Random Number Generator

Marc Los Huertos

Thursday, October 22, 2015

# Introduction

This is an R Markdown document that you can use to generate random numbers. I suggest you generate these numbers before you go the field and print them out.

We will determine the locations for soil sampling using a random design format. Determine the approximate length and width in meters of the grassy area at your park location. Using a random number generator (e.g., www.random.org), find one number to serve as your x coordinator and a second number to serve as your y coordinate. Repeat this four more times. Select a corner of the grassy area to serve as your 0,0 coordinate, then using the randomly generated x,y pairs, determine your five locations for sampling soil.

Note: Typically you might get random numbers between 1 and 100. But you can scale them to the size of the park. Pace out (or use a meter tape) the x and y coordinates for the part and you can use these max values to scale the random numbers to ensure unbiased sampling. If the sample lands on a sidewalk or sand or building, throw the number out. I usually generate double the random values to avoid having to find new numbers on the fly.

# Process

## Decide What Distribution to Use?

First, we need to decide how we want our sampling distributed. In most cases, you want to have an equal probability that any sample would be sampled -- thus we should use a uniform distribution. In other words, we want random numbers that fall equalling within a range as opposed to numbers that are drawn from a normal distribution that will generate more values near the mean. In R, we can use the runif() function. Look up the help menu for runif().

Let's generate 5 numbers. Notice that the default range is used, confirm the range you received with the range that R used.

N = 5  
runif(n=N)

## [1] 0.09956852 0.86098867 0.65719141 0.07373322 0.06701554

## Decide the range of values.

To make it simple, we will keep these between 0 and 100. So, we will create a min and max between 0 and 100, respectively.

runif(n=N, min=0, max=100)

## [1] 68.83187 38.86047 84.98572 76.87289 16.20513

If you know the size of the field before you sampling you can scale the runif() to those numbers. Let's say the sampling location is 40 meters in the N/S direction (y) and 340 meters in the E/W direction (x). Let's create a matrix with scaled, random x and y coordinates.

x = 340 and y = 40, but these can change!

x = runif(n=N, min=0, max=340)  
y = runif(n=N, min=0, max=40)  
column = c("x", "y")  
matrix(c(x, y), nrow=N, byrow=F, dimnames=list(1:N, column))

## x y  
## 1 81.26291 12.40019  
## 2 210.53232 11.79843  
## 3 212.07543 34.84599  
## 4 309.64821 7.98164  
## 5 272.57341 12.92031

The values have way more signficant figures that we need! Let's round to the nearest meter.

x = round(runif(n=N, min=0, max=340), 0)  
y = round(runif(n=N, min=0, max=40), 0)  
column = c("x", "y")  
matrix(c(x, y), nrow=N, byrow=F, dimnames=list(1:N, column))

## x y  
## 1 275 10  
## 2 43 2  
## 3 331 13  
## 4 122 15  
## 5 244 17

## Generating enough points

If the sample lands on a sidewalk or sand or building, throw the number out, so I suggest you create way more points than you need. If you need 20, make 100.

## Scaling on the Fly

Since we don't have the opportunity to visit these parks beforehand, we will have to scale the points on the fly. For example, say the park is 40 meters in the North (y-axis) direction and 340 paces in the East (x-axis) direction.

N=5  
x = round(runif(n=N, min=0, max=1), 3)  
y = round(runif(n=N, min=0, max=1), 3)  
column = c("x", "y")  
matrix(c(x, y), nrow=N, byrow=F, dimnames=list(1:N, column))

## x y  
## 1 0.501 0.026  
## 2 0.681 0.525  
## 3 0.222 0.910  
## 4 0.633 0.390  
## 5 0.918 0.142

And you get a random coordinates of 0.288 and 0.788. Then our points will be .788 x 340 (268 meters) to the east and .288\*40 = 11.5 to the north. I decided to keep one more significant figure for the north because the values are constrained by the long-narrow shape of this park. NOTE: To do this in the field, you want to bring your calculator or the computer.

Finally, I create let's create a bunch of values that you can use in the field today:

N=5  
x = round(runif(n=N, min=0, max=1), 3)  
y = round(runif(n=N, min=0, max=1), 3)  
column = c("x", "y")  
sampling = matrix(c(x, y), nrow=N, byrow=F, dimnames=list(1:N, column))  
sampling[1:5,]

## x y  
## 1 0.255 0.430  
## 2 0.629 0.477  
## 3 0.965 0.799  
## 4 0.492 0.494  
## 5 0.282 0.599

Note: you may want more. It's up to you. Remember you have a few parks to sample, so you might want to make sure you have lots of values.

## Ordering the sampling

When you have dozens of sampling locations it often makes sense to order the sampling, so you don't criss cross a sensitive habbitat. Of course, this may also save time. But you want to think about, "does this create a sampling bias"? If so, you would avoid this step.

sampling.ord = sampling[order(sampling[,1]),]; sampling.ord

## x y  
## 1 0.255 0.430  
## 5 0.282 0.599  
## 4 0.492 0.494  
## 2 0.629 0.477  
## 3 0.965 0.799