*The Monte Carlo code explained, in greater detail*

Preliminary work:

* First, we ensure that there is no systematic correlation between age and analytical error in our dataset; a simple plot of age vs. error clearly demonstrates that there is not (plot not shown; see Table X for data).
  + We plot our measured ages as a histogram, and determine that a normal, Gaussian distribution is a fair approximation of our data. We confirm our assertion by making a QQ plot in MATLAB.
  + We then calculate the mean, standard deviation and standard error for our ages.
* We plot our analytical errors as a histogram, and determine that a gamma distribution best describes the data. We confirm our assertion with a gamma QQ plot.
  + We then use MATLAB to calculate the shape and scale factors (which describe the gamma distribution) of our analytical errors.

The code, explained:

* The first step in our MATLAB code is to designate a mean age by taking our true population mean (12.9 million years) and then randomly selecting an age from a normal distribution with mean of zero and a standard deviation equal to the standard error of our true population (0.07 million years).
  + The first for-loop in our code directs MATLAB to:
    - Select a random age from a normal distribution about the selected mean (see step 1; true mean with standard error accounted for) and a standard deviation of 0.66 (our standard deviation for sample ISS).
    - Select a random analytical error from a gamma distribution with the shape and scale factors determined from the actual dataset;
    - Add the randomly selected age to an error randomly selected from a normal distribution with a mean of zero and a standard deviation equal to the error selected from the standard deviation (to account for the error to being positive or negative)
  + In our second for-loop, we instruct MATLAB to tally up each of the times the randomly selected age+error is greater than some maximum age-point that we specify (the “exceedence” tally)
  + We repeat all steps up to this point N number of times, equal to the number of analyses in our real-world dataset (83 times if only considering ISS ages, or 247 considering the our entire Breiduvik zircon population).
    - If, in each of N runs, the randomly selected age+error never exceeds the maximum age cut that we specify, we start a second tally (the “non-exceedence” tally)
  + We call this collection of N random selections and subsequent tallies a “set;” We repeat a set NN times (e.g., 10, 100, 1000 times), and call that a “cycle”
    - After completing a cycle, we can repeat the process (optional) but consider a minimum age cut instead of a maximum age cut, to ensure symmetry in our random distribution
  + At the end of each cycle, we take the count from the “non-exceedence” tally, and divide it by the number of sets (NN) that we ran. This gives us a “non-exceedence probability”
    - This non-exceedence probability allows us to make statements like “in a Monte Carlo simulation based off of a sample size of 83, with a maximum age cut off of the mean + 2.64 million years, we found no ages older than the cut-off 750 times out of 1000. That is to say, there is a 75% chance that we would NOT find a zircon with an age greater than the mean + 2.64 million years, and 75% chance that Breiduvik does NOT have an age range > 5.28 million years”