The Power of Simulations

Ellie Sun

11/27/2017

Introduction to Simulations

Statistics is a topic that is seen in a plethora of fields because being able to collect, organize, analayze and present data is an ability that is required in nearly all professions. It is also vital that people are able to make decisions based off of that data. Sometimes it is also imporatant to be able to predict what till happen, or at least present the possible outcomes. That is where simulations come in. According to Hallgren, a statistics professor at The University of New Mexico “Simulation studies allow researchers to answer specific questions about data analysis, statistical power, and best-practices for obtaining accurate results in empirical research.” Simulations provide more insight and help analyze results. This posts dives into the applicability of generating simulations in R and how they are very useful in solving problems.

Random Number Simulation

One convient built in function of R is sample.int(). It allows you to generate n random numbers between 1 and n. It is useful in generating an order or assingning people a random number. In the following example, I set n=10. If you try this code your result should be different because R is generating the list of random integers. Another way to generate random numbers is runif(), also showed below. It takes the number of desired outputs, a min and a max. It is also determined by the value inputted into set.seed().

```
#Example Using sample.int
sample.int(10)
```

```
##  [1]  9  6  4  3  1  7  2  5  8 10
```

```
#runif
runif(20, min = 1, max = 5)
```

```
##  [1] 3.242867 2.853353 3.822081 2.618098 2.888319 4.485196 2.744132
##  [8] 2.010244 2.177757 2.206537 1.630664 4.743495 4.484637 3.601495
## [15] 2.100702 2.004302 2.083890 4.655864 2.928797 4.772954
```

```
set.seed(3)
runif(20, min = 1, max = 5)
```

```
##  [1] 1.672166 4.230066 2.539769 2.310937 3.408403 3.417576 1.498534
##  [8] 2.178404 3.310440 3.523917 3.048064 3.020096 3.136141 3.228998
## [15] 4.471678 4.318835 1.445797 3.814753 4.589953 2.118930
```

Useful Statistical Tests in R

Generating data to test is one part of running a simulation. To continue analyzing data, it is useful to use t.test() and cor.test() to look at data and compare data. Below is a t.test on the vector x that is generated with rnorm. The cor.test was done on the data set faithful that is found in R.

```
set.seed(3)
x=rnorm(25) #produces a N(0,1) sample of size 25
t.test(x)
```

```
##
##  One Sample t-test
##
## data:  x
## t = -1.8654, df = 24, p-value = 0.07439
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.60826489  0.03072478
## sample estimates:
##  mean of x
## -0.2887701
```

```
attach(faithful) #resident dataset
cor.test(faithful[,1],faithful[,2])
```

```
##
##  Pearson's product-moment correlation
##
## data:  faithful[, 1] and faithful[, 2]
## t = 34.089, df = 270, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8756964 0.9210652
## sample estimates:
##       cor
## 0.9008112
```

Population Simulation

Simulations often try and project how things will grow or change. A major thing to look at is popultion. Below is a simulation that Columbia has down regarding birth and gender. The generated a sample of 400 to see how many femails would be born and also considered the posibilities of twins. They then ran that simulation 1000 times and ploted the results in the histogram to show the potential distribution of girls out of 400 births.

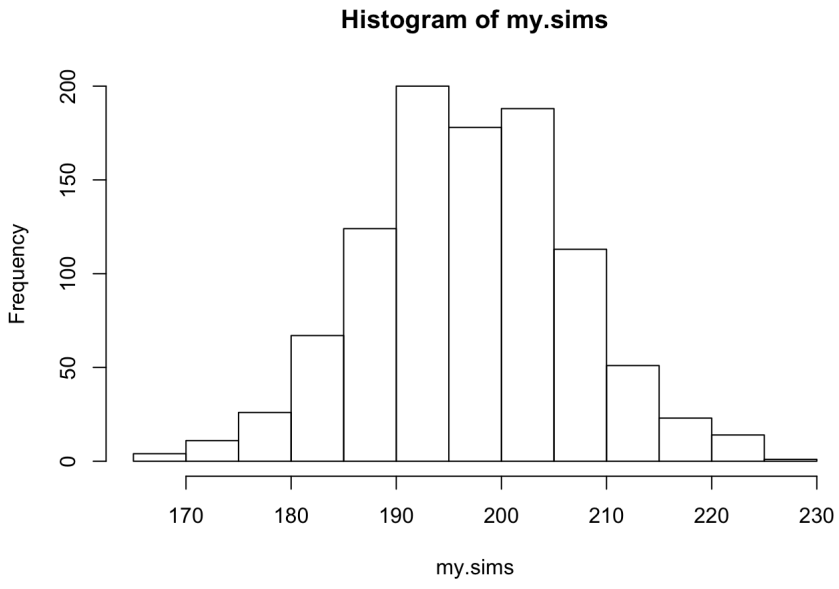
```
birth.type <- sample (c("fraternal twin","identical twin","single birth"),
size=400, replace=TRUE, prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
girls <- rep (NA, 400)
for (i in 1:400){
if (birth.type[i]=="single birth"){
girls[i] <- rbinom (1, 1, .488)}
else if (birth.type[i]=="identical twin"){
girls[i] <- 2*rbinom (1, 1, .495)}
else if (birth.type[i]=="fraternal twin"){
girls[i] <- rbinom (1, 2, .495)}
}

n.girls <- sum (girls)
girl.sim<-function(x){
birth.type <- sample (c("fraternal twin","identical twin","single birth"),
size=x, replace=TRUE, prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
girls <- ifelse (birth.type=="single birth", rbinom (400, 1, .488),
ifelse ( birth.type=="identical twins", 2*rbinom (400, 1, .495),
rbinom (400, 2, .495)))
return(sum(girls))
}

girl.sim(400)
```

```
## [1] 195
```

```
my.sims<-replicate(1000, girl.sim(400))
hist(my.sims)
```



The Monte Carlo Simulation

The Monte Carlo Simulation is a computer generated simulation that involves random sampling from probability distributions under a variation of conditions. To generate a Monte Carlo simulation in R, you use the function set.seed to generate reproducible random numbers as long as the set.seed amount is constant. The example below is courtesy of Stanford.

```r
#Suppose we rolled two fair dice. What is the probability that their sum is at least 7? We will approach this by s
imulating many throws of two fair dice, and then computing the fraction of those trials whose sum is at least 7. I
t will be convenient to write a function that simulates the trials and returns TRUE if the sum is at least 7 (we c
all this an event), and FALSE otherwise.

isEvent = function(numDice, numSides, targetValue, numTrials){
  apply(matrix(sample(1:numSides, numDice*numTrials, replace=TRUE), nrow=numDice), 2, sum) >= targetValue
}

set.seed(0)
#try 5 trials
outcomes = isEvent(2, 6, 7, 5)
mean(outcomes)
```

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

```r
#10,000 trials
set.seed(0)
outcomes = isEvent(2, 6, 7, 10000)
mean(outcomes)
```

```
## [1] 0.5843
```

Conclusion

R is a great tool that has built in statistical functions that allow people to easily generate simulations. It can generate random numbers, provide probability distributions and run tests on data to provide all relevant statistical data. There are many other packages that also help explore more simulations and are often designed for specific fields, it could be useful in business, engineering or medicine.

References

http://www.tqmp.org/RegularArticles/vol09-2/p043/p043.pdf

https://web.stanford.edu/class/bios221/labs/simulation/Lab_3_simulation.html

http://www.columbia.edu/~cjd11/charles_dimaggio/DIRE/resources/R/simRreg.pdf

http://had.co.nz/stat480/lectures/15-simulation.pdf

http://www.stat.ufl.edu/archived/casella/ShortCourse/MCMC-UseR.pdf

http://www4.stat.ncsu.edu/~davidian/st810a/simulation_handout.pdf

http://www.me.utexas.edu/~me302/classnotes/MODELING/sld003.htm