Simulating Data

Preamble

R is great, but the #rstats community is freaking **awesome**

- This tutorial comes from Ariel Muldoon (twitter: @aosmith16)
- 100% of the credit goes to Ariel!
- The original tutorial is here and she has other really fantastic posts, too!

Why Simulate?

Because collecting data is ***really, really hard***

- Simulating data is useful when you're trying to prove something theoretically.
- Example via twitter thread by Eiko Fried
    - does adding noise to a variable change the underlying correlation with another variable (e.g., by adding noise you are basically saying that variable 1 is not super precise or adding measurement error)?
    - create your own data or correlation matrix and do it yourself!
    - no need to collect data from participants
- Goal of today is to take you through some R functions that are useful for simulating data

Generating random numbers

One way to generate numeric data in `R` is to pull random numbers from some defined distribution. This can be done via functions for generating random deviates. These functions start with `r` (for *random*).

What distributions do you know?

- Normal distribution (`rnorm()`)
- Uniform distribution? (`runif()`)

The rnorm() function

- The `rnorm()` function pulls values from a normal distribution
- Sometimes it's because you think the data *should* be normally distributed
- Other times it's because you just need some random numbers and you don't care very much about what they are or if they follow any particular distribution

```
rnorm(n, mean = 0, sd = 1)
```

- The `n` argument is the number of values to generate.
- The `mean` and `sd` are the parameters of the distribution. You can see the default values are set to specific values. Note that `sd` is the *standard deviation*, not the variance.

Setting the random seed

- If you went into `R` right now and ran the code `rnorm(n = 5, mean = 0, sd = 1)`...then you ran that code *again*, you would get a different set of numbers as your output.
- This is really bad for reproducibility! How can someone replicate your simulation study if the numbers they are getting are totally different?
- To get reproducibile random numbers, you need to **set the seed** via the `set.seed()` function.
- To my knowledge, it does not matter what number you put in the `set.seed()` function -- just as long as it is set to something. For instance, this class is Psych 4175, so maybe use `set.seed(4175)`. I often do `set.seed(1234)` because I'm such a creative person (lol)
- If you set the seed *before* running `rnorm()`, and then run the code twice, you should get the same set of numbers.

Change parameters in `rnorm()`

- For getting a quick set of numbers it's easy to use the default parameter values in `rnorm()` -- this is the standard normal distribution (mean = 0, sd = 1)
- You can change the parameter values to pull from a different normal distributions
- IQs have a mean of 100 and standard deviation of 15. If you want 20 simulated IQ scores, you would run `rnorm(n = 20, mean = 100, sd = 15)`.