Post 2 Probability, Coin Flips, and Stirling’s Approximation

Patrick Chao

```r
library(ggplot2)
library(ggthemr)
ggthemr("light")
set.seed(133)
```

Introduction

One of my favorite areas of mathematics is probability. Probability is both very interesting and challenging, as small changes in problem statements can result in drastically different questions. A classic example of probability questions involve coin tosses.

Consider the following scenariso. What is the probability of flipping two heads in a row?
What is the average number of coin flips until two heads in a row?

These problems seem very similar, although the second question is much more challenging than the first. The first problem follows readily from elementary probability. The chance of flipping two heads in a row is 14, because there is a one half chance for each flip.

However, by adding the words “average number until”, this is talking about expected value, which can be much trickier to deal with. If you are curious, feel free to spend a few minutes trying to work out the question. If you are lazy, try to at least come up with a guess for two flips in a row. How about three flips in a row?

What is unique about probability is the scenarios are often very understandable and simple, but the solution may be very complex. However, due to the simplicity, we can simply model the model space using for loops.

Computer Simulation Modeling

The great thing about computers is that we may ask the computer to flip thousands or millions of coins in less than a second, and ask the computer to compute the average. In this post, I will talk about creating functions with flexible arguments to accomodate all forms of problems.

In the program below, I have two arguments, *numHeadsInARow* and *numTrials*. These are pretty self-explanatory, in this case we have *numHeadsInARow=2*, and I set the number of trials to be 10000 by default. This short script contains a for loop which iterates over each trial. I count the number of flips necessary to reach two heads in a row in each trial in the variable *numFlips*. I also have a variable *numConsecutiveHeads*, which keeps track of the number of consecutive heads. This increments by one when there is a head, and resets to zero on tails. The middle of the program contains a while loop, which continues until the number of consecutive heads is met. By nature of the problem, we could get exceedingly unlucky and have trials of dozens of flips until two heads, but this is quite unlikely (we will see how unlikely later on!). After keeping track of the number of flips in each iteration, I store this inan array *flips*. I return the average of the values in *flips*.

```r
headsInARow <- function(numHeadsInARow,numTrials=10000,returnMean=TRUE){
  flips <- rep(0,numTrials)
  for(i in 1:numTrials){
    #keep track number of consecutive heads
    numConsecutiveHeads <- 0
    #keep track of total number of flips, starting at 0
    numFlips <- 0

    #flip until you get the number of heads in a row
    while(numConsecutiveHeads<numHeadsInARow){

      #Simulate a coin flip
      flip <- runif(1)
      #Increment the total number of flips in this iteration
      numFlips <- numFlips+1
      #if greater than 0.5, heads
      if (flip>0.5){
        #Add one to the counter of number of consecutive heads
        numConsecutiveHeads <- numConsecutiveHeads+1
      }else{
        #If tails, reset the counter of number of heads
        numConsecutiveHeads <- 0
      }
    }

    #Accumulate total number of flips
    flips[i] <- numFlips
  }
  #Return the average
  if(returnMean){
    mean(flips)
  }else{
    flips
  }
}
```

```r
headsInARow(numHeadsInARow = 2)
```

```
## [1] 5.9351
```

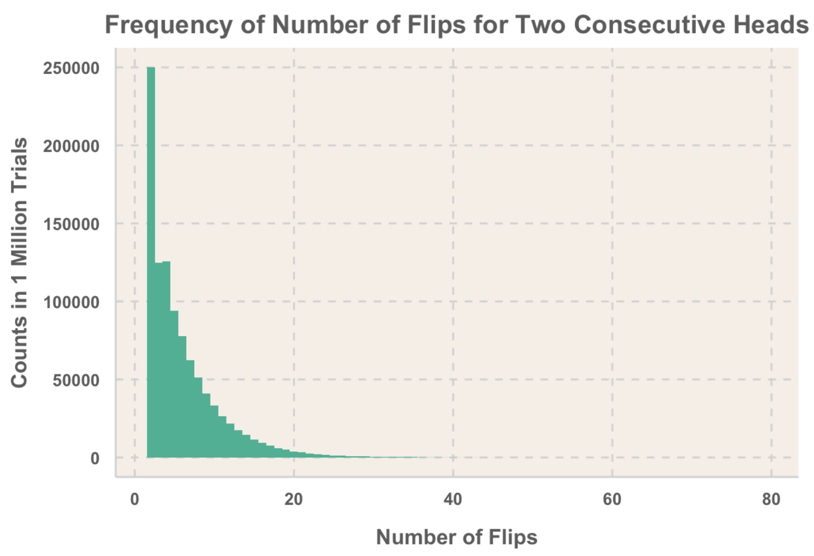
Wow, the answer seems to be about 6! That quite surprising, and a bit counterintuitive.

Probability Density

But we aren't done yet with investigating the two heads case. An even more challenging question is investigating the probability density distribution of the number of flips needed for two heads. Through 10000 test cases, we are pretty confident the average is around 6, but what does the distribution look like?

By creating an array that contains all of the individual trials, we may just return the array and plot it! Again utilizing the computer's memory capabilities we are able to tackle this tougher problem.

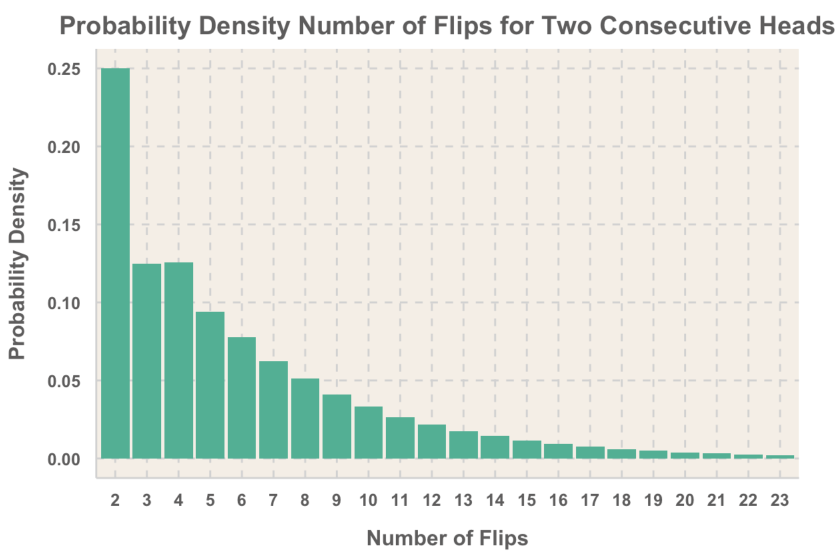
```
flips <- headsInARow(numHeadsInARow = 2,numTrials=1000000,returnMean = FALSE)
ggplot()+aes(x=flips)+geom_histogram(binwidth = 1)+labs(x="Number of Flips",y="Counts in 1 Million Trials",title="
Frequency of Number of Flips for Two Consecutive Heads")
```




```
freq <- table(flips)
proportions <- prop.table(freq)
```

To make this a bit nicer, we can instead look at the plot for up two 23 flips as this encompasses more than 99% of the cases. Additionally, a more general method is not considering frequency, but rather proportion. This instead represents the probability density distribution that we desired. We can find the proportions using the *table* and *prop.table* methods.

```
freq <- table(flips)
proportions <- prop.table(freq)
propsTruncated <- as.data.frame(proportions[1:22])
ggplot(data=propsTruncated,aes(x=flips,y=Freq))+geom_col()+labs(x="Number of Flips",y="Probability Density",title=
"Probability Density Number of Flips for Two Consecutive Heads")
```



What about more heads in a row?

What do you think the answer for three flips in a row is? With the power of computer programs, we just need to copy the line with one number changed.

```
headsInARow(numHeadsInARow = 3)
```

```
## [1] 13.9731
```

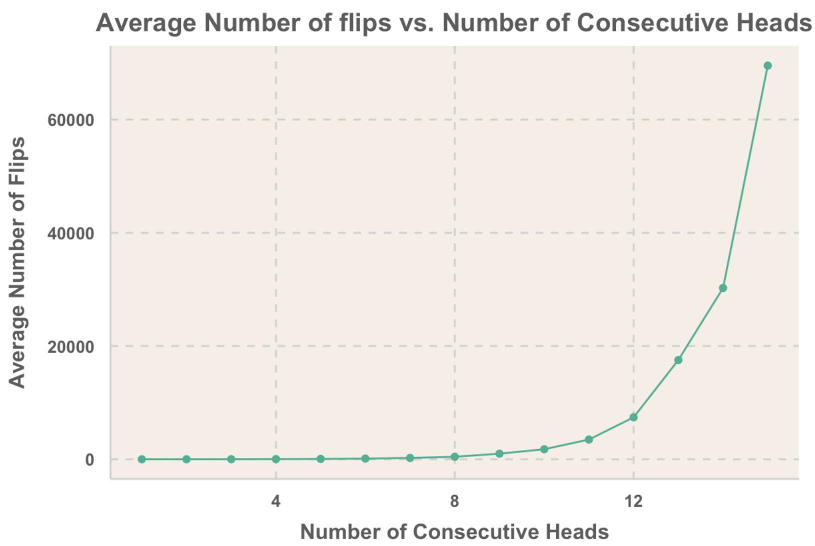
The answer is 14! That is also quite unexpected, and seems higher than expected to most people.

What do you think the answer is for 4, 5, or 10 flips? What does the trend look like? Does it grow linearly, quadratically, exponentially?

Again, with computer simulations and plotting software this may be answered with a few lines of code.

```
findAverageFlips <- function(lowerBound=1,upperBound=10){
  averageNumFlips <- rep(0.0,upperBound-lowerBound+1)
  for(numHeads in lowerBound:upperBound){
    flips <- headsInARow(numHeadsInARow = numHeads,numTrials=100)
    averageNumFlips[numHeads-lowerBound+1] <- flips
  }
  averageNumFlips
}

lower <- 1
upper <- 15
averageNumFlips <- findAverageFlips(lower,upper)
```

```
ggplot()+aes(x=lower:upper,y=averageNumFlips)+geom_point()+labs(x="Number of Consecutive Heads",y="Average Number
of Flips",title="Average Number of flips vs. Number of Consecutive Heads")+geom_line()
```



The trend is exponential! This may have been very challenging to work out by hand, but computer software we can even work out the case with 15 heads in a row. I would run for larger numbers but for larger cases, the probability of getting that many heads in a row gets so small that each trial takes thousands of iterations.

Markov Chains

Although it is nice to do some computer models, I will briefly delve into how we could evaluate the exact average number of trials.

There are a few methods of solving the situation with two or three consecutive heads. The method I will be speaking about is Markov Chains. The essence of Markov Chains is to link different states via probabilities, and then find the equilbrium state of the Markov Chain.

For now, we will work with the two consecutive heads case.

Let us define three states: a_0, a_1, a_2. a_i represents the state where there are currently i consecutive heads. Thus we start flipping at a_0, and a coin flip resulting in heads moves from a_i to a_{i+1}, while tails leads from a_i to a_0.

Now, we may calculate the expected number of flips to reach state H_2 recursively, by determining the expected value for reaching a_1. Denote H_i as the expected number of flips to reach a_i from a_0. This means that $h_0 = 0$.

If we are currently at a_0, there is a 12 chance we flip tails and stay at a_0, and a 12 chance we flip heads and end up at a_1. Thus if we let these states represent the expected number of flips, then we can say:

$H_1 = 12(H_1 + 1) + 12(1)$. Now there are a few important details in the equation. The first term represents that we flip tails, and stay at a_0. Thus the number of flips is 1 plus the expected number of flips to go from a_0 to a_1, or H_1. On the other hand, if we flip heads then we immediately go to H_1, so there is one flip and a 12 chance of that occuring. Solving for this, we obtain $H_1 = 2$ as expected.

Now the amazing thing is that this generalizes beautifully! If we would like H_2, then we can follow the same logic. Imagine that we are already at a_1, meaning we have one consecutive head (our last flip was heads). We may express H_2 as:

$H_2 = H_1 + {}^{12}(H_2 + 1) + {}^{12}(1)$. The reasoning for this is because it takes on average H_1 flips to reach a_1, and from a_1 we can either reach a_2 or return back to a_0. The middle and right terms are identical as before, except we replace H_1 with H_2. Solving for H_2, we get 6, as our simulations predicted.

This formulation is very simple to extrapolate into a recursive formula. By expanding the terms a bit, we have:

$H_n = 2H_{n-1} + 2$

Notice that if we plug this in with $H_2 = 6$, we get $H_3 = 14$ as our model also predicted. This is quite a nice formula from a seemingly bizarre scenario! Notice that this essentially doubles the previous value by 2 and adds a bit, meaning the growth of the function is exponential as our plots showed.

As a bit of closure, the closed form is:

$H_n = 2(2^n - 1)$ I will leave this as an exercise to the reader.

Coin Flip Distribution

Now that we have gone in depth on how to calculate the number of flips to obtain a certain number of consecutive heads, we shall examine how many heads you get when flipping n coins. If we have a fair coin, then if we flip a coin 200 times, how many heads do you expect? The logical answer is 100, since we expect about half the flips to be heads because the coin is fair. However, a tougher question is how likely are you to get exactly 100 heads? Try to think of an exact percentage before continuing. Your intuition may push you to believe that it is quite likely, and we will investigate exactly how likely.

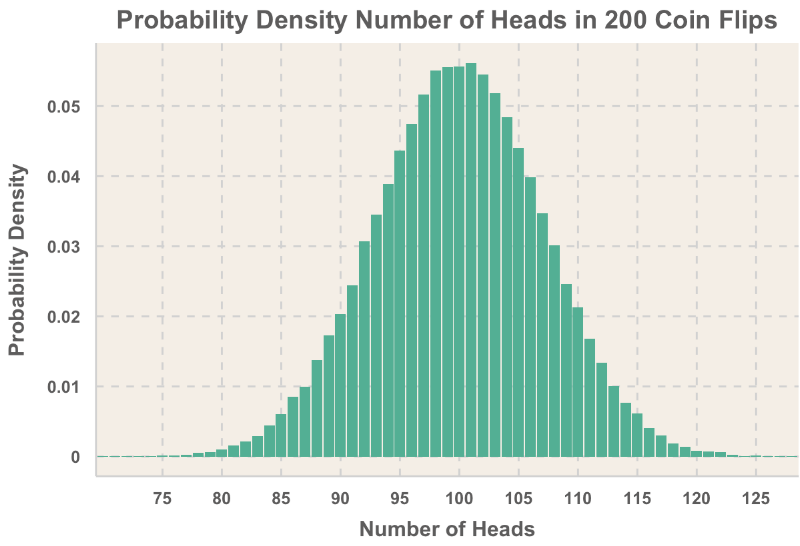
Modeling the Distribution of Heads in 200 Flips

Again, we may model the number of heads in 200 coin flips using the same modeling technique as mentioned before. Creating modular code is invaluable in these scenarios.

```
distributionOfHeads<- function(numFlips=200,numTrials=50000){
  heads <- rep(0,numTrials)
  for(trial in 1:numTrials){
    #keep track number of heads
    numHeads <- 0

    #flip a coin numFlip times
    for(j in 1:numFlips){

      #Simulate a coin flip
      flip <- runif(1)
      #if greater than 0.5, heads
      if (flip>0.5){
        #Add one to the counter of number of heads
        numHeads <- numHeads+1
      }
    }

    #Accumulate total number of flips
    heads[trial] <- numHeads
  }
  #Return the distribution over numFlips flips and numTrials trials
  heads
}
```

The function *distributionOfHeads* returns an array of length *numTrials* where each entry represents the number of heads in *numFlips* flips.

```
heads <- distributionOfHeads(numFlips=200,numTrials=100000)
freqOfHeads <- table(heads)
propHeads <- as.data.frame(prop.table(freqOfHeads))
ggplot(data=propHeads,aes(x=heads,y=Freq))+geom_col()+labs(x="Number of Heads",y="Probability Density",title="Prob
ability Density Number of Heads in 200 Coin Flips")+scale_x_discrete(breaks=seq(75,125,5), labels =seq(75,125,5) )
+scale_y_continuous(breaks=seq(0.00,0.05,0.01), labels =seq(0.00,0.05,0.01) )
```



When running 100000 trials, this involves $200 \cdot 100000$ coin flips, or 20 million flips. This takes quite a while to run. Notice that the graph is relatively normal, as following the central limit theorem. However, something of note is the y-axis. Even for values around 100, the probability density is about 0.05 – 0.06, or 5 – 6%! By looking at the proportions, the probability of getting 100 heads is only about 5.7%, which is a bit more than 1 in 20. What is going on here?

The exact probability of getting 100 heads in 200 flips is $\frac{\binom{200}{100}}{2^{200}}$. As you may remember from your probability class, $\binom{200}{100}=\frac{200!}{100!100!}$. Now how do we evaluate this? Using wolfram alpha, $200!$ is about $7.88\cdot 10^{374}$ so it is obviously not feasible to evaluate something like this. 2^{200} is about 10^{60}, so we are dealing with some huge numbers.

Evaluating Binomial Coefficients

The tool we will use is Stirling's approximation. $n!\approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ The first step to evaluating $p=\frac{200!}{100!100!2^{200}}$ is taking a log. Then we may apply Stirling's approximation.

$\ln p=\ln(200!)-\ln(100!^2 2^{100})=\ln(200!)-2\ln(100!)-200\ln(2)$ If we take a log of Stirling's approximation, we get: $\ln n!\approx \ln(\sqrt{2\pi n})+n\ln n -n$

Plugging this into the equation for $\ln p$, we get: $\ln p\approx \ln(\sqrt{2\pi 200})+(200 \ln (200)-200)-2(\ln(\sqrt{2\pi 100})+(100 \ln (100)-100))-200\ln(2)$

I will now go into how you can use some hand wavey approximations to get a pretty good estimate for p, going through all the assumptions and steps.

The point of using Stirling's approximation is to create a manageable method of evaluating these probabilities, so let's assign some back of the envelope approximations. $e\approx 2.78$, so $e^{4.6}\approx 100$ after testing a few values, $e^{5.3}\approx 200$, and $e^{0.7}\approx 2$. Note that we could also take out the square roots from the natural logs, but in terms of approximation, we can actually use them to help us.

The first term is the natural log of $\sqrt{2\pi 200}$. $400*3.14\approx 1256$, which is pretty much between $35^2=1225$ and $36^2=1296$, so let's say $35.5\approx\sqrt{1256}$. $e^{3.6}\approx 1256$, meaning $\ln(\sqrt{2\pi200})\approx 3.6$. Following the same steps for the $\ln(\sqrt{2\pi100})$, we get $25\approx \sqrt{2\pi 100}$, and $e^{3.2}\approx 25$. Thus $\ln(\sqrt{2\pi100})\approx 3.2$.

Now we may just plug in values!

$$\begin{split} \ln p &\approx 3.6+(200\cdot 5.3-200)-2(3.2+(100\cdot 4.6-100))-200\cdot 0.7\\ &=3.6+(200\cdot 4.3)-2(3.2+(100\cdot3.6))-140\\ &=3.6+860-2(3.2+360)-140\\ &=-2.8\\ p\approx e^{-2.8}=0.0608 \end{split}$$

After all these calculations, we end up with our answer of $0.0608=6.1\%$! The true value of $\frac{\binom {200}{100}}{2^{200}}=5.6\%$, but with a ton of approximations we got quite a respectable answer.

Conclusion

So the general question is usually, so what? The purpose of this blog post is to give examples of broader ideas and perspectives. Writing functions to tackle math problems is incredibly powerful, and can be quite simple at times. They can be a great first step in understand the situation, and can perform mechanical tasks millions of times to test out hypotheses.

Additionally, some intimidating things like evaluating binomial coefficients and dealing with values of up to 10^{374} may be simplified by applying numeric approximations and determining approximations of logs.

Sources

Coin Flip EV Coin Flip EV 2 Markov Process Wiki Markov Chains Stirling's Approximation Wikipedia Stirling's Approximation Wolfram