Function Documentation

This file will list out the functions and give a description of them. This can be used as a refernce file so that you don't have to lumber through Julia.R if you have any questions about whcich function you want to use.

Some notes on notation:

1. z will be used to represent a complex value. In much of this code, if you apply a function to a numeric or integer, it will be coerced into a complex value in the process.
2. $f(z)$ is the function we will use for the Julia set.
3. $f_2(z)$ will be shorthand for $f \circ f(z)$ and inductively $f_n(z) = f_{n-1} \circ f(z)$.

About Julia Sets

For many of our examples, we'll look be using the function $f(z) = z^2 - 0.4 + 0.6i$. In reality, we can take f to be any rational function (quotient of two polynomials). As we apply f over and over again, i.e., create a sequence $\{f_n(z)\}$, either the values will remain bounded or will tend towards infinity, $|f_n(z)| \to \infty$.

We'll use the notation J_f to refer to the Julia set associated with f. A complex number, z, is said to belong to J_f if $|f_n(z)|$ remains bounded. Often times, J_f is referred to as the filled Julia set while the "true" Julia set is the boundary of J_f.

We aren't going to make much of a big fuss about the distiction between the fiilled and true Julia sets. Our plots will often color code based on how many iterations it takes for $|f_n(z)|$ to become very large (determined by the bound parameter of our R function). In this case, the color black will be used for the filled Julia set. There are some cases where the filled Julia set is color coded while the unbounded points are set to a constant color.

For each plot, I'll try to include information about the color coding. If it's unclear (or I haven't documented it yet), feel free to ask and I'll put it at the top of my priority list and get back to you as quickly as possible.

Most Useful Functions

This section will contain the functions that you see the most.

- thefunction:
  - Takes in a value z (numerics or ints coerced to complex) and outputs f(z) = zˆ2 - 0.4 +0.6i.
  - If you want to change the function, you'll need to copy and manually edit the code of "thefunction". Every .R script in this repo should start with "thefunction" being called outside of Julia.R for added specificty.
  - If there is a way around manually editing the code, please let me know. I would really like to have "thefunction" to be a function of z and "f(z)".
- iterboundnthird
  - This is a more advanced than the iterboundn and iterboundnsecond (see next section).
  - Output is the first N value for which there exists n such that $|f_n(z) - f_N(z)| < \varepsilon$ holds, for some $n < N$.
  - Otherwise, output is iterations plus 100 if z is not in Julia set and output is iterations + 1 if z is in the Julia set.
- iterunbounded
  - Returns first n value for which $|f_n(z)|$ is larger than the bound.
  - If $|f_n(z)|$ remains smaller than bound, then iterations + 100 is returned to say that z belongs to the Julia set.
- xseq
  - Creates a sequence to be used for the real values of each pixel.

  - It just call seq so it is pretty unnecesary.
- yseq
  - Same as xseq but for the imaginary values.
  - Also unnecessary and will probably be moved to the next section at somepoint.
- fill
  - Created to replace initmatrix (see next section) since initmatrix was a laborious task and can't be easily parallelized.
  - First, one must create the matrix of size length(xseq) by length(yseq) and fill it with the integer values from 1 to length(xseq)*length(yseq).
  - It is suggested that the matrix you create gets converted to complex values before running fill, especially with large matrices.
  - fill uses the entry value to figure out the position in the matrix (i$^{\text{th}}$ row j$^{\text{th}}$ column) and outputs z = x[i] + y[j]i.
- filliterunb
  - Combines fill and iterunbounded.
  - Designed to take in a numeric matrix and output a numeric matrix with the idea that numeric values require less RAM than complex values.
  - This will hopefully allow us to work with even larger matrices.
- filliterbound
  - Combines fill and iterboundnthird.
  - Designed for the same putpose as filliterunb allowing us to work around the complex values.

Less Used Functions

- iterating
  - Takes in a complex value and the number of iterations you want to apply.

  - Outputs the value returned after applting thefunction that many times.
- in.Julia2
  - Determines if z belongs to the Julia set.
  - Applies thefunction a fixed number of times. If the output is too large (determined by bound), then FALSE is returned.
  - Otherwise, TRUE is returned because z likely belongs to Julia set.
- iteratingbound
  - Same as in.Julia2 but was designed to be quicker.
- iterboundn
  - Like in.Julia2 and iteratingbound, the main purpose is to determine if z belongs to the Julia set.
  - This also tests for convergence and outputs n when z and f(z) are very close ($|f(z) - z| < \varepsilon$ where ε can be set by you).
  - If it is determined that z is outside the Julia set, then iterations plus 100 is returned.
  - If z is determined to belong to the Julia set, but $|f(z) - z|$ never gets small enough, then iterations + 1 is returned.
- iterboundnsecond
  - As the name implies, a more advanced version of iterboundn.
  - Only difference is that if iterations/2 < n <= iterations and $|f_n(z) + 0.4 - 0.6i| < \varepsilon$, then n is returned
- initmatrix
  - Creates a matrix full of the complex values in the viewscreen that we are interested in.
  - Can run very slowly and replaced by more advanced functions, whence it is in this section.
  - I would not recommend that this function be used.