Simple Monte Carlo

Patrick Wilson

4/17/2019

Simple Example

We start off with a very simple test. Behold the t-test! You have probably heard of it[1].

$$t = \frac{\overline{X} - \mu_0}{\hat{\sigma}/\sqrt{n}}$$

Here we will power a simple hypothesis test utilizing the Design, Simulate, and Record framework.

Design Your Model

You will likely build a function for this which can facilitate multiple scenarios let's keep it simple. For complex models you will have more parameters.

```
n <- 25      # sample size
mu <- 7.5   # true mean
sigma <- 15  # true SD
mu0 <- 0    # mean under the null hypothesis

reps <- 10000  # number of simulations
```

Simulate Your Model and Record Your Result

There is where the simulation begins. It is a simple loop that creates a distribution of hypothetical differences from a normal model. Then for each draw from the distribution, test to see if you can detect a difference from the null given your signficance level.

```
set.seed(04192019)

pvalues <- numeric(reps)
for (i in 1:reps) {
  x <- rnorm(n, mu, sigma)
  #Run your test for each repetition and record your p-value
  pvalues[i] <- t.test(x, mu = mu0)$p.value
}
```

The proportion of statistically significant effects is your *power*

```
mean(pvalues < 0.05)
```

```
## [1] 0.6623
```

[1] As an aside I am simulating a normal model and testing with a t-test. We generally want to match the design with the analyses but sometimes this is not possible or desirable. A good example would be a cox model, with any design choice you would need to simulate a baseline hazard but you may still want the robustness of a cox model. For a sample size of 25 there won't be a big differences between the t and normal. With larger samples sizes the difference wouldn't be noticeable