# MATH 728D: Machine Learning Lab #18: Expected Values

John Burkardt

February 6, 2019

*When I come to the casino, I expect to make money; so why do they look so happy to see me?*

## 1 Second-Chance

In the game of Second Chance, a drum contains ping pong balls numbered from 1 to 100, which are constantly jumbled. The player pushes a button and a single ball emerges from the drum. The player pushes the button a second time, to release a second ball. Whichever numbered ball is higher, that is the number of dollars that the player wins. The casino charges $75 to play the game. Is the game fair, biased to the casino, or to the player?

To simulate this game, we need to simulate selecting two integers, at random, between 1 and 100. However, we can't allow the same value to be selected twice. This means we can't simply call `r = randi(100);` twice. Instead, we can use the function `r = randperm ( 100, 2 );`, which returns in the vector `r` two distinct randomly selected integers between 1 and 100.

1. Write a program that simulates the game of Second Chance, printing out the two selected values; Run it 5 times;
2. Now run your program 5000 times, storing the score each time; determine the expected value of the game by summing (score minus 75), and dividing by 5000; Report the expected value. Is $75 a fair price for the game?
3. Make a histogram of the 5000 scores (that is, the frequency of each score between 1 and 100). What is the most likely score? What is the average score?

## 2 Roulette

An American roulette wheel has 36 numbered slots, half red and half black, as well as two green slots labeled "0" and "00". Note that 0 and 00 are not considered red or black, and not considered even or odd. A player can wager an amount $B on a single number $D$ between 1 and 36, on red or black, on odd or even, and many other bets. The wheel is spun and the ball randomly lands in a slot $S$. The payoff table is:

| Slot S | Payoff |
|---|---|
| D | player wins 35 * $B |
| 4 numbers | player wins 8 * $B |
| even or odd | player wins $B |
| red or black | player wins $B |
| not what the player chose | player loses $B |

Assume that the roulette ball is equally likely to land in any of the 38 slots. The *expected value* of the game is found by computing the sum of the payoff of every outcome times its probability.

1. Write a program that simulates roulette by returning a random value between 1 and 38, with 37 and 38 being the 0 and 00 slots; run your program 5 times;
2. Play roulette 1000 times, always betting $100 dollars on #7. What is your average winning per game? This is an estimate of the expected value of the bet;
3. Suppose you start with $10,000, and you always bet $100 on "odd" (that is, an odd value between 1 and 35, which does **not** include 37!). Suppose you play until you are bankrupt. How many bets can you expect to make? Answer this question by carrying out this process 1,000 times and averaging;

# 3 Chuck-a-Luck

In the game of Chuck-a-Luck, a player chooses a particular "lucky number" $D$ between 1 and 6. Then three fair dice are rolled, and the player wins or loses depending on how many dice show the value $D$. The payoff table is:

| D's | Payoff |
|---|---|
| 0 | player loses $B |
| 1 | player wins $B |
| 2 | player wins $2*B |
| 3 | player wins $10*B |

Note that you can simulate rolling three dice, and count the number equal to 5 (for instance) by

```
lucky = 5;
r = randi ( 6, 3, 1 );
count = sum ( r == lucky );
```

1. Write a Chuck-a-Luck simulation program that plays the game 5 times in a row, always choosing the same lucky number $D$ and a bet of $100; for each game, report the result of the dice throw, the payoff, and the cumulative winnings;
2. Simulate 1000 games of Chuck-a-Luck, and estimate the expected value of playing a round of the game, given a bet of $100;
3. Suppose you start with $10,000, and you always bet $100 on 5 and you play until you are bankrupt. How many bets can you expect to make? Answer this question by carrying out this process 1,000 times and averaging;

# 4 The Game of Life

In the game of life, you're born, you live a while, and then you die. Here are some simplified statistics for a population of 100,000 people, recording how many of those people survive to age 10, 20, ..., up to age 120 (surprise, none!).

| Age | Population |
|---|---|
| 0 | 100,000 |
| 10 | 99,184 |
| 20 | 98,771 |
| 30 | 97,393 |
| 40 | 95,603 |
| 50 | 92,632 |
| 60 | 85,802 |
| 70 | 73,100 |
| 80 | 50,564 |
| 90 | 17,915 |
| 100 | 932 |
| 110 | 2 |
| 120 | 0 |

We can use the same idea of expected value to estimate the expected lifetime of a person who has just been born into this population.

1. Define `age = 0:10:120` and `pop = [100000, 99184, 98771, ..., 2, 0 ]`;
2. Write a program that simulates life. It creates a person, and then checks every ten years to see if the person is still alive. The probability of survival from age to age $+10$ is $p = \frac{\text{Pop(age+10)}}{\text{Pop(age)}}$. So to decide if the person reached the age of 10, get a random value `r=rand();` and if `r < p` then the person made it at least to age 10, and you can then check to see if they make it to age 20 as well. Keep moving ahead 10 years at a time until the person doesn't make it. Now simulate 5 lives this way, and print out their lifespans
3. If you can get your single life simulation working, try to simulate 1000 lives. Record the length of each life, and take the average. This is an estimate for the expected value of the lifespan of people in this population.