Statistical Analysis, MSCA 31007, Lecture 1

Hope Foster-Reyes

September 29, 2016

```
options(scipen = 5)
```

Explore the probabilities associated with the experiment of tossing a simulated fair coin multiple times.

1. Convergence of Probability of Tail to 0.5

a. We will check that the frequency of "Tails" (outcome equals 1) converges to 0.5 as the number of tosses grows. What does this say about the fairness of the coin?

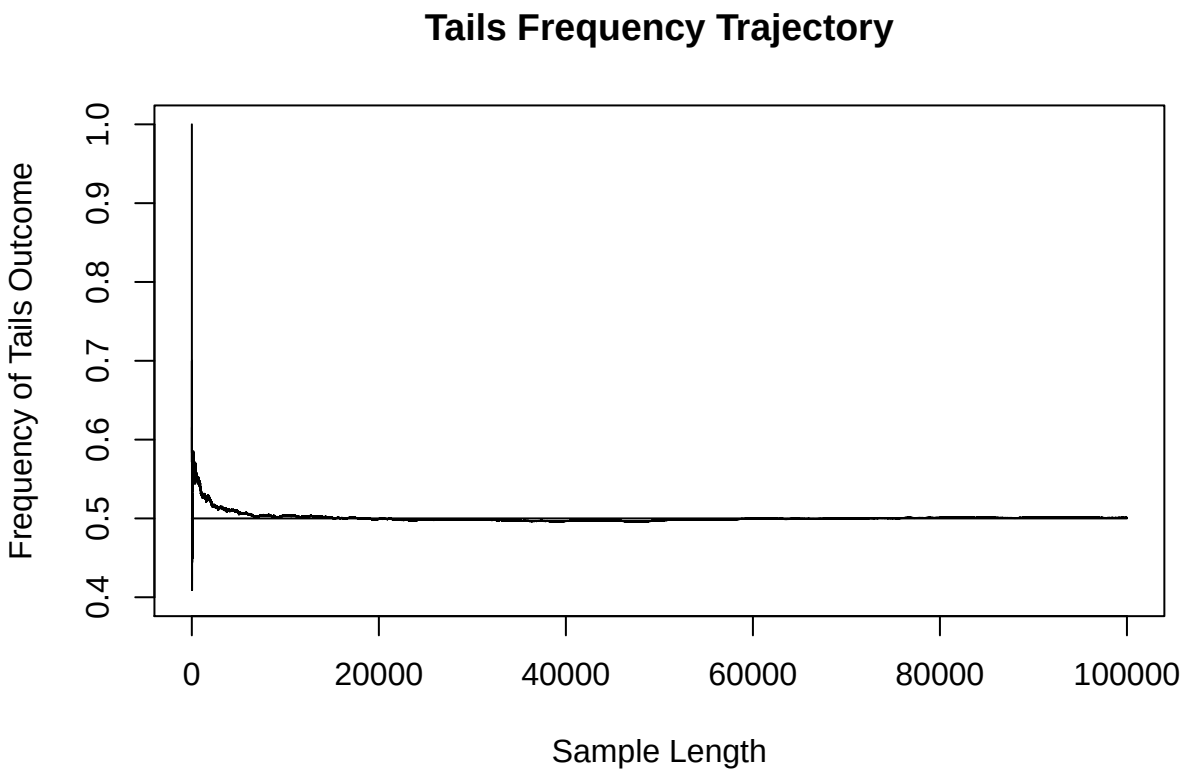
Through the axioms of probability and the Equally Likely Rule, we know that the probability of two equally likely mutually exclusive (disjoint) are equivalent and add to 1, therefore each have a probability of 0.5.

This experiment demonstrates this phenomenon, with our computer-generated pseudo-random simulation of a fair coin. In this case the two equally-likely events are a result of Heads and a result of Tails, each of whose probability is 0.5.

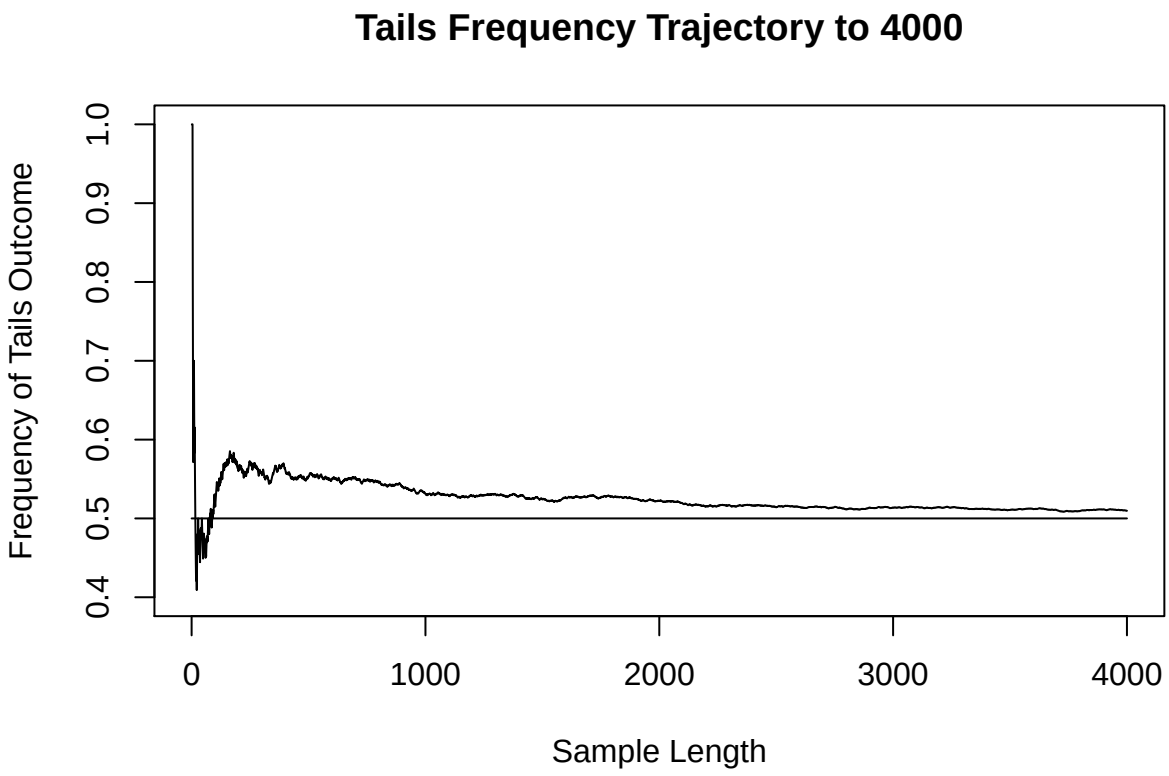
Per the definition of probability and randomness, we also know that in the long run the frequency of the outcome Tails in n empirical trials of this fair coin should converge at 0.5 as n gets larger. Thus we expect the empirical frequency to converge on the theoretical probability of 0.5.

The definition of a fair coin is one one in which both outcomes, Heads and Tails, are equallly likely. Hence by checking that the frequency of Tails converges on 0.5, we are also testing whether our computer-generated coin is fair.

```
# Seed for reproducibility
set.seed(12345)

num.flips <- 100000
#num.flips <- 200000

# Create a random sample of num.flips coin flips, represented by 0 = Heads and 1 = Tails
# Store this sample in the vector flips
flips <- sample(0:1, num.flips, replace = T)

# Create a vector of length num.flips containing the cumulative sum of the flips,
#   incrementing by 1 for each outcome of Tails
trajectory <- cumsum(flips)

# Create a vector of length num.flips containing the running frequency of tails,
#   with each entry representing the calculated frequency after the nth flip
freq.tails <- trajectory / (1:num.flips)

# Create a line graph of all data in the tails frequency vector
plot(1:length(freq.tails), freq.tails, ylim = c(0.4, 1), type = "l",
     ylab = "Frequency of Tails Outcome", xlab = "Sample Length",
     main = "Tails Frequency Trajectory")
lines(c(0, num.flips), c(0.5, 0.5))
```




```
# Create a line graph honing in on the first 4000 entries in the tails frequency vector
plot(1:4000, freq.tails[1:4000], ylim = c(0.4, 1), type = "l",
     ylab = "Frequency of Tails Outcome", xlab = "Sample Length",
     main = "Tails Frequency Trajectory to 4000")
lines(c(0,4000), c(0.5, 0.5))
```



b. Interpret what you see on the graphs.

What we see in the graphs is a behavior in which the trajectory of frequencies jumps up and down erratically as the number of trials is small. The first toss is Tails, so we see a big jump early on.

```
(head(freq.tails, 20))
```

```
##  [1] 1.0000000 1.0000000 1.0000000 1.0000000 0.8000000 0.6666667 0.5714286
##  [8] 0.6250000 0.6666667 0.7000000 0.6363636 0.5833333 0.6153846 0.5714286
## [15] 0.5333333 0.5000000 0.4705882 0.4444444 0.4210526 0.4500000
```

```
(max(freq.tails[0:6]))
```

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

Then the trajectory demonstrates a slightly larger tendency toward Heads, later again a more pronounced tendency toward Tails.

```
(max(freq.tails[20:1000]))
```

```
## [1] 0.5853659
```

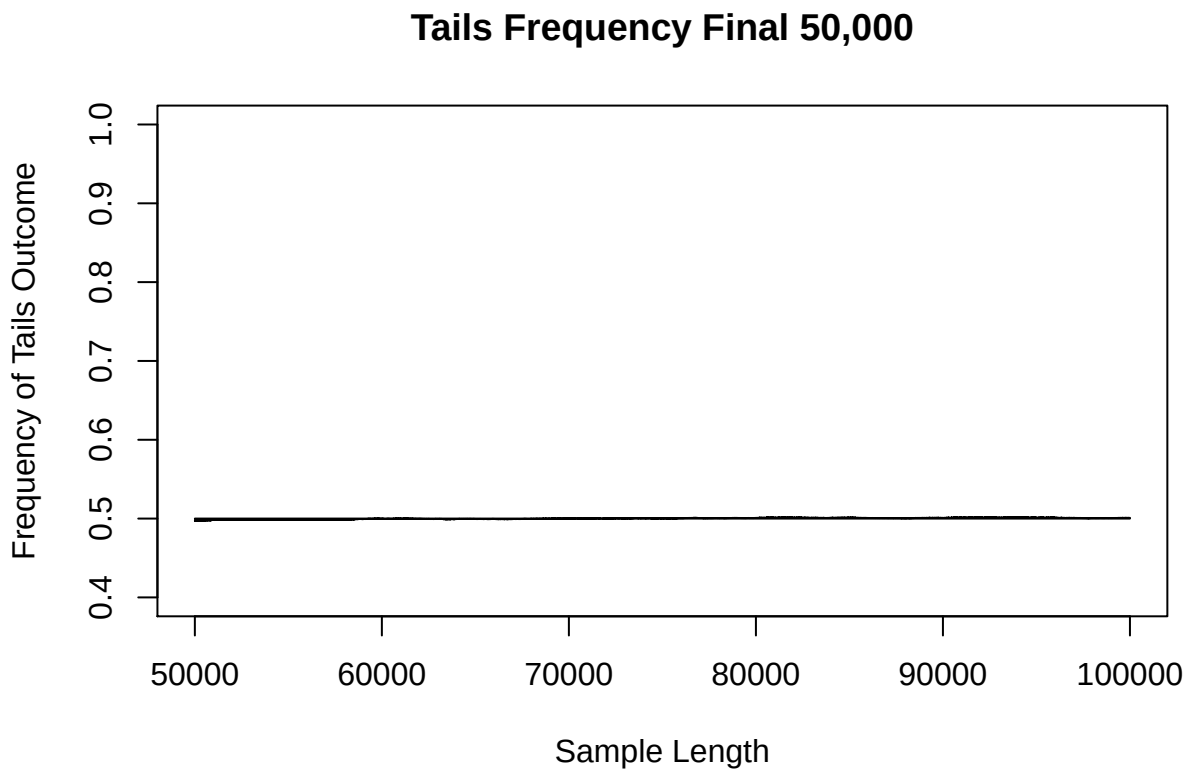
```
(min(freq.tails[20:1000]))
```

```
## [1] 0.4090909
```

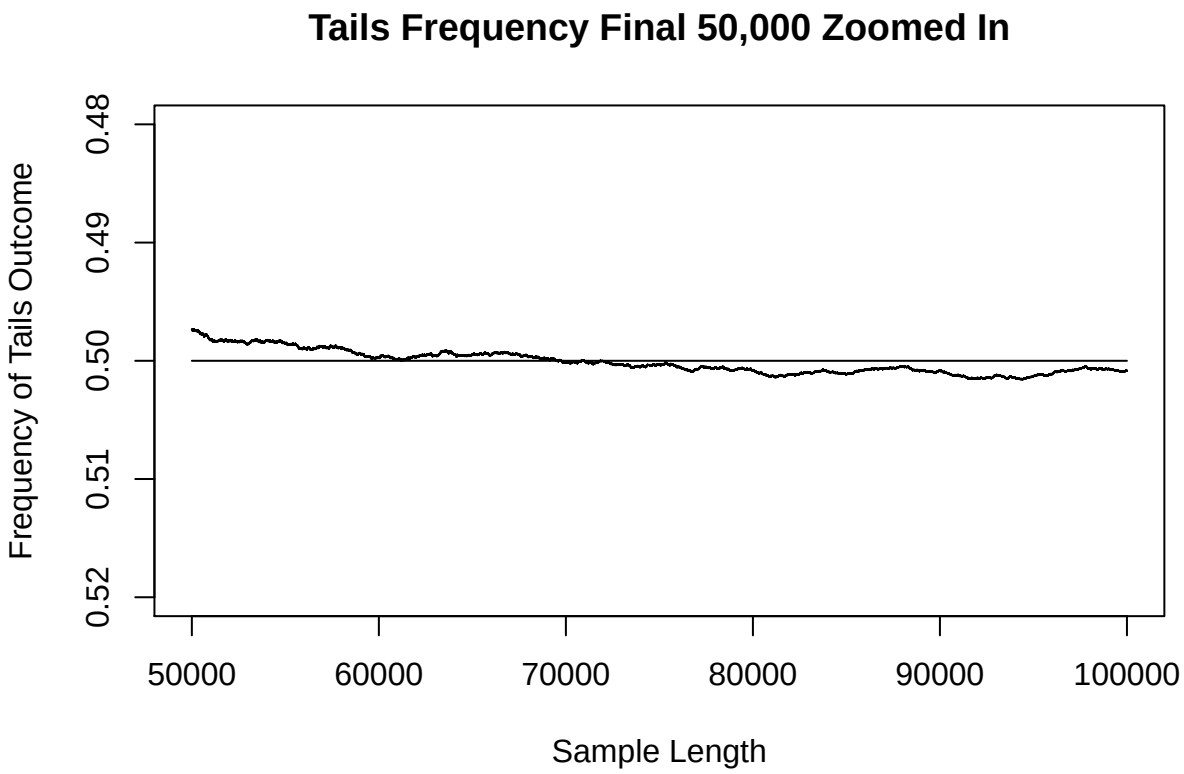
All of which simply demonstrates that our random experiment is unpredictable in the short term. While it may be surprising that the trajectory has such an extended tendency toward tails (lying above 0.5 for most of our experiment), as the trajectory becomes longer and longer (n becomes larger and larger), we can see the frequency of flips with the outcome Tails moving closer and closer to 0.5, the theoretical probability of getting Tails.

Let's take a closer look at the final 5000 entries:

```
# Create a line graph honing in on the next 50000 entries in the tails frequency vector
plot(50000:100000, freq.tails[50000:100000], ylim = c(0.4, 1), type = "l",
     ylab = "Frequency of Tails Outcome", xlab = "Sample Length",
     main = "Tails Frequency Final 50,000")
lines(c(50000, 100000), c(0.5, 0.5))
```




```
# And Zoomed In
plot(50000:100000, freq.tails[50000:100000], ylim = c(0.52, 0.48), type = "l",
     ylab = "Frequency of Tails Outcome", xlab = "Sample Length",
     main = "Tails Frequency Final 50,000 Zoomed In")
lines(c(50000, 100000), c(0.5, 0.5))
```



When we change the scale, we can see that our experiment still varies and is varying both above and below the 0.5 line. But at the original scale the frequency is nearly indistinguishable from a straight line at the 0.5 frequency mark. At the 100,000th flip, the tail frequency is 0.50084.

2. Check Your Intuition About Random Walks

2.1 One Trajectory

```
# Seed for reproducibility
set.seed(12345)

# Increase the number of flips
num.flips <- 1000000

# Create a random sample of num.flips coin flips to simulate a gambling game, where
#   Heads loses $1 and Tails pays $1
flips.wealth <- (sample(0:1, num.flips, replace = T) - 0.5) * 2
(table(flips.wealth))
```

```
## flips.wealth
##     -1      1
## 499933 500067
```

a. Find at least one alternative way of simulating variable Flips (in my code, flips.wealth).

```
# This is a transformation of ~ Binom(1, 0.5), so we can also produce the
#   sample with rbinom()
set.seed(12345)
binom.wealth <- rbinom(num.flips, 1, 0.5)
flips.wealth <- (binom.wealth - 0.5) * 2
(table(flips.wealth))
```

```
## flips.wealth
##     -1      1
## 499933 500067
```

```
# We can also simply hard code our options
set.seed(12345)
flips.wealth <- sample(c(-1, 1), size = num.flips, replace = T)
(table(flips.wealth))
```

```
## flips.wealth
##     -1      1
## 499933 500067
```

b. Check your intution by answering questions before calculation:

How much do you expect the trajectory of wealth to deviate from zero?

Intuitively we can expect the trajectory of wealth to deviate from zero as much as $2-$3, as we imagine that the coin could perhaps immediately land on Tails 2-3 times.

What is the probability that this intuitive guess is underestimating, and that instead the coin will land on Tails 4 or 5 times in a row in its initial flips?

```
# This can be calculated easily as the probability of 4-5 successes in the
#   distribution X ~ Binom(5, 0.5)

(dbinom(0:5, size = 5, prob = 0.5))
```

```
## [1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
```

So, in 5 tosses there is a ~15% proability that we would get 4 tails in a row, and a 3% probability that we would get 5 tails a row. It seems like our prediction is likely under; let's find out how much.

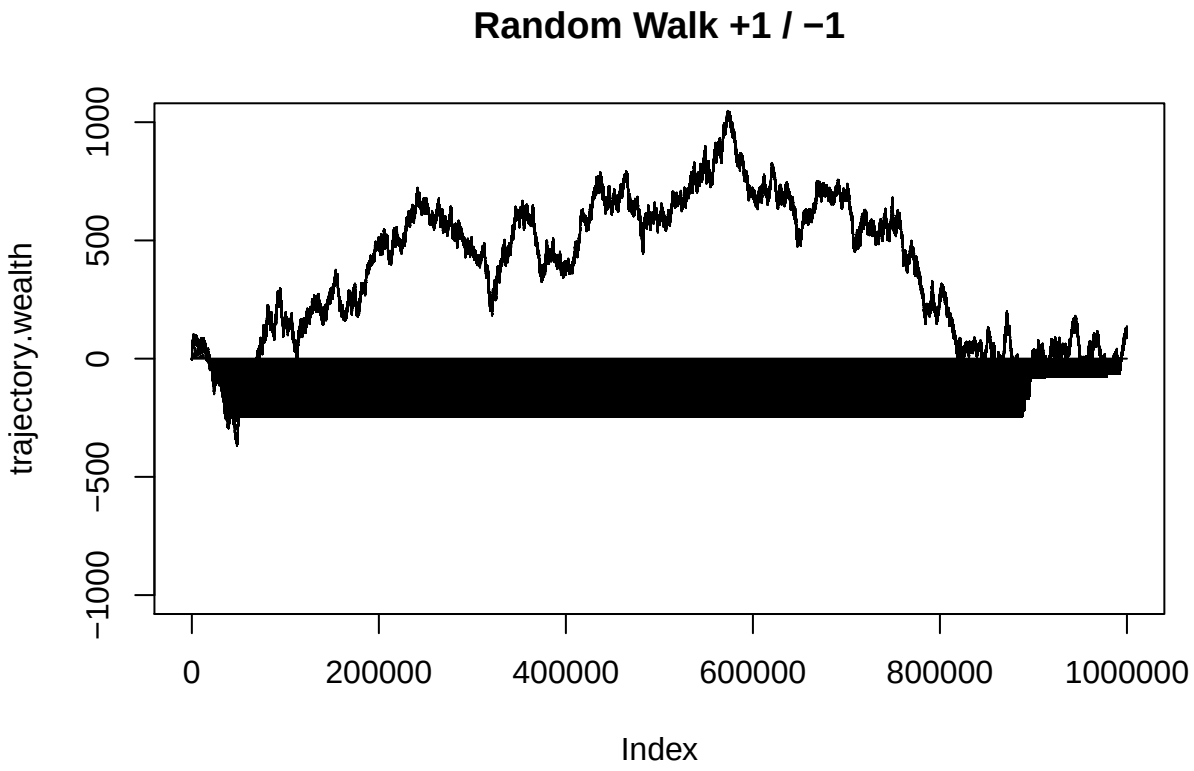
How long do you expect it to stay on one side above or below zero?

Intuitively have a sense that the trajectory will erraticaly jump above and below zero. Since remaining above or below the line would represent consecutive flips of Heads or Tails, we estimate that, similar to our above guess, that the wealth of our hypothetical gambler would not often remain consecutively negative or positive longer than 3-4 flips.

c. How do the observations match our prior expectations?

We run the experiment:

```r
# Create our random walk, a vector of length num.flips containing the cumulative sum of
#   the flips, incrementing by 1 for each outcome of Tails, and -1 for each outcome of Tails.
#   This is a one-dimensional random walk, moving +1 or -1 with equal probability.
trajectory.wealth <- cumsum(flips.wealth)

# Create a line graph of our wealth trajectory. In this case rather than plotting the
#   frequency of Tails we are plotting the position, in terms of wealth, of our gambler,
#   as she 'walks along randomly' in either the +1 or -1 direction.
plot(trajectory.wealth, ylim = c(-1000, 1000), type = "l",
     main = "Random Walk +1 / -1")
lines(c(0, num.flips), c(0, 0))
```



And our expectations are wildly inaccurate!

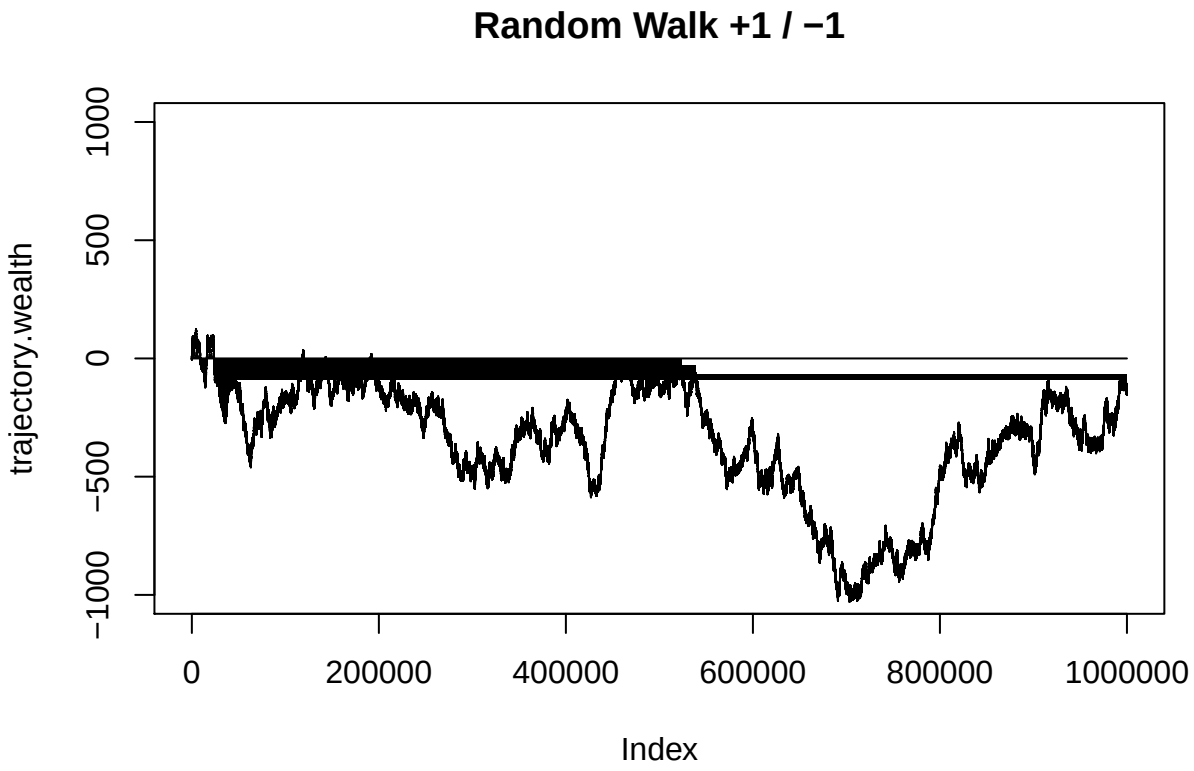
While we did consider the probability the coin would land on 4-5 consecutive heads or tails, and that probability is small, what we did not consider is the compound effect of repeated tendencies toward heads or tails. It's not just 3/5 that causes a trend away from zero, but 3/5 (or 4/5 or 5/5) multiple times compounded.

Our sample demonstrates a general upward trend from ~50k to 600k trials, despite constant fluctuations up and down within that trend.

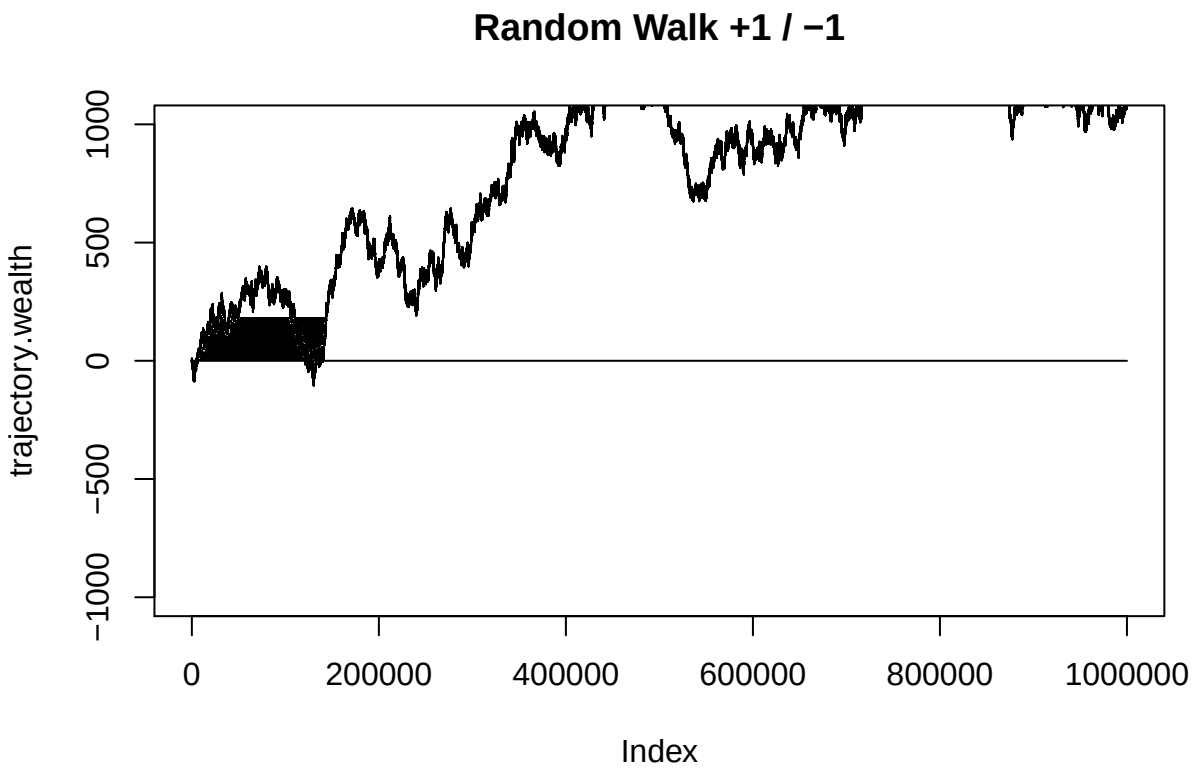
Running the experiment with alternate seeds:

```r
set.seed(33333)
flips.wealth <- sample(c(-1, 1), size = num.flips, replace = T)
trajectory.wealth <- cumsum(flips.wealth)
plot(trajectory.wealth, ylim = c(-1000, 1000), type = "l",
```

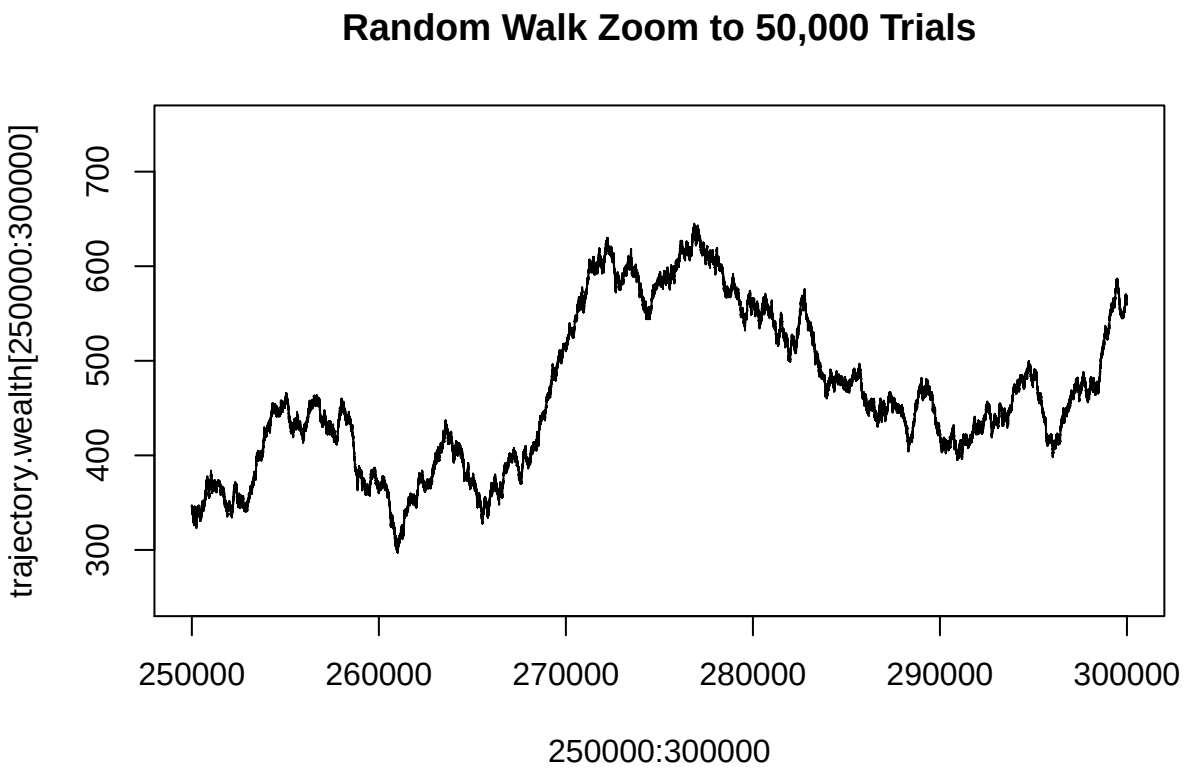
```
     main = "Random Walk +1 / -1")
lines(c(0, num.flips), c(0, 0))
```



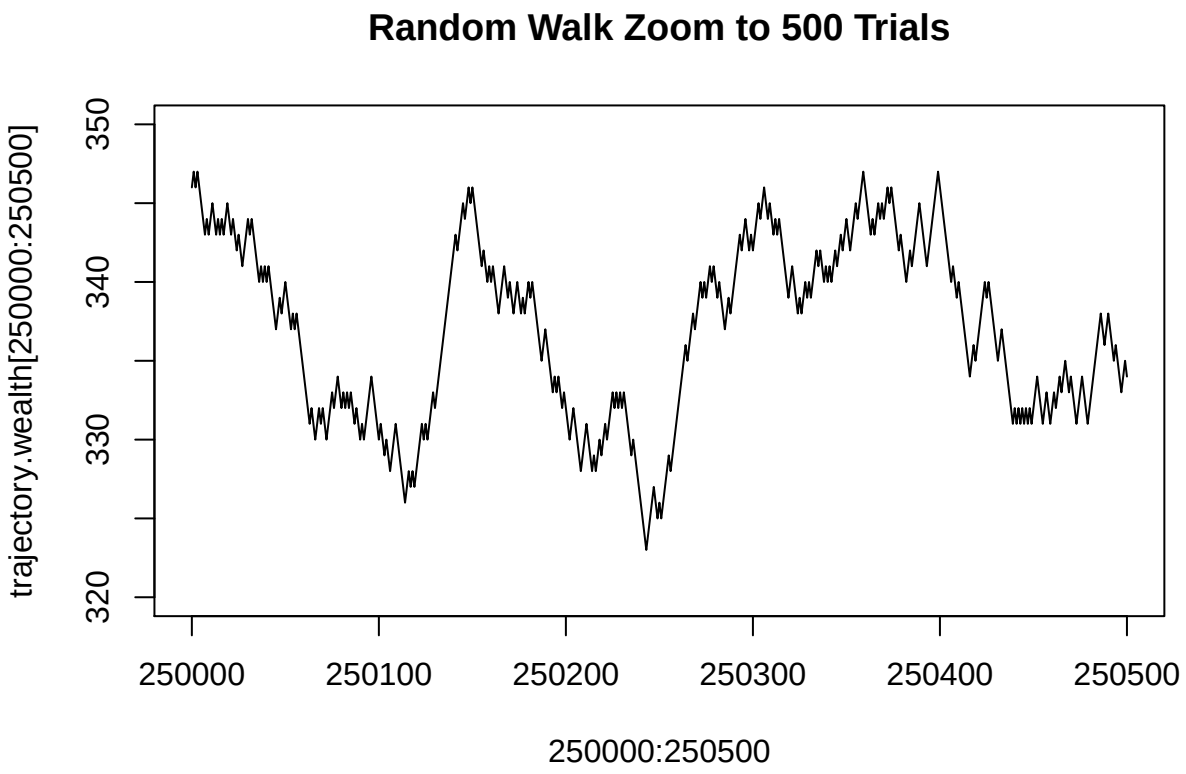

```
set.seed(54321)
flips.wealth <- sample(c(-1, 1), size = num.flips, replace = T)
trajectory.wealth <- cumsum(flips.wealth)
plot(trajectory.wealth, ylim = c(-1000, 1000), type = "l",
     main = "Random Walk +1 / -1")
lines(c(0, num.flips), c(0, 0))
```




```
# Zooming In
plot(250000:300000, trajectory.wealth[250000:300000], ylim = c(250, 750), type = "l",
     main = "Random Walk Zoom to 50,000 Trials")
lines(c(250000, 300000), c(0, 0))
```




```
# Zooming In
plot(250000:250500, trajectory.wealth[250000:250500], ylim = c(320, 350), type = "l",
     main = "Random Walk Zoom to 500 Trials")
lines(c(250000, 250500), c(0, 0))
```



So, while our intuition correctly perceived that there would be fluctuations within a set of 5-10 trials, what we didn't foresee was that these fluctuations compound on themselves and occur at larger and larger scales, as shown by our sample in which the distance from zero extends beyond our original y-axis limit.

The zoomed charts demonstrate that our fluctuations appear remarkable similar as we zoom in, on scales of 1 million, 300k and only 500 trials.

2.2 Multiple Trajectories

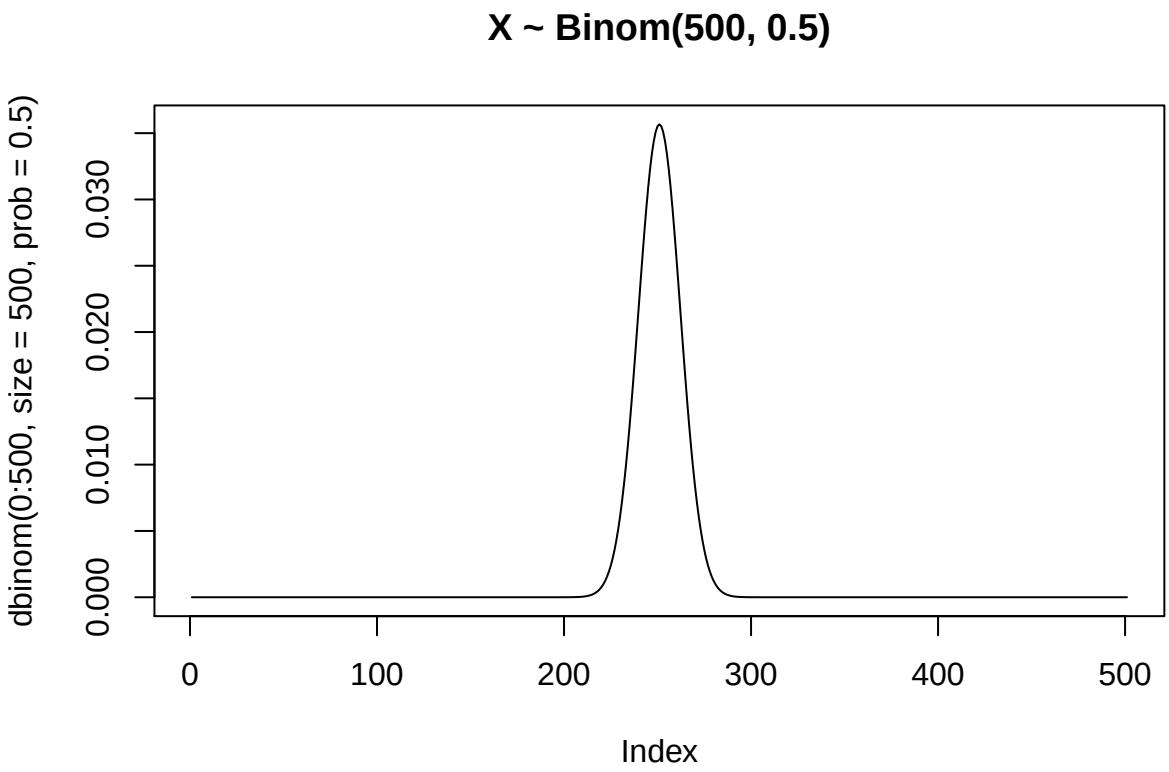
In our next experiment we look at the probability of our trajectory ending a certain distance from zero.

a. What do you expect the probabilities of the following events to be?

N_h ***is the number of "Heads" and*** N_t ***is the number of "Tails" in 500 coin flips.***

Let's look at the distribution of the number of "successes" in 500 coin flips, X ~ Binom(500, 0.5):

```
plot(dbinom(0:500, size = 500, prob = 0.5), type = "l", main = "X ~ Binom(500, 0.5)")
```



Estimate $P(|N_h - N_t| < 5)$***:***

This is the probability that the absolute value of the number of successes minus the number of failures will be less than 5, or, in other words, the successes minus failures are within 5 of zero.

Doing a little algebra, we can express this in terms of the number of successes, if x is our successes and x' is our failures:

$P(|x - x'| < 5)$

$P(|x - (500 - x)| < 5)$

$P(|2x - 500| < 5)$

$P(247.5 > x < 252.5)$

This makes sense, our event is those outcomes in which the number of successes is within 2.5 of the center of our distribution.

Without doing the math, we'll guess that this will be about a 10% probability.

Estimate $P(|N_h - N_t| > 25)$***:***

In this example, we're looking for a number of successes that is within 12.5 of our center.

Let's guess that this is a 25% probability.

b. Estimate the Probabilities

To run an experiment that will empirically estimate these probabilities, we'll convert the flips.wealth sample of 1,000,000 coin flips into a matrix of 2000 random walk samples, each 500 long:

```r
# Recontruct sample with original seed
set.seed(12345)
flips.wealth <- sample(c(-1, 1), size = num.flips, replace = T)

# Convert the sample into a matrix with 500 columns and 2000 rows
matrix.wealth <- matrix(flips.wealth, ncol = 500)
# Then apply cumsum over each row and transform, so that each row is a random walk
trajectories.matrix <- t(apply(matrix.wealth, MARGIN = 1, cumsum))

# Observe the dimensions of our matrix
dim(trajectories.matrix)
```

```
## [1] 2000  500
```

```r
# Using our final column (cumsum), calculate the proportion of our 2000 walks that ended
#   less than 5 away from the zero starting point
size.sample <- 2000 # Each random walk is a sample in our experiment
size.event.under.5 <- sum(abs(trajectories.matrix[,500]) < 5) # Count each valid result
(p.under.5 <- size.event.under.5 / size.sample)
```

```
## [1] 0.18
```

```r
# Calculate this proportion for those walks that ended less than 25 away
size.event.under.25 <- sum(abs(trajectories.matrix[,500]) < 25) # Count each valid result
(p.under.25 <- size.event.under.25 / size.sample)
```

```
## [1] 0.7485
```

While mathematically equivalent, our random walk matrix approaches the original problem of number of heads minus number of tails from a different angle than the probability distribution we charted above.

On our chart of the distribution each iterative success represents x+1. Each failure does not impact x. * Our distribution is from 0 to 500. * Our expected value is in the centerl of this distribution. * 250 represents exactly the same number of heads and tails, 251 represents one more head than tails, and 249 one less.

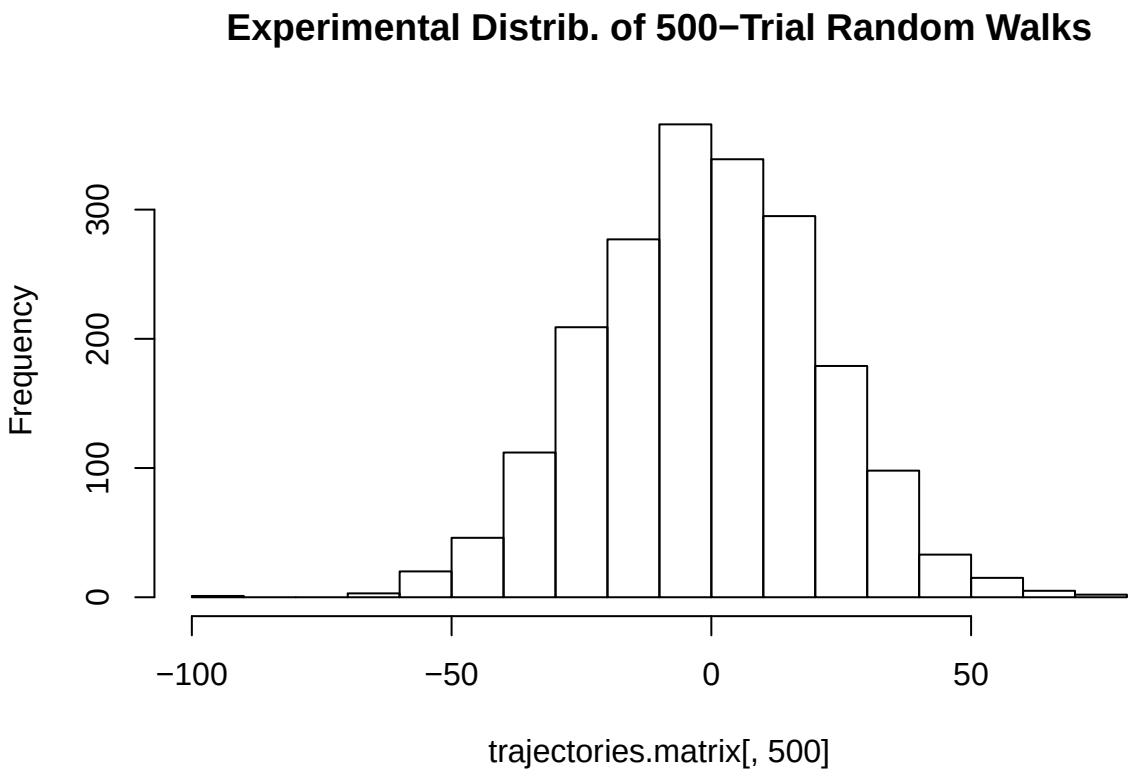
On the other hand, our random walk represents each success as +1 and each failure as -1. * Our possible values range from -250 to 250. * Our expected value is 0. * 0 represents exactly the same number of heads and tails, 1 represents one more head than tails, and -1 one less.

In our trajectory matrix, rather than each entry representing a flip which would need to be totaled to give us our random variable, each entry represents the current running value of our random variable as it progresses through the trials.

Therefore the final column of the matrix is the final result of our experiment. By summing this column and dividing by our number of random walks, 2000, we arrive at an estimate of the distance away from our expected value each of the 2000 experiments achieves.

Let's chart now a histogram of our 2000 experiments:

```r
hist(trajectories.matrix[,500], main = "Experimental Distrib. of 500-Trial Random Walks")
```



c. How many times out of 2,000 runs:

Do trajectories end less than 5 points away from zero (5 is 1% of 500 tosses)?

360 or 0.18 of our trajectories ended less than 5 points away from zero.

Do trajectories end more than 25 points away from zero (25 is 5% of 500 tosses)?

1497 or 0.7485 of our trajectories ended less than 25 points away from zero.

d. Interpret the results. How did they correspond to your intuition?

Our estimate of the number of trajectories under 5 was roughly accurate, but our estimate of the number under 25 was greatly under.

??? XXX ???

2.3 Time On One Side

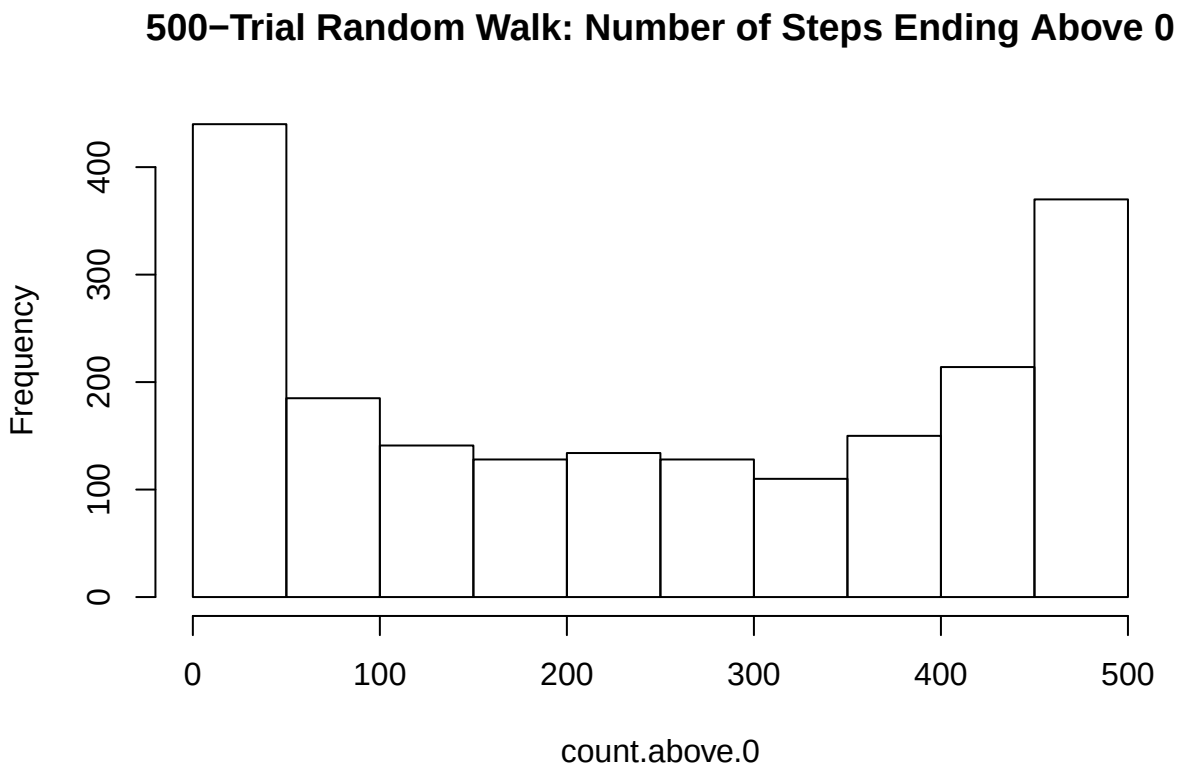
a. How long do you expect trajectory of random walk to spend on one side from zero, below or above?

Given heads and tails are equally likely we would naturally expect the random walk to spend half of its time on one side and the other half on the opposite side. We have seen that for any individual walk this can widely vary, however for 2000 trajectories we expect our result to be 0.5 plus or minus 10%.

b. Interpret the results. Was your intuition correct?

We'll choose time above zero and calculate that proportion for our sample of 2000 random walks.

```
# For each random walk trajectory, count the number of steps that ended above 0
count.above.0 <- apply(trajectories.matrix, MARGIN = 1,
                       function(z) sum(z > 0))

# Chart the distribution
hist(count.above.0, main = "500-Trial Random Walk: Number of Steps Ending Above 0")
```



Our estimate is correct in a way–the average number of steps above 0 in our sample of 2000 is approximately 250.

```
(mean.above.0 <- mean(count.above.0))
```

```
## [1] 244.254
```

However, surprisingly this central number does not represent the majority of observations, and far more common is those walks spending dramatically more or less of their time on one side of zero.

c. Explain the observed distribution.

This is a U-shaped bi-modal distribution whose greatest frequencies are at the two extremes of the variable rather than in the center.

d. Search for the name of the law that we are observing on the last histogram.

This behavior is known as the arcsine law of random walk theory, which is a special case of the beta distribution, as described:

- https://en.wikipedia.org/wiki/Arcsine_laws_(Wiener_process)
- https://en.wikipedia.org/wiki/Arcsine_distribution
- https://en.wikipedia.org/wiki/Beta_distribution

Test: Relationship Between Slope and Correlation

```
# Read sample data. Store sample.csv in working directory of project.
sample <- read.table('sample.csv', header = T)

(sd.X <- sd(sample$x))
```

```
## [1] 2.811966
```

```
(sd.Y <- sd(sample$y))
```

```
## [1] 2.258984
```

```
(cor.XY <- cor(sample$x, sample$y))
```

```
## [1] 0.9113875
```

```
(cov.XY <- cov(sample$x, sample$y))
```

```
## [1] 5.789304
```

```
(a <- cov.XY / (sd.X)^2)
```

```
## [1] 0.7321603
```