Slot Machine Assessment

Matthew Sullivan
December 22nd, 2022

In this manuscript, we find that the Expected value of a simple slot machine ending its bonus feature is approximately 512 rolls. Furthermore, given specific reward outcomes we find that the expected reward for this slot machine will be $243.4.

The manuscript is partitioned into 5 sections. Section 1 restates the question. Section 2 defines the probabilities used to solve these problems along with stating and proving the results. Section 3 computationally approximated the mathematical results and compares them to the results from a sample slot machine. Section 4 is the python code for the approximated mathematical results. Finally, Section 5 is the python code for the test slot machine.

1 The Question

Imagine a simple three-reel slot machine. Each of the three reels contain a total of 72 symbols and on each spin one of these symbols will be randomly chosen to land. In the mix of symbols is a special bonus 7 symbol with the following frequencies:

Reel 1: 3 7s
Reel 2: 2 7s
Reel 3: 1 7s

Above each of the three reels is a seven-segment bonus meter. Whenever a bonus 7 symbol lands on a reel it contributes to the bonus meter for its reel, increasing it by one. On a spin where all seven segments on all three reels are filled, the player is awarded a bonus of 200 coins and the bonus feature ends. All three meters are reset to zero for the next spin and a new bonus feature begins. If a 7 lands on a reel whose meter is full but the other meters are not completed, the player is awarded a bonus of 2 coins for that reel and the bonus feature continues.

Write a mathematical proof of the average number of spins needed to complete a bonus feature, as well as the average number of coins won. Provide a simulation which verifies your results in a language of your choice. C++ or python is preferred but use anything you're comfortable with

2 Probabilities and Expected Values

Definition 1. Define the probability that reel i rolls a 7 to be $p_i = \frac{4-i}{72}$.

Definition 2. Given a reel that has spun k times, define it to be *k-completing* if the reel has outputted exactly 7 sevens and the last roll was a seven.

Let us list the probabilities required to compute the answer. For a given reel i, we will need:

- $p_{i,k}(x)$ The probability that the reel spun k times has exactly x sevens.
- $c_{i,k}$ The probability that the reel spun k times is k-completing.
- $d_{i,k}$ The probability that the reel spun k times has outputted at least 7 sevens and the reel is not k-completing.

Claim 1.

$$p_{i,k}(x) = \binom{k}{x} \cdot p_i^x \cdot (1-p_i)^{k-x}.$$

Proof. We must find the number of possible combinations of x sevens and $k-x$ non-sevens. There are $\binom{k}{x}$ possible combinations to place the sevens and the remaining position are filled with non-sevens.

We are given that there are x sevens with probability p_i^x and $k-x$ non-sevens with probability $(1-p_i)^{k-x}$. Therefore,

$$p_{i,k}(x) = \binom{k}{x} \cdot p_i^x \cdot (1-p_i)^{k-x}.$$

□

Claim 2.

$$c_{i,k} = \binom{k-1}{6} \cdot p_i^7 \cdot (1-p_i)^{k-7}.$$

Proof. Since the reel is k-completing, we must find the number of possible combinations of 7 sevens and $k-7$ non-sevens. Since a seven appears in the last position, there are $\binom{k-1}{6}$ possible combinations to place the sevens and the remaining position are filled with non-sevens.

We are given that there are 7 sevens with probability p_i^7 and $k-7$ non-sevens with probability $(1-p_i)^{k-7}$. Therefore,

$$c_{i,k} = \binom{k-1}{6} \cdot p_i^7 \cdot (1-p_i)^{k-7}.$$

□

Claim 3.

$$d_{i,k} = 1 - c_{i,k} - \sum_{j=0}^{6} p_{i,k}(j)$$

Proof. We choose to find the probability of the compliment of this event. In particular, the probability $\bar{d}_{i,k}$ when the reel is k-completing or has less than 7 sevens. In doing so, we will find that $d_{i,k} = 1 - \bar{d}_{i,k}$.

A reel being k-completing and a reel having less that 7 sevens appear are mutually exclusive events, therefore, by Claims 1 and 2 $\bar{d}_{i,k} = c_{i,k} + \sum_{j=0}^{6} p_{i,k}(j)$. It follows that

$$d_{i,k} = 1 - c_{i,k} - \sum_{j=0}^{6} p_{i,k}(j).$$

□

Definition 3. Let E_e be the expected number rolls of a bonus feature.

Theorem 1.

$$E_e = \sum_{k=7}^{\infty} k \cdot \left(c_{1,k} \cdot c_{2,k} \cdot c_{3,k} + \sum_{\{h,i,j\}=\{1,2,3\}} c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k} \right).$$

Proof. E_e is the expected number rolls of a bonus feature. This is equal the value of an event times the probability such an event occurs over all possible events or alternatively, $E_e = \sum_{x \in X} x \cdot pr(x)$.

We choose to partition our events into the number of rolls k it takes for the bonus feature to end. The value of such an event will be k.

There are 7 different events that are the disjoint union of this event occurring depending on which of the three reels are k-completing and which are not k-completing but contain at least 7 sevens.

Therefore, the probability that we end on a k roll will be the sum of these 7 probabilities. Thus, the probability of a bonus feature ending on a k-roll is

$$c_{1,k} \cdot c_{2,k} \cdot c_{3,k} + \sum_{\{h,i,j\}=\{1,2,3\}} c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k}.$$

Since a bonus feature can not end when $k < 7$, our summation starts at $k = 7$ and we have our desired result. $\square$

Definition 4. Let E_s be the expected number of bonus sevens that appear for the slot machine. Let $E_{i,k}$ be the expected number of bonus sevens that appear on reel i, given k spins, the reel has at least 7 sevens, and the reel is not k-completing.

Claim 4.

$$E_{i,k} = \frac{1}{d_{i,k}} \sum_{x=8}^{k} (x-7) \cdot p_{i,k}(x).$$

Proof. $E_{i,k}$ is the conditional expected value based on the conditional probability that after k spins, reel i had at least 7 sevens appear, and the reel is not k-completing.

$$E_{i,k} = \sum_{x=0}^{k} (x-7) \cdot p(x),$$

where $p(x)$ is the probability that after k spins, reel i had at least 7 sevens appear, the reel is not k-completing, and the reel had x sevens appear.

Clearly $p(x) = 0$ for $x < 7$ as we can not have at least 7 sevens appear and x sevens appear simultaneously for $x < 7$. For $x = 7$, the value of a roll is 0, therefore, our sum start at $x = 8$.

Let A_x be the event that reel i has exactly x sevens appear. By definition, $p(A_x) = p_{i,k}(x)$.

Let B be the event that, after k rolls, reel i had at least 7 sevens appear and the reel is not k-completing. By definition, $P(B) = d_{i,k}$.

By definition of conditional probabilities,

$$p(x) = p(A_x|B) = \frac{p(A_x \cap B)}{p(B)}.$$

Since $x > 7$,

$$p(x) = \frac{p(A_x \cap B)}{p(B)} = \frac{p(A_x)}{p(B)}.$$

Simplifying, we get

$$\begin{aligned}
E_{i,k} &= \textstyle\sum_{x=0}^{k}(x-7)\cdot p(x) \\
&= \textstyle\sum_{x=0}^{k}(x-7)\cdot \frac{p(A_x)}{p(B)} \\
&= \textstyle\sum_{x=0}^{k}(x-7)\cdot \frac{p_{i,k}(x)}{d_{i,k}} \\
&= \textstyle\frac{1}{d_{i,k}}\sum_{x=8}^{k}(x-7)\cdot p_{i,k}(x)
\end{aligned}$$

□

Theorem 2.

$$E_s = \sum_{k=8}^{\infty}\sum_{\{h,i,j\}=\{1,2,3\}} (E_{h,k} + E_{i,k}) \cdot c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + E_{h,x} \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k}.$$

Proof. E_s is the expected number of bonus sevens. This is equal the value of an event times the probability such an event occurs over all possible events or alternatively, $E_s = \sum_{x \in X} x \cdot pr(x)$.

We choose to partition our events into the number of rolls k it takes for the bonus feature to end and into which reels are k-completing and which reels are not. The value of such an event will be the sum of the expected values of the reels that are not k-completing.

Since at least one reel is k-completing, it follows that

$$E_s = \sum_{k=8}^{\infty}\sum_{\{h,i,j\}=\{1,2,3\}} (E_{h,k} + E_{i,k}) \cdot c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + E_{h,x} \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k}.$$

□

3 Concluding Remarks

The expected value for the number of rolls of a bonus feature is

$$E_e = \sum_{k=7}^{\infty} k \cdot \left(c_{1,k} \cdot c_{2,k} \cdot c_{3,k} + \sum_{\{h,i,j\}=\{1,2,3\}} c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k} \right),$$

outlined in Theorem 1. This is an infinite summation and thus the summation must be approximated. In the code found in Appendix A, the summation is completed from $k = 7$ to $k = 10000$ and outputs an approximate value of $E_e \approx 512$. After finding the average number of rolls over 100000 test samples of a slot machine (found in Appendix B), E_e was sampled to be $E_e \approx 512.9$, a match to our result.

The expected value for the number of extra sevens of a bonus feature is

$$E_s = \sum_{k=8}^{\infty} \sum_{\{h,i,j\}=\{1,2,3\}} (E_{h,k} + E_{i,k}) \cdot c_{j,k} \cdot d_{h,k} \cdot d_{i,k} + E_{h,x} \cdot c_{i,k} \cdot c_{j,k} \cdot d_{h,k},$$

outlined in Theorem 2. This is an infinite summation and thus the summation must be approximated (Or Alternatively, we could use Real Analysis to approximate the summations as they are similar to the geometric series). In the code found in Appendix A, the summation is completed from $k = 7$ to $k = 10000$ and outputs an approximate value of $E_e \approx 21.7$. After finding the average number of rolls over 100000 test samples of a slot machine (found in Appendix B), E_e was sampled to be $E_e \approx 21.7$. Given the approximate value of 21.7 bonus sevens from out calculations, the expected earnings of the slot machine will be $\$200 + 2 \cdot \$21.7 = \$243.4$. Based on the random sampling of slot machine in python, the expected earning of the slot machine will be $\$200 + 2 \cdot \$21.7 = \$243.4$, a match to our findings.

It is important to note that the implemented code for E_s has some reductions over the formula as products like $E_(i,k) \cdot d_{i,k}$ have cancelling terms. Furthermore, the function "kExpectedBonusSevens" in Appendix A takes advantage of the fact that the iterative terms in $E_{i,k}$ are related by a factor dependent on known values, thus reducing its time complexity.

Furthermore, the code in Appendix A generalizes the problem to slot machines with variable probabilities for reels and variable bonus feature meter length. This can be extended to my proofs, however, I thought it best to answer the specific question given.

4 Appendix A: Expected Value Approximations in Python

```
# factorial(c) computes c! = 1*2*3*...*c.

    def factorial(c):
    answer = 1
    while c>1:
        answer = c * answer
        c -= 1
    return answer


# choice(c, r) computes the binomial coefficient c choose r.

    def choice(c,r):
    answer = 1
    temp = c-r
    while c > temp:
        answer = answer * c
        c -= 1
    return answer/factorial(r)

# kExactly(k, MeterLength, P1) computes the probability that a bonus
# meter of size "MeterLength" is filled on the kth roll on a single reel with
# the success symbol having probability P1.

    def kExactly(k, MeterLength, P1):
    return choice(k-1,MeterLength-1)*P1**MeterLength*(1-P1)**(k-MeterLength)

# kAtleast(k, MeterLength, P1) computes the probability that at the kth
# roll the bonus meter of length "MeterLength" is filled on a single reel with
# the success symbol having probability P1.
```

```
def kAtleast(k, MeterLength, P1):
    return 1 - (choice(k-1, MeterLength-1)*P1**MeterLength*(1-P1)**(k-MeterLength)+\
sum(choice(k, i)*P1**i*(1-P1)**(k-i) for i in range(0, MeterLength)))

# kExpectedBonusSevens(k,Meterlength,P1) computes the expected number
# of bonus sevens at the kth roll of a single reel given the bonus meter of
# length "MeterLength" is filled and the success symbol has probability P1.


def kExpectedBonusSevens(k, MeterLength, P1):
    i = MeterLength + 1
    summation = (i-MeterLength)*choice(k, i)*P1**i*(1-P1)**(k-i)
    temp = summation
    while i < k:
        temp = (temp*(i-MeterLength+1)*(k-i)*P1)/((i-7)*(i+1)*(1-P1))
        summation += temp
        i +=1
    return summation

# Choose your probability of success symbols here.

P1 = 1/24
P2 = 1/36
P3 = 1/72

# Choose the length of the bonus meter here as "MeterLength".

    MeterLength = 7

# Count is the number of times the reels are spun. Note count increases by
1 at the start of the while loop to start # at "MeterLength".

    = MeterLength - 1

# ExpectedEnd will be an approximation of the expected number of reel
rolls to end the bonus feature.

ExpectedEnd = 0
```

```
# ExpectedBonusSevens will be an approximation of the expected number
of bonus sevens at the end of a bonus feature.

ExpectedBonusSevens = 0

# This loop computes an approximate (bounding the infinite summation by
# 10000) expected value of when the bonus feature ends and the expected
# number of sevens. The outputs are:

# “count” the bound for the infinite summation of the expected value;
# “ExpectedEnd” the expected number of rolls required to end the bonus
# feature; and
# “ExpectedBonusSevens” the expected number of bonus sevens rolled after
# a bonus feature has ended.

while count < 10000:
    count += 1
    A = kExactly(count, MeterLength, P1)
    B = kExactly(count, MeterLength, P2)
    C = kExactly(count, MeterLength, P3)
    Aat = kAtleast(count, MeterLength, P1)
    Bat = kAtleast(count, MeterLength, P2)
    Cat = kAtleast(count, MeterLength, P3)
    ExpectedEnd += count * (A * Bat * Cat +\
    B * Aat * Cat +\
    C * Aat * Bat +\
    A * B * Cat +\
    A * C * Bat +\
    B * C * Aat +\
    A * B * C)
    ExpectedBonusSevens +=
    A * Cat * kExpectedBonusSevens(count, MeterLength, P2) +\
    A * Bat * kExpectedBonusSevens(count, MeterLength, P3) +\
    B * Cat * kExpectedBonusSevens(count, MeterLength, P1) +\
    B * Aat * kExpectedBonusSevens(count, MeterLength, P3) +\
    C * Bat * kExpectedBonusSevens(count, MeterLength, P1) +\
    C * Aat * kExpectedBonusSevens(count, MeterLength, P2) +\
    A * B * kExpectedBonusSevens(count, MeterLength, P3) +\
```

```
    A * C * kExpectedBonusSevens(count, MeterLength, P2) +\
    B * C * kExpectedBonusSevens(count, MeterLength, P1)

print(count, ExpectedEnd, ExpectedBonusSevens)
```

5 Appendix B: Test Slot Machine in Python

```
import random

# "tests" is the number of tests performed.

tests = 100000

# "i" is the counter for test i.

i = 0

# "ExpectedRolls" will count number of rolls performed over all tests.

ExpectedRolls = 0

# "ExtraSevens" counts the number of extra sevens (above 7) over all reels
and tests.

ExtraSevens = 0

# The outside loop covers all tests. For each test, the inside loop simulates
# a slot machine using the random.randint function to generate random
# numbers, counters C1, C2, C3 to count the number of sevens in each reel
# and counter "count" to count the number of rolls in a single simulation.
# ExpectedRolls is the sum of the final "counter" of each simulation and
# ExtraSevens is the sum of C1, C2, C3 over each simulation.

while i < tests:
    i += 1
    C1 = 0
    C2 = 0
```

```
    C3 = 0
    count = 0
    while C1 < 7 or C2 < 7 or C3 < 7:
        count += 1
        if random.randint(1,24) == 1:
            C1 += 1
        if random.randint(1,32) == 1:
            C2 += 1
        if random.randint(1,72) == 1:
            C3 += 1
    ExpectedRolls += count
    ExtraSevens += (C1 + C2 + C3 - 21)
```

The expected number of rolls will be the "ExpectedRolls" divided by "tests" and the expected number of # extra sevens will be "ExtraSevens" divided by "tests".

```
print(ExpectedRolls/tests,ExtraSevens/tests)
```