# Confidence Interval

August 5, 2020

## 1 Estimating the Bias of a Coin

Let's say we are tossing a coin but we don't know the bias of the coin and we want to find out the bias. So we toss the coin 500 times. The coin has a bias of $p = 0.7$ in reality (meaning that the probability of heads on any toss is $p = 0.7$ independent of all other tosses), but we don't know that.

[18]:
```
NUM_TOSSES = 500   # change this!
TRUE_BIAS = 0.7    # also change this!

tosses = stats.bernoulli.rvs(TRUE_BIAS, size=NUM_TOSSES)
tosses
```

[18]:
```
array([0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0])
```

Above, `1` represents heads and `0` represents tails. Now we can take the mean of all the tosses to find the bias.

[19]:
```
estimated_bias = np.mean(tosses)
estimated_bias
```

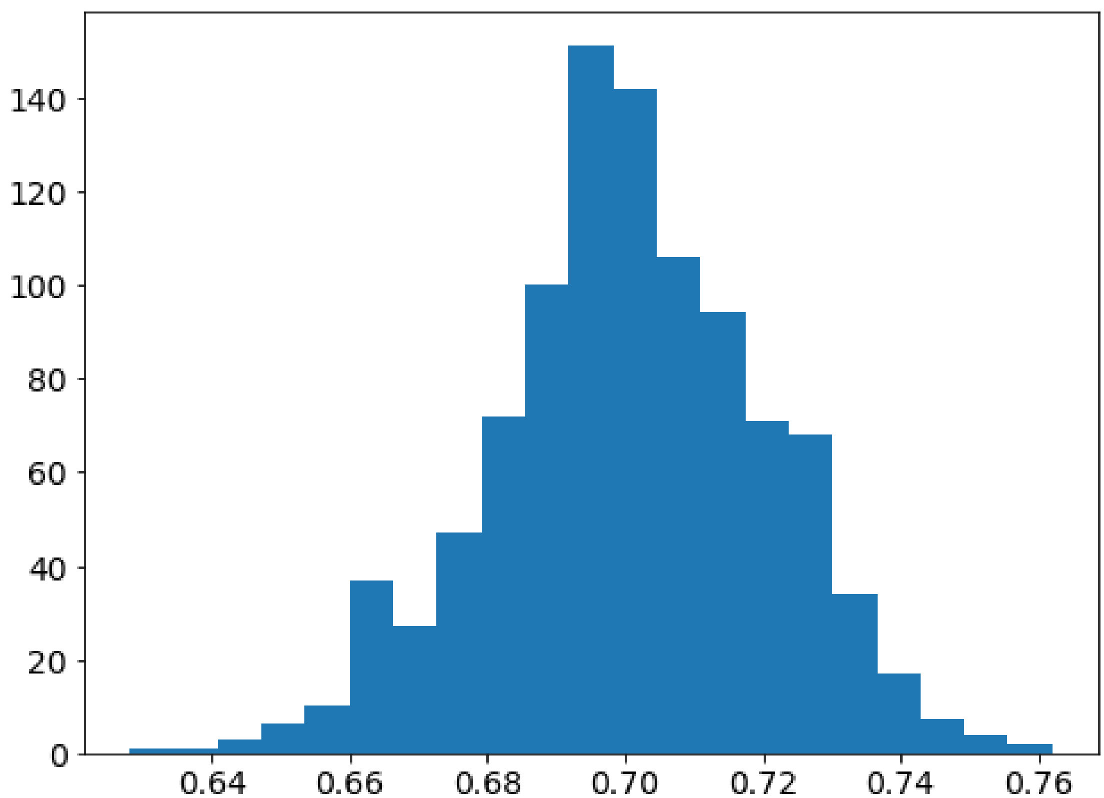
[19]: 0.726

But how confident are we in this estimate? What happens when we toss the coin 100 (or `NUM_TOSS`) times again and calculate the bias? Let's run the entire experiment (tossing `NUM_TOSS` times and finding the mean) `NUM_EXPERIMENTS` times and plot a histogram of the estimates we get.

[30]:
```
NUM_EXPERIMENTS = 1000  # change this!

def experiment():
    """Toss a coin `NUM_TOSS` times and calculate the bias by taking
    the mean number of heads in the sample.
    """
    tosses = stats.bernoulli.rvs(TRUE_BIAS, size=NUM_TOSSES)
    estimated_bias = np.mean(tosses)
    return estimated_bias

samples = [experiment() for _ in range(NUM_EXPERIMENTS)]
plt.hist(samples, bins=21)
plt.show()
```



We can see that there is some variability in the estimate that we can obtain. Most of the time we get an estimate which is close to the actual bias of 0.7, but sometimes we get values as low as 0.6 and as high as 0.8 (run the cell multiple times, since this is random, the range varies a little between runs). So if I were to tell you that you must be correct with 95% probability, what range of values would you tell me the bias of the coin might be? You will be able to answer this question after doing question 3 on the Discussion 7B worksheet.