5130$	$0.4870 < p_B < 0.4933$
3	$0.5036 < p_A < 0.5139$	$0.4861 < p_B < 0.4964$
4	$0.5009 < p_A < 0.5104$	$0.4896 < p_B < 0.4991$
5	$0.4978 < p_A < 0.5101$	$0.4899 < p_B < 0.5022$
6	$0.4965 < p_A < 0.5068$	$0.4932 < p_B < 0.5035$
7	$0.4943 < p_A < 0.5045$	$0.4955 < p_B < 0.5057$
8	$0.4938 < p_A < 0.5005$	$0.4995 < p_B < 0.5062$
9	$0.4912 < p_A < 0.4967$	$0.5033 < p_B < 0.5088$
10	$0.4859 < p_A < 0.4955$	$0.5045 < p_B < 0.5141$
12	$0.4779 < p_A < 0.4877$	$0.5123 < p_B < 0.5221$

Table-3: A with Strategy II with different goal values; B with Strategy I difference threshold 3

We see from the table, only with small goal values such as 1, 2, 3 or 4, **Strategy II** is slightly better than **Strategy I**. But the overall benefits are very negligible. For goal 1, 2, 3, or 4, player A could achieve the goal in one toss which leaves A and B basically using the same strategy.

Strategy III:

Strategy I is still the best among the ones we found. The basic idea is to reduce the score gap between player A and B. For this strategy, player B uses **Strategy I** with difference threshold 3 for comparison. Player A uses **Strategy III** which is identical to **Strategy I** with a difference threshold of 3 except we choose to only subtract B's score with a special set of roll values.

For example, suppose we choose our special set of roll values be {5}. Whenever B's score is 3 or more than A's score, A rolls the die, and if A rolls 1, 2, 3, 4 or 6, then A would add the roll to A's score. If A rolls 5, then A would subtract B's score with 5.

We only need to modify the code in the while loop (shown below is the case as the example):

```
% A's turn
if player == A
    if limit - score_B <= 6
        score_B = score_B - roll;
    else
        if score_B - score_A >= 3
            if roll == 5
                score_B = score_B - roll;
                if score_B < 0
                    score_B = 0;
                end
            else
                score_A = score_A + roll;
            end
        else
            score_A = score_A + roll;
        end
    end
end
```

We can get a table for different special set values:

Special Set Values	99.73% Confidence Interval for A	99.73% Confidence Interval for B
{4}	$0.4215 < p_A < 0.4288$	$0.5712 < p_B < 0.5785$
{5}	$0.4161 < p_A < 0.4231$	$0.5769 < p_B < 0.5839$
{6}	$0.4040 < p_A < 0.4151$	$0.5849 < p_B < 0.5960$
{4, 5}	$0.4169 < p_A < 0.4316$	$0.5709 < p_B < 0.5794$
{4, 6}	$0.4098 < p_A < 0.4173$	$0.5816 < p_B < 0.5912$
{5, 6}	$0.4026 < p_A < 0.4093$	$0.5907 < p_B < 0.5974$
{4, 5, 6}	$0.4104 < p_A < 0.4174$	$0.5826 < p_B < 0.5896$

Table-4: A with Strategy III with different special set values; B with Strategy I difference threshold 3

We see from the table, with the special set of roll values we chose, **Strategy III** is not better than **Strategy I.** We only have 4, 5, or 6 as values in this special set of roll values because when we subtract B's score, we want to subtract a large number. However, this was just our assumption that large roll values are more effective if they are in our special set of roll values. It is hard to make certain argument with this assumption. Besides, there are many possibilities with this strategy. For example, what happens if we use a different threshold value different to 3, and what if we choose to include smaller roll values 1, 2 or 3 in our special set of roll values? Thus, **Strategy III** is just an example to illustrate that a strategy sometimes can be too versatile and complex to be useful. So, let us make it simpler.

Strategy IV:

This time, the strategy we implement with player A would be a simpler version of **Strategy III**. This strategy is similar as **Strategy III**. For this strategy, we remove the part where we check if A and B's score difference surpasses the difference threshold 3. So, player A would always try to subtract to B's score unless A gets some special set of values. We have B uses **Strategy I** with difference 3 for comparison.

We can modify the code, code shown below has the special set of roll values as {4, 5, 6}:

```
% A's turn
if player == A
    if limit - score_B <= 6
        score_B = score_B - roll;
    else
        if roll >= 4
            score_A = score_A + roll;
        else
            score_B = score_B - roll;
            if score_B < 0
                score_B = 0;
            end
        end
    end
end
```

We can get a table:

Special Set Values	99.73% Confidence Interval for A	99.73% Confidence Interval for B
{6}	$0.0133 < p_A < 0.0149$	$0.9851 < p_B < 0.9867$
{4, 5}	$0.0159 < p_A < 0.0183$	$0.9817 < p_B < 0.9841$
{4, 6}	$0.0486 < p_A < 0.0520$	$0.9480 < p_B < 0.9514$
{5, 6}	$0.0897 < p_A < 0.0937$	$0.9063 < p_B < 0.9103$
{4, 5, 6}	$0.1977 < p_A < 0.2051$	$0.7949 < p_B < 0.8023$

Table-5: A with Strategy IV with different special set values; B with Strategy I difference threshold 3

We see that this new strategy is the worst one so far, not working very well.

Strategy V:

Lastly, we want to investigate more with the **Convention Strategy**. In the beginning of this section, we set the threshold as 6, and we want to study what happens with the result with another threshold. We call this variation of the **Convention Strategy** as **Strategy V**. For comparison, player A would use **Strategy I&V** with a difference threshold 3, and player B would use **Strategy 0&V.** We will test different threshold values for **Strategy V**. We can get:

Threshold	99.73% Confidence Interval for A	99.73% Confidence Interval for B
1	$0.6448 < p_A < 0.6547$	$0.3453 < p_B < 0.3552$
2	$0.6360 < p_A < 0.6439$	$0.3561 < p_B < 0.3640$
3	$06225. < p_A < 0.6310$	$0.3690 < p_B < 0.3775$
4	$0.6062 < p_A < 0.6177$	$0.3823 < p_B < 0.3938$
5	$0.5913 < p_A < 0.5985$	$0.4015 < p_B < 0.4087$
6	$0.5763 < p_A < 0.5809$	$0.4191 < p_B < 0.4237$
7	$0.5632 < p_A < 0.5728$	$0.4272 < p_B < 0.4368$
8	$0.5559 < p_A < 0.5612$	$0.4388 < p_B < 0.4441$
9	$0.5442 < p_A < 0.5558$	$0.4442 < p_B < 0.4558$
10	$0.5392 < p_A < 0.5466$	$0.4534 < p_B < 0.4608$
11	$0.5311 < p_A < 0.5385$	$0.4615 < p_B < 0.4689$
12	$0.5243 < p_A < 0.5348$	$0.4652 < p_B < 0.4757$

Table-6: A with Strategy I&V, B with Strategy 0&V --- with different threshold values

We can see that with smaller threshold, the more overall benefit by implementing the **Convention Strategy.** It makes sense as for threshold being 1 implies A or B's score is 32 so it is much likely to win the game with the next toss. For a situation where A has score 22 and B has

score 21, and another situation where A has score 10 and B has score 11, it is clear that the latter situation is more severe.

IV. Investigation on Strategy I:

In this section, we will have player A and player B both use **Strategy I** we discussed in previous section, but with the different difference values. Running the simulations, we can get a table:

(Diff A, Diff B)	99.73% Confidence Interval for A	99.73% Confidence Interval for B
(2, 2)	$0.4948 < p_A < 0.5044$	$0.4956 < p_B < 0.5052$
(2, 3)	$0.4311 < p_A < 0.4397$	$0.5603 < p_B < 0.5689$
(2, 4)	$0.4522 < p_A < 0.4614$	$0.5386 < p_B < 0.5478$
(2, 5)	$0.4707 < p_A < 0.4787$	$0.5213 < p_B < 0.5293$
(2, 6)	$0.4871 < p_A < 0.5013$	$0.4987 < p_B < 0.5129$
(2, 7)	$0.5057 < p_A < 0.5143$	$0.4857 < p_B < 0.4943$
(2, 8)	$0.5201 < p_A < 0.5269$	$0.4731 < p_B < 0.4799$
(3, 3)	$0.4959 < p_A < 0.5044$	$0.4956 < p_B < 0.5041$
(3, 4)	$0.4932 < p_A < 0.5017$	$0.4983 < p_B < 0.5068$
(3, 5)	$0.5044 < p_A < 0.5148$	$0.4852 < p_B < 0.4956$
(3, 6)	$0.5194 < p_A < 0.5284$	$0.4716 < p_B < 0.4806$
(3, 7)	$0.5349 < p_A < 0.5428$	$0.4572 < p_B < 0.4651$
(3, 8)	$0.5473 < p_A < 0.5552$	$0.4448 < p_B < 0.4527$
(4, 4)	$0.4953 < p_A < 0.5029$	$0.4971 < p_B < 0.5047$
(4, 5)	$0.5075 < p_A < 0.5148$	$0.4852 < p_B < 0.4925$
(4, 6)	$0.5192 < p_A < 0.5288$	$0.4712 < p_B < 0.4808$
(4, 7)	$0.5320 < p_A < 0.5414$	$0.4586 < p_B < 0.4680$
(4, 8)	$0.5443 < p_A < 0.5507$	$0.4493 < p_B < 0.4557$
(5, 5)	$0.4975 < p_A < 0.5039$	$0.4961 < p_B < 0.5025$
(5, 6)	$0.5061 < p_A < 0.5209$	$0.4791 < p_B < 0.4939$
(5, 7)	$0.5214 < p_A < 0.5304$	$0.4696 < p_B < 0.4786$
(5, 8)	$0.5334 < p_A < 0.5385$	$0.4615 < p_B < 0.4666$
(6, 6)	$0.4956 < p_A < 0.5050$	$0.4950 < p_B < 0.5044$
(6, 7)	$0.5072 < p_A < 0.5178$	$0.4822 < p_B < 0.4928$
(6, 8)	$0.5157 < p_A < 0.5284$	$0.4716 < p_B < 0.4843$
(7, 7)	$0.4931 < p_A < 0.5052$	$0.4948 < p_B < 0.5069$
(7, 8)	$0.5039 < p_A < 0.5154$	$0.4846 < p_B < 0.4961$
(8, 8)	$0.4936 < p_A < 0.5051$	$0.4949 < p_B < 0.5064$

Table-7: A with Strategy I, B with Strategy I --- with different difference threshold values

We only included the case with unique difference threshold pair values, since player A and B both use **Strategy I,** it is the same idea for having (2, 4) and (4, 2) as our difference threshold pair values. Also note that for same difference threshold value pairs (2, 2), (3, 3) … the result we get is very similar which is reasonable considering the two players basically use the same strategy with no difference. The table gives us a comprehensive summary, and it is very clear to observer that in most of the cases, if we use difference threshold 3, it is more likely for us to win the game.

V. Conclusion and Comment:

From the previous discussions we can see that for this particular dice game, a more complex strategy does not always result with a higher win probability. Instead, our first and simplest strategy, it ended up being the most effective one among the ones we have found. This might because the game itself is very simple, and our first strategy implements the core idea to win the game which is to always reduce gap in the score, and when there is not much difference in the gap, we add the score to move on toward the goal. Even though the other three strategies we found did not work well, they help us to see this essential idea of the game. Sometimes, a straightforward approach is the key to success.