Simulations Part 2

Side Note

```
w<-c(TRUE, TRUE, FALSE)
```

```
w
```
output: TRUE TRUE FALSE

```
class(w)
```
output: logical

```
as.numeric(w)
```
output: 1 1 0

Conclusion: Changing the class of a logical vector into a numeric vector, changes the TRUE for 1 and the FALSE for 0.

Types of Probabilities

1. Theoretical Probabilities

2. Experimental Probabilities

Law of Large Numbers

Law of Large Numbers: In the long run, as the number of trials increases and increases, the proportion of the outcomes get closer to the theoretical probability values.

Number of Tosses	Number of Observed Heads	Percent of Observed Heads	Expected Percent of Heads
10	6	60%	50%
100	48	48%	50%
500	271	54.2%	50%
1000	461	46.1%	50%
5000	2533	50.66%	50%
10,000	5081	50.81%	50%

Experiment: Flipping a Coin

Theoretical Probability of Getting Heads: 0.5

Experimental Probability of Getting Heads: We need to flip the coin n times and see how many times you "see" heads. Divide that number by the number of flips.

Flip	Outcome	Is it a Head? TRUE/FALSE
1	Tails	FALSE
2	Tails	FALSE
3	Tails	FALSE
4	Tails	FALSE
5	Heads	TRUE
6	Heads	TRUE
7	Heads	TRUE
8	Tails	FALSE
9	Heads	TRUE
10	Tails	FALSE
11	Tails	FALSE
12	Tails	FALSE
13	Tails	FALSE
14	Tails	FALSE
15	Heads	TRUE
16	Tails	FALSE
17	Heads	TRUE
18	Heads	TRUE
19	Tails	FALSE
20	Heads	TRUE

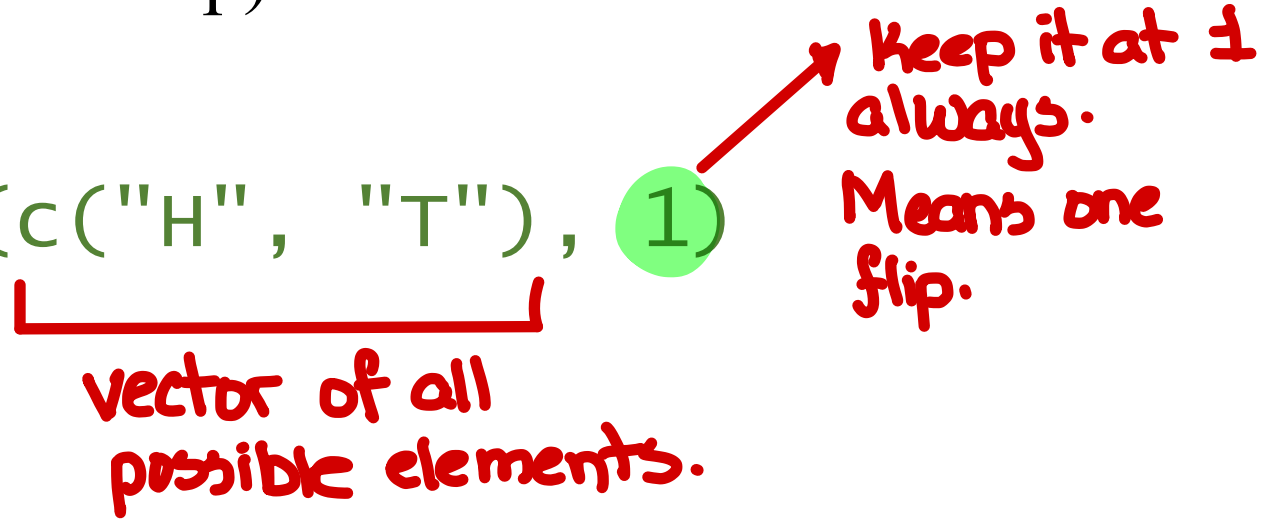
Last class we learned how to simulate 1 flip.

experimental prob of getting heads $= \frac{8}{20} = 0.40$

Simulations

Step 1: Simulate One Trial (One Flip)

```
coin_flip <- sample(c("H", "T"), 1)
```



Step 2: Check the criteria that you are trying to compute the probability on.

Checking if that flip is "Heads" (because that is the one that you wanted to compute the probability on)

```
check_flip <- coin_flip == "H"
```

Boolean expression
TRUE if meet the criteria you are trying to compute the probability on.

Step 3: Create a Function that performs 1 Trial and checks if it met the criteria.

In our example, the function will flip the coin once and checks if it is Heads

```
coin_function <- function(i) {
  coin_flip <- sample(c("H", "T"), 1)
  check_flip <- coin_flip == "H"
  return(check_flip)
}
```

You can set i = to anything and it will run the function.

Step 4: Run the function n times (i.e., 100000 times)

Means you flip the coin n times (i.e., 10000 times) and every time you flip you record if it was a heads or not.

vector

function name

```
iterations <- map_dbl(1:100000, coin_function)
```

output: 1 1 0 0 0 1 0 1 ...

This means H H T T T H T H ..

map_dbl will run the function (flip the coin) 100,000 times (the number of elements in the vector.

Step 5: To get the experimental probability, sum the number of times the criteria was met, divided by the total number of trials n.

Sum the number of times you got heads and divide it by the number of times you flipped the coin.

```
experimental_prob <- sum(iterations)/length(iterations)
```

Total # of Heads

Total # of flips

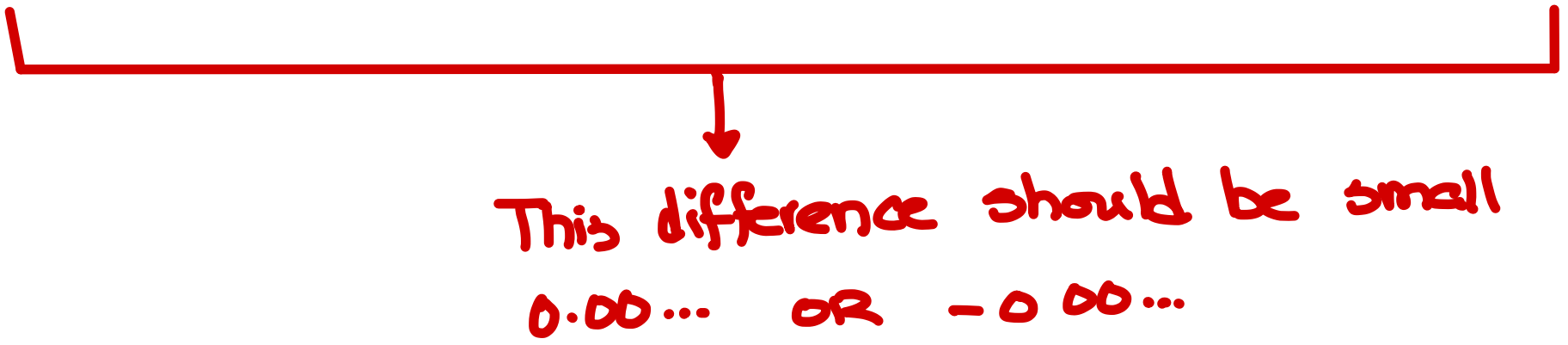
We expect this number to be very close to 0.5

Step 6: Theoretical probability

```
theoretical_prob <- 1/2
```

Step 7: Theoretical probablity vs experimental

```
theoretical_prob - experimental_prob
```



Full Codes

```
coin_function <- function(i) {
   coin_flip <- sample(c("H", "T"), 1)
   check_flip <- coin_flip == "H"
   return(check_flip)
}
iterations <- map_dbl(1:100000, coin_function)
experimental_prob <- sum(iterations)/length(iterations)
theoretical_prob <- 1/2
theoretical_prob - experimental_prob
```