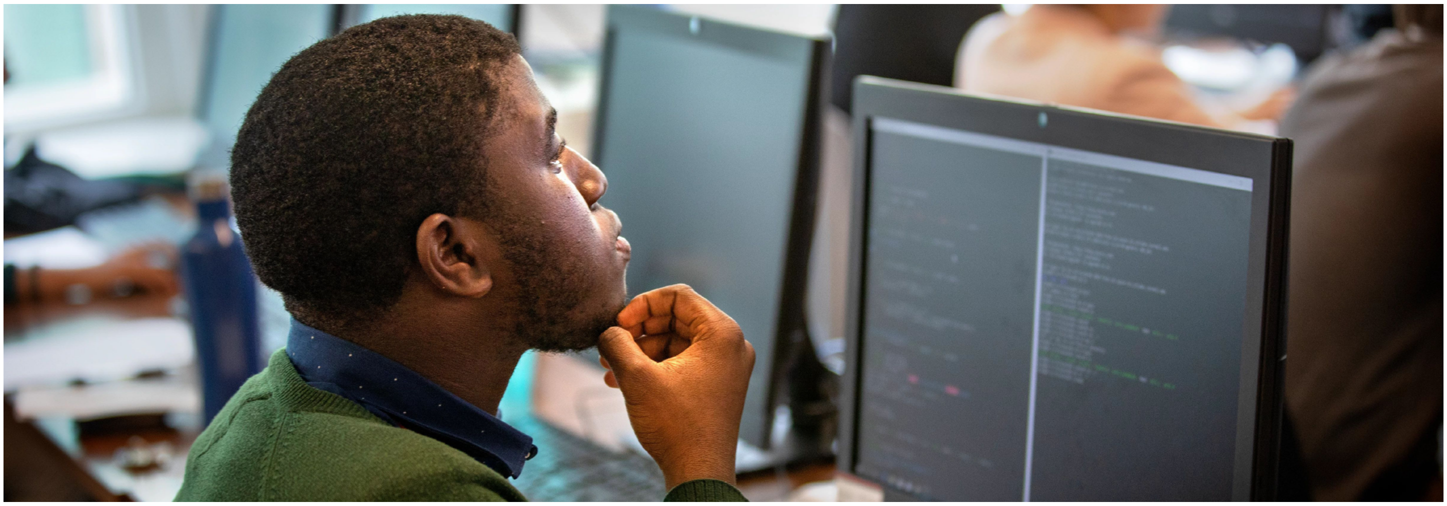
TOOL

Using Simulation to Understand Uncertainty

Collecting multiple samples to evaluate the uncertainty around your results can be expensive and time consuming. In data science, simulation offers a cost-effective method to understand uncertainty. Essentially, simulation uses computers to mimic the process of drawing many different samples from a population. Use this tool to help you run a simulation and understand the uncertainty associated with your results.



Running a Simulation

When you run a simulation, you start by asking the question: What would I see in my data if there was nothing interesting going on in the population? By starting with this question and using the simulation to examine the variation inherent in your data, you examine the chance that you would have seen an interesting result in your sample just due to randomness. If the chances that you would see the same result if nothing unexpected was happening are small, you should be confident that what you see in your sample is a real signal and not just due to chance. However, if a large sample statistic is often observed due to randomness, you should not be very certain that the conclusions from your sample should be generalized to the larger population.

Here is a step-by-step guide to performing a simulation in R:

Step 1: Use the `sample()` function to simulate a random sample of the same size as your data set. Set the sample function so that it matches what you would expect based on current knowledge (nothing interesting is going on in the population). Calculate the sample statistic on this simulated sample.

Step 2: Use a `for` loop to repeat your simulation from Step 1 over and over. This will create many many new sampled data sets under the assumption that nothing unexpected is going on in the population. Calculate the sample statistic with each iteration, and keep track of them in a vector.

Step 3: Use the vector of sample statistics from Step 2 to draw a histogram of the null distribution of the sample statistic. Use that histogram to see how the sample statistic varies from sample to sample even under the baseline assumption that nothing interesting was going on. Calculate the mean and standard deviation of this histogram.

Using R With This Tool

The portions of this tool with a grey background are code text you can use to do the examples included in this tool. You can also modify them to use with your own data. In these examples:

- Commands are the lines of code that don't begin with a pound sign (#). Type these lines into R to carry out the command.
- Commented text begins with one pound sign and explains what the code does.
- Code output begins with two pound signs.

Running a Simulation in R

Now we will follow these steps in an example. Suppose it is well-known that 50% of all cases of common cold are cured within a week without medicine. A new drug to treat the common cold was tested on ten patients, and seven patients reported that their cold was gone within a week, so you see a higher success rate (70%) than the baseline of 50%. How certain should you be that this is a real signal, i.e., the drug helps cure common cold, and that this result isn't due to random chance? If eight patients reported recovery within a week, you would probably be more certain, but how much more? What if nine out of ten patients reported recovery?

Step 1: In the code chunk below, we have set the probability that the drug works to 0.5, created a vector result that stores the outcome of ten simulated patients, and calculated the sample statistic — success rate of the drug — in the variable `p_sim`.

You can run this code chunk to see that the success rate is 60%. This variation is just due to randomness, since we explicitly set the population success rate at 0.5.

```
set.seed(1) # Set the seed for reproducibility

outcome = c("Worked", "Did not Work") # Vector of possible outcomes
result = sample(outcome, 10,       # Pull 10 samples from the outcome vector
     replace = TRUE, prob = c(0.5, 0.5)) # The probability of picking each
                                         # outcome is 0.5
result # View the resulting vector:
##  [1] "Did not Work" "Did not Work" "Worked"       "Worked"       "Did not
## Work"
##  [6] "Worked"       "Worked"       "Worked"       "Worked"       "Did not
## Work"

p_sim = mean(result == "Worked") # calculate the proportion that "Worked"
p_sim # View the proportion:

## [1] 0.6
```

Step 2: Use a `for` loop to simulate a large number (`nsim = 100000`) of random samples (each with size 10) from the population assuming the drug works 50% of the time. Calculate the sample statistic of each random sample, and store these sample statistics in a vector `store_p`.

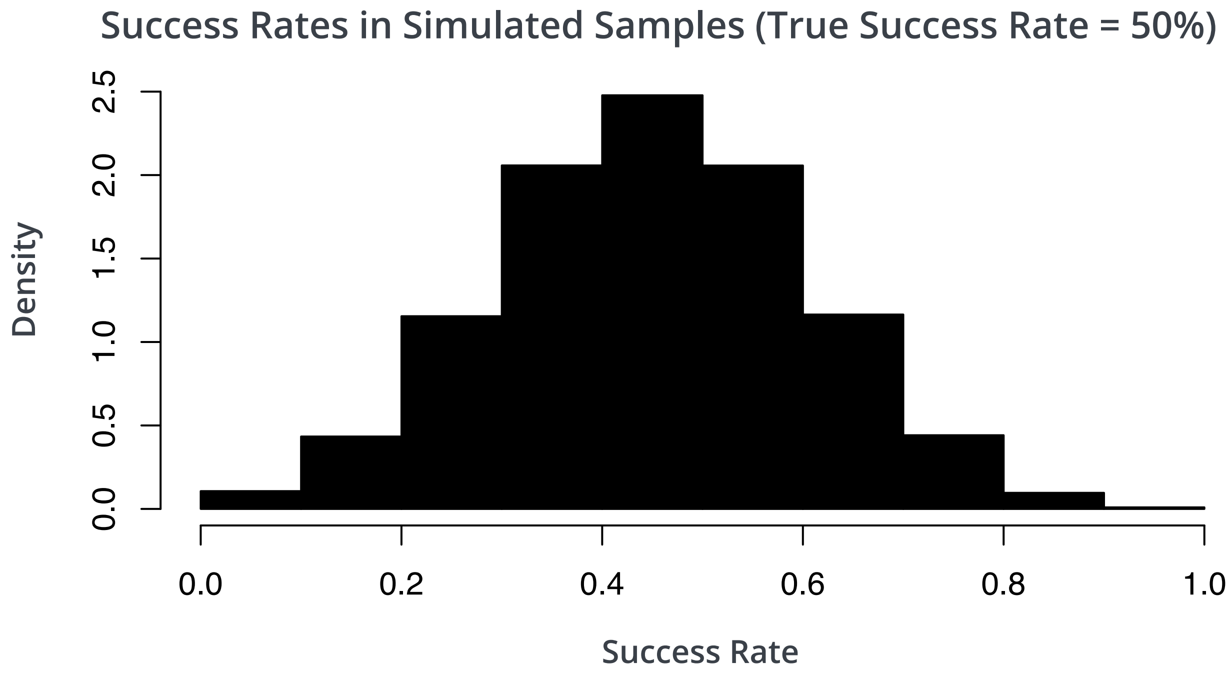
```
set.seed(1) # Set the seed for reproducibility

# Set up:
nsim = 100000 # Number of iterations
store_p = rep(0, nsim) # Vector in which to store sample statistics

# For loop to repeat Step 1 multiple times:
for (i in 1:nsim){
         outcome = c("Worked", "Did not Work")
    result = sample(outcome, 10, replace = TRUE, prob = c(0.5, 0.5))
    p_sim = mean(result == "Worked")
    store_p[i] = p_sim
}
```

Step 3: Use a histogram to see how the sample success rate of the drug varied from sample to sample, even when you explicitly set its true success rate in the population at 50%. When you draw a null distribution with your data, make sure to check that it is centered around the value specified by your null hypothesis.

```
hist(store_p, breaks = seq(0, 1, 0.1), freq = FALSE, col = "black",
     main = "Success Rates in Simulated Samples (True Success Rate = 50%)",
     xlab = "Success Rate")
```



Interpreting a Simulation

You can examine the histogram you made with the simulation to help you understand the null distribution of the sample statistic.

The **mean** of this histogram tells you what you should expect to see if there is nothing interesting going on in the population. As expected, since we set a 50% population success rate in our simulation, we see that sample success rates across many random samples are concentrated around 50%.

The **standard deviation** of this histogram gives you a sense of the variability you should expect to see in the sample statistic. This tells you how much the sample statistic (success rate) would vary from sample to sample on average if the drug only had a 50% chance of working on each patient. When this standard deviation is small, a large value of the sample statistic would provide strong evidence that what you find in your sample is a real signal and not just due to randomness. However, if this standard deviation is large, even if you see a large sample statistic, take it with a grain of salt before generalizing your finding to the population.