MATH 310 PROJECT 1

KATE SANDERS

The purpose of this project was to explore a coin-flipping scenario in which getting two consecutive coins to land on heads is a success. Let the random variable X equal the number of flips of a coin that are required to observe heads on two consecutive flips. The sample space of X contains all countable integers above and including two, since it requires at least two trials to obtain consecutive heads.

In this paper, I present the equation, visualization, and select values of both the probability mass function (pmf) and cumulative distribution function (cdf) of X. Additionally, I find the moment-generating function and use its derivatives to find the mean, variance and standard deviation of X. To check my findings, I created a program in Python that, using the pmf equation, records the pmf and cdf values for all $x \leq 101$ and calculates $E(X)$ and $E(X^2)$. The program also runs a coin-flip simulation to test the results experimentally.

1. The Probability-Mass-Function

To find patterns that could lead to a pmf, I created Table 1. This table enumerates all possible combinations of coin flips that lead to success for $2 \leq x \leq 8$ trials. Table 1 also records the number of successful combinations for each x, as well as the number of combinations beginning with Heads, H_x, and Tails, T_x. To get from trial x to trial $(x+1)$, T and/or H can be appended to the front of the current combinations depending on the current first flip. Only T can be appended before H, but both T and H can be appended before T. Using the number of combinations starting with Heads and Tails at trial x, it is possible to determine the total number of combinations possible at trial $x+1$, which can be written as $(T+H)_{x+1}$:

$$(T+H)_{x+1} = H_x + 2(T_x) \quad .$$

This causes H_x, T_x, and the total number of combinations to follow the Fibonacci Sequence (F). The beginning of this pattern can be seen in Table 1. Let F_{x-1} represent the number of coin flip combinations that lead to success with x coin flips. Any Fibonacci number can be calculated from the following equation:

$$F_x = \left(\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^x - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^x \right) \quad .$$

Each coin flip is independent, which means that the probability of each of the x coin flips can be multiplied together to get the probability of that particular set of x coin flips occurring. Also, assuming a fair coin, there is a $\frac{1}{2}$ chance that the coin lands on Heads and a $\frac{1}{2}$ chance that the coin lands on Tails. Knowing this, the the probability for any given combination of x coin flips can be found using p^x, where $p = \frac{1}{2}$. Combining the probability of x coin flips with the previously determined number of combinations for x trials, F_{x-1}, results in the pmf of X:

$$P(X = x) = (F_{x-1})p^x \quad .$$

The cdf is simply the sum of each pmf from 0 to x:

$$P(X \leq x) = \sum_{i=0}^{x} (F_{i-1})p^i \quad .$$

Table 1. Exploring Patterns in the Number of Combinations for the First 7 Trials.

Trial (x)	*Enumeration of Combinations*	*Number of Combinations* (F_{x-1})	*Combinations Starting with Heads* (H_x)	*Combinations Starting with Tails* (T_x)
2	HH	1	1	0
3	THH	1	0	1
4	$TTHH$ $HTHH$	2	1	1
5	$TTTHH$ $HTTHH$ $THTHH$	3	1	2
6	$TTTTHH$ $HTTTHH$ $THTTHH$ $HTHTHH$ $TTHTHH$	5	2	3
7	$TTTTTHH$ $HTHTTHH$ $TTHTTHH$ $HTTTTHH$ $THTTTHH$ $HTTHTHH$ $THTHTHH$ $TTTHTHH$	8	3	5
8	$TTTTTTHH$ $HTHTHTHH$ $TTTTHTHH$ $HTHTTTHH$ $TTHTHTHH$ $HTTHTTHH$ $THTTTTHH$ $HTTTTTHH$ $THTTHTHH$ $HTTTHTHH$ $THTHTTHH$ $TTHTTTHH$ $TTTHTTHH$	13	5	8

Select values from the pdf and cdf can be viewed in Table 2. A probability histogram for the pmf is shown in Figure 1. According to the figure, the probability of getting two consecutive heads at exactly x trials decreases as the number of trials increases. A graph of the cdf is shown in Figure 2. This figure shows that the the probability of flipping two consecutive heads by at least the x^{th} trial approaches 1 as x approaches infinity, supporting the formula for the pmf.

TABLE 2. Select Theoretical Values from the pmf and cdf of X.

x	$P(X = x)$	$P(X \leq x)$
2	$\frac{1}{4}$	$\frac{1}{4}$
3	$\frac{1}{8}$	$\frac{3}{8}$
4	$\frac{1}{8}$	$\frac{1}{2}$
5	$\frac{3}{32}$	$\frac{19}{32}$
6	$\frac{5}{64}$	$\frac{43}{64}$
7	$\frac{1}{16}$	$\frac{47}{64}$
8	$\frac{13}{256}$	$\frac{201}{256}$
...	...	...
40	$5.75E-05$	0.999756333

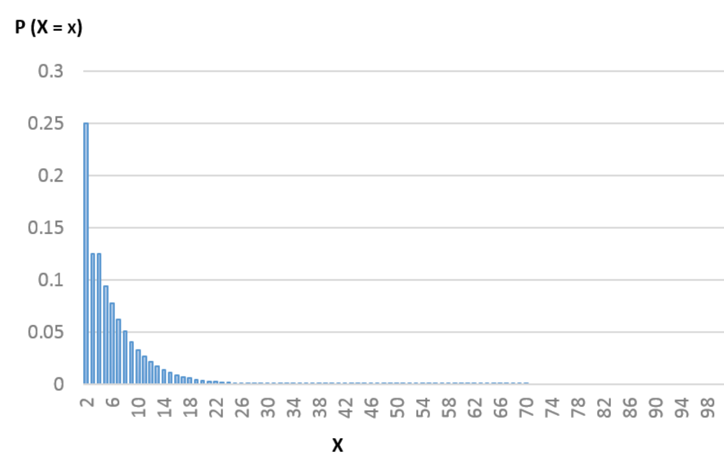

FIGURE 1. A Probability Histogram of Theoretical Values from the Probability Mass Function of X

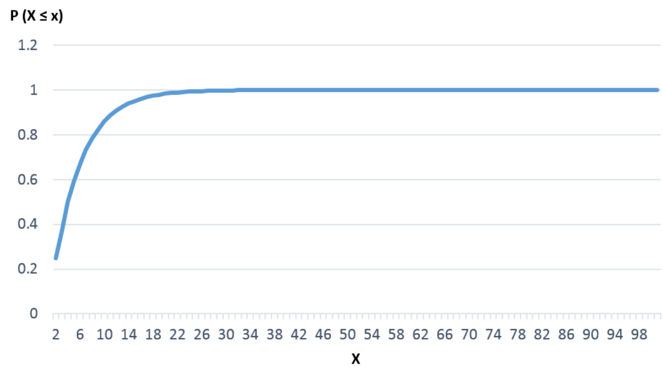

FIGURE 2. A Line Graph of Theoretical Values from the Cumulative Distribution Function of X

2. THE MOMENT GENERATING FUNCTION

The moment generating function for X can be expressed as:

Let $\varphi = \left(\frac{1+\sqrt{5}}{2}\right)$ and $\psi = \left(\frac{1-\sqrt{5}}{2}\right)$

Identities:

$\varphi + \psi = 1$

$\varphi - \psi = \sqrt{5}$

$\varphi\psi = -1$

$$\begin{aligned}
M(t) &= \sum_{x=2}^{\infty} e^{tx} p^x F_{x-1} \\
&= \sum_{x=2}^{\infty} e^{tx} p^x \left(\frac{1}{\sqrt{5}}\varphi^{x-1} - \frac{1}{\sqrt{5}}\psi^{x-1} \right) \\
&= \frac{1}{\sqrt{5}} \sum_{x=2}^{\infty} (e^t p)^x \varphi^{x-1} - \psi^{x-1} \\
&= \frac{1}{\sqrt{5}} \left[\sum_{x=2}^{\infty} (e^t p)^x \varphi^{x-1} - \sum_{x=2}^{\infty} (e^t p)^x \psi^{x-1} \right] \\
&= \frac{1}{\sqrt{5}} \left[\frac{1}{\varphi} \sum_{x=2}^{\infty} (e^t p\varphi)^x - \frac{1}{\psi} \sum_{x=2}^{\infty} (e^t p\psi)^x \right] \\
&= \frac{1}{\sqrt{5}} \left[\frac{1}{\varphi} \left(\frac{(e^t p\varphi)^2}{1 - e^t p\varphi} \right) - \frac{1}{\psi} \left(\frac{(e^t p\psi)^2}{1 - e^t p\psi} \right) \right] \\
&= \frac{(e^t p)^2}{\sqrt{5}} \left[\frac{\varphi}{1 - e^t p\varphi} - \frac{\psi}{1 - e^t p\psi} \right] \\
&= \frac{(e^t p)^2}{\sqrt{5}} \left[\left(\frac{\varphi}{1 - e^t p\varphi} \right) \left(\frac{1 - e^t p\psi}{1 - e^t p\psi} \right) - \left(\frac{\psi}{1 - e^t p\psi} \right) \left(\frac{1 - e^t p\varphi}{1 - e^t p\varphi} \right) \right] \\
&= \frac{(e^t p)^2}{\sqrt{5}} \left[\frac{(\varphi - e^t p\varphi\psi) - (\psi - e^t p\varphi\psi)}{(1 - e^t p\psi)(1 - e^t p\varphi)} \right] \\
&= \frac{(e^t p)^2}{\sqrt{5}} \left[\frac{\varphi - \psi}{1 - e^t p(\varphi + \psi) + (e^t p)^2 \varphi\psi} \right] \\
&= \frac{(e^t p)^2}{\sqrt{5}} \left[\frac{\sqrt{5}}{1 - e^t p - (e^t p)^2} \right] \\
&= \frac{(e^t p)^2}{1 - e^t p - (e^t p)^2}
\end{aligned}$$

The moment-generating function can by checked by plugging in $t = 0$ and $p = \frac{1}{2}$, since we are assuming a fair coin. With these values,

$$M(0) = \frac{p^2}{1 - p - p^2} = \frac{(\frac{1}{2})^2}{1 - \frac{1}{2} - (\frac{1}{2})^2} = 1 \quad ,$$

which is the correct value for $M(0)$. To find $M'(t)$ and $M''(t)$ I used an online derivative calculator[1].

$$\begin{aligned} M'(t) &= \frac{2p^2 e^{2x}}{-p^2 e^{2x} - pe^x + 1} - \frac{p^2 e^{2x}\left(-2p^2 e^{2x} - pe^x\right)}{\left(-p^2 e^{2x} - pe^x + 1\right)^2} \\ &= -\frac{p^2\left(pe^x - 2\right) e^{2x}}{\left(p^2 e^{2x} + pe^x - 1\right)^2} \end{aligned}$$

$$\begin{aligned} M''(t) = &-\frac{p^3 e^{3x}}{\left(p^2 e^{2x} + pe^x - 1\right)^2} - \frac{2p^2\left(pe^x - 2\right) e^{2x}}{\left(p^2 e^{2x} + pe^x - 1\right)^2} \\ &+ \frac{2p^2\left(pe^x - 2\right) e^{2x}\left(2p^2 e^{2x} + pe^x\right)}{\left(p^2 e^{2x} + pe^x - 1\right)^3} \\ = &\frac{p^2 e^{2x}\left(p^3 e^{3x} - 5p^2 e^{2x} + 3pe^x - 4\right)}{\left(p^2 e^{2x} + pe^x - 1\right)^3} \end{aligned}$$

3. Calculating Mean and Varience

The mean of X can be found by setting $t = 0$ for $M'(t)$, which was derived in Section 2. Therefore, the mean is

$$\mu = E(X) = M'(0) = -\frac{p^2\left(p - 2\right)}{\left(p^2 + p - 1\right)^2} = -\frac{(\frac{1}{2})^2\left(\frac{1}{2} - 2\right)}{\left((\frac{1}{2})^2 + \frac{1}{2} - 1\right)^2} = 6 \ .$$

Similarly, the variance of X can be calculated as follows:

$$\begin{aligned} \sigma^2 &= M''(0) - (M'(0))^2 \\ &= \frac{p^2(p^3 - 5p^2 + 3p - 4)}{(p^2 + p - 1)^3} - \left(-\frac{p^2\left(p - 2\right)}{\left(p^2 + p - 1\right)^2}\right)^2 \\ &= \frac{(\frac{1}{2})^2((\frac{1}{2})^3 - 5(\frac{1}{2})^2 + 3p - 4)}{((\frac{1}{2})^2 + \frac{1}{2} - 1)^3} - \left(-\frac{(\frac{1}{2})^2\left(\frac{1}{2} - 2\right)}{\left((\frac{1}{2})^2 + \frac{1}{2} - 1\right)^2}\right)^2 \\ &= 58 - 36 \\ &= 22 \ . \end{aligned}$$

The standard deviation of X is simply:

$$\sigma = \sqrt{22} \approx 4.69 \ .$$

[1] https://www.derivative-calculator.net/

4. Computer Calculations

To test the pmf function and moment-generating function derived in this project, I created a computer program in Python. The program can be viewed in the Appendix. This program calculates both $E(X)$ and $E(X^2)$ using the pmf formula to find the mean and variance. It also keeps track of the pmf and cdf value for every x. The values of the pmf and cdf for each trial were recorded and saved in a text file that could be converted into an Excel spreadsheet. This spreadsheet was then used to generate Figures 1 and 2.

The Python program also ran an simulation of the coin-flip problem. This simulation, which ran 100,000 times, kept track of the number of coin flips needed to obtain two consecutive heads. The experimental frequency of each x was divided by the total number of recorded flips to get the experimental probability of x. In this way, the experimental pmf and cdf were recorded in a text file alongside the theoretical pmf and cdf. Using Excel, I created a probability histogram comparing the expected and experimental probability for $x \leq 25$, which can be seen in Figure 3. Select probability values can be more closely compared in Table 3. The experi-

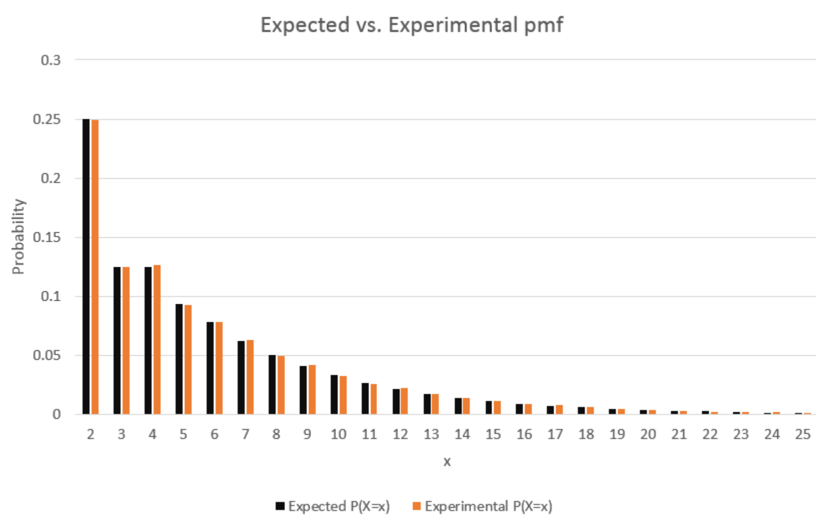

Figure 3. A Probability Histogram Comparing the Theoretical Values from the Probability Mass Function of X to the Experimental Probability of Each x

mental probabilities closely match the theoretical probabilities, further supporting the pmf equation derived in Section 1. For further comparison, refer to Figure 4. This shows the theoretical and experimental cumulative distributions for $x \leq 25$, which are almost exactly the same.

Additionally, the program prints the mean, variance, and standard deviation from the simulation alongside the theoretical mean, variance, and standard deviation. This allows for comparison of these metrics. The values of the theoretical and an experimental mean, varience, and standard deviation can be viewed in Table

4. These experimental measurements are very close to the theoretical measurements found in Section 3, verifying the pmf and moment-generating function of X calculated in Sections 1 and 2.

TABLE 3. Select Theoretical and Experimental Probabilities.

x	*Theoretical Probability*	*Experimental Probability*
2	0.25	0.24927
3	0.125	0.12467
4	0.125	0.12649
5	0.09375	0.09266
10	0.033203125	0.03295
15	0.011505127	0.01161
20	0.003987312	0.00366
25	0.001381874	0.00139

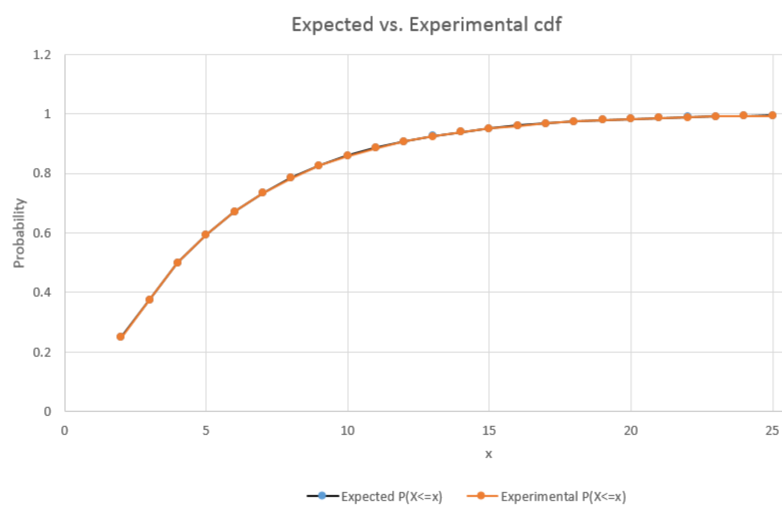

FIGURE 4. A Line Graph Comparing the Theoretical Values from the Cumulative Distribution Function of X to the Experimental Cumulative Probabilities.

TABLE 4. Comparison of Theoretical and Experimental Measurements of X

	Theoretical	*Experimental*
Mean	6	6.0142
Variance	22	22.29802
Standard Deviation	4.69	4.722078

APPENDIX

```
import statistics, random, math, collections

# enter the destired file location for "file.txt"
file = open('file.txt', 'w')

def coin_flips():
    flips = 0
    prev = 0
    finished = False
    # flips a coin until there are consecutive heads
    while not finished:
        flips += 1
        coin = random.randint(0, 1)
        if coin == 1 and prev == 1:
            # return the number of flips needed for consecutive heads
            return flips
        prev = coin

def experiment():
    repeats = 100000
    values = []
    # runs coin_flips() a bunch of times and saves each result
    while repeats > 0:
        repeats-=1
        if coin_flips() <= 101:
            values.append(coin_flips())
    return values

def expected():
    # dictionary to record the values of the pmf and cdf
    values = {}
    cdf = 0
    mean = 0
    factorial_moment = 0
    var = 0
    # there is a 1/2 chance of the coin landing on heads
    p = 0.5
    fib1 = 1
```

```
    fib2 = 0
    # only 100 iterations are needed to see the cdf approach 1
    for i in range(100):
        # sum starts at 2
        spot = i+2
        # F_0 = 1
        if i != 0:
            fibtemp = fib2
            fib2 = fib1
            fib1 = fib1+fibtemp
        # calculation for P(X = x)
        pmf = fib1 * (p**spot)
        # calculation for P(X <= x)
        cdf+=pmf
        # add pmf and cdf to dictionary:
        values[spot] = (pmf, cdf)
        # E(X) calculation
        mean += (pmf * spot)
        # E(X**2) calculation
        factorial_moment += (pmf * (spot)**2)
    # calculate Variance = E(X**2) - E(X)**2
    var = factorial_moment - (mean**2)
    # Round the results:
    results = [math.ceil(mean), math.ceil(var), values]
    return results

def get_experimental_pdf_cdf_dictionary(experiment_results):
    res = collections.Counter(experiment_results)
    values = {}
    total = sum(res.values())
    cdf = 0
    for k in range(2,101):
        pmf = res[k]/total
        cdf += res[k]/total
        values[k] = (pmf, cdf)
    return values


def main():
    #create file headers
    file.write("x\texpected pmf\texperimental pmf\texpected
        cdf\texperimental cdf\n")
    experiment_results = experiment()
    experiment_pdf_cdf =
        get_experimental_pdf_cdf_dictionary(experiment_results)
    experiment_mean = statistics.mean(experiment_results)
    experiment_stdev = statistics.stdev(experiment_results)
    experiment_var = statistics.variance(experiment_results)
    expected_results = expected()
    expected_pdf_cdf = expected_results[2]
    for k in range(2,101):
        file.write(str(k) + "\t" + str(expected_pdf_cdf[k][0])+ "\t"
            +str(experiment_pdf_cdf[k][0])+ "\t" +
```

```
              str(expected_pdf_cdf[k][1])+ "\t"
                  +str(experiment_pdf_cdf[k][1])+ "\n" )
   file.close()
   print("Expected Mean: ", expected_results[0],
        "\tExperimental Mean: ", experiment_mean)
   print("Expected Variance: ", expected_results[1],
        "\tExperimental Variance: ", experiment_var)
   print("Expected Standard Deviation: ",
       round(math.sqrt(expected_results[1]), 2),
        "\tExperimental Standard Deviation: ", experiment_stdev)


main()
```