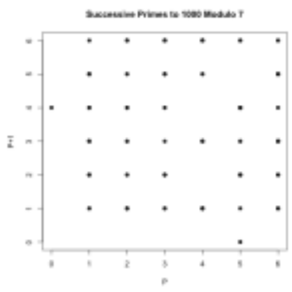
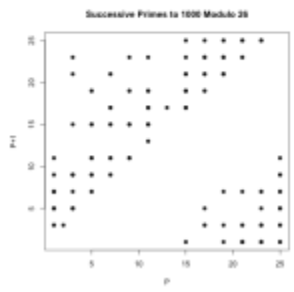
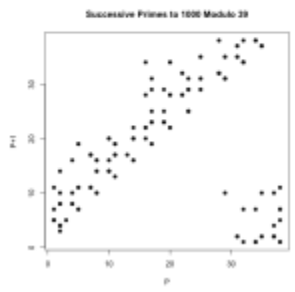
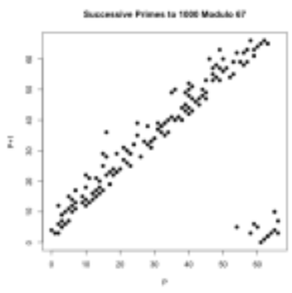
Plotting Successive Primes Modulo A Number

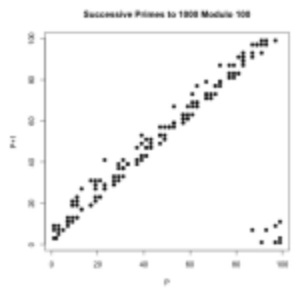
2015-11-30 BY GENE











tl;dr: https://github.com/ology/Math/blob/master/primes/sequence-pairs

File this under [Recreational mathematics](#)…

I have this passionate curiosity for seeing the primes "folded in on themselves" –

constrained to a phase space. I also have a burning desire to visualize successive events. I suppose that I am searching for “strange attractors”, as with the famous water drip experiment.

I wanted to see the prime numbers this way, but they just ascend rapidly. Boring!

If they are “reduced” by dividing, modulo some number, they will be bounded (although still growing). Some examples will illustrate what I mean: 1 modulo 7 is 1. 2 modulo 7 is 2… 6 modulo 7 is 6. 7 modulo 7 is 0. If you do this to the prime number sequence (2, 3, 5, 7, 11, 13…), for 7 you get: 2, 3, 5, 0, 4, 6… For my visualization of successive primes, the points would therefore be (2,3), (3,5), (5,0), (0,4), (4,6)… – Never above the modulo, which is 7 in this case.

In order to generate the numbers and points, and visualize everything, I built the perl code above. An explanation follows:

I get the prime number sequence with the excellent Math::Prime::XS module. Then from user provided **start** and **stop** values, I iterate, taking each value as the modulo. Next, I get the new sequence of the modified primes (with the modulo):

```
for my $mod ( $start .. $stop ) {
    my @sequence = map { $_ % $mod } @primes;
    ...
```

For the animation below, **start** was 2 and **stop** was 100. The number of primes below, my chosen number, 1000 is 97. So there are 96 points. Most overlap. This code creates x and y lists:

```
for my $p ( 0 .. @sequence - 1 ) {
    if ( defined $last ) {
        #print "[$sequence[$last],$sequence[$p]]\n";
        push @x, $sequence[$last];
        push @y, $sequence[$p];
        $last = $p;
    }
    else {
        $last = $p;
        next;
    }
}
```

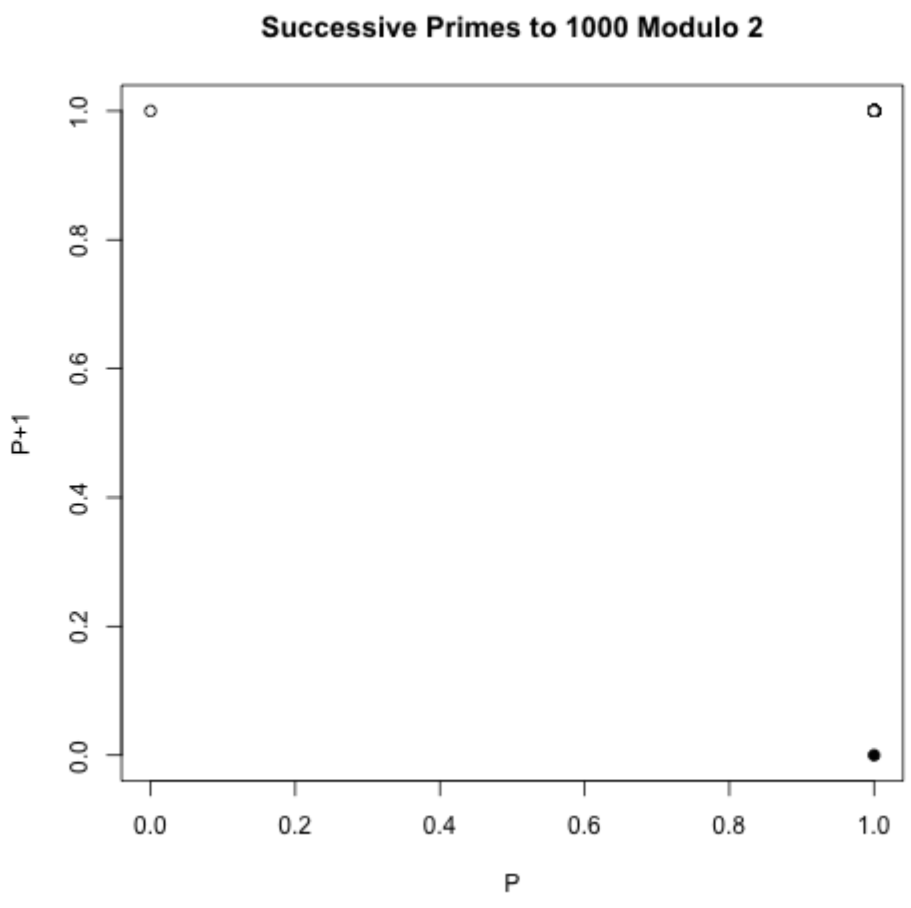
Next I scatter plot the points by interfacing with the venerable, R environment:

```
$R->run( "png(file='$file')" );
$R->run( 'x <- c(' . join( ',', @x ) . ')' );
$R->run( 'y <- c(' . join( ',', @y ) . ')' );
$R->run( 'z <- c(' . join( ',', map { $prop{$_} } @y ) . ')' );
$R->run( "plot(x, y, pch=21, bg=gray(z), xlab='P', ylab='P+1', main='Succes
$R->run( 'dev.off()' );
```

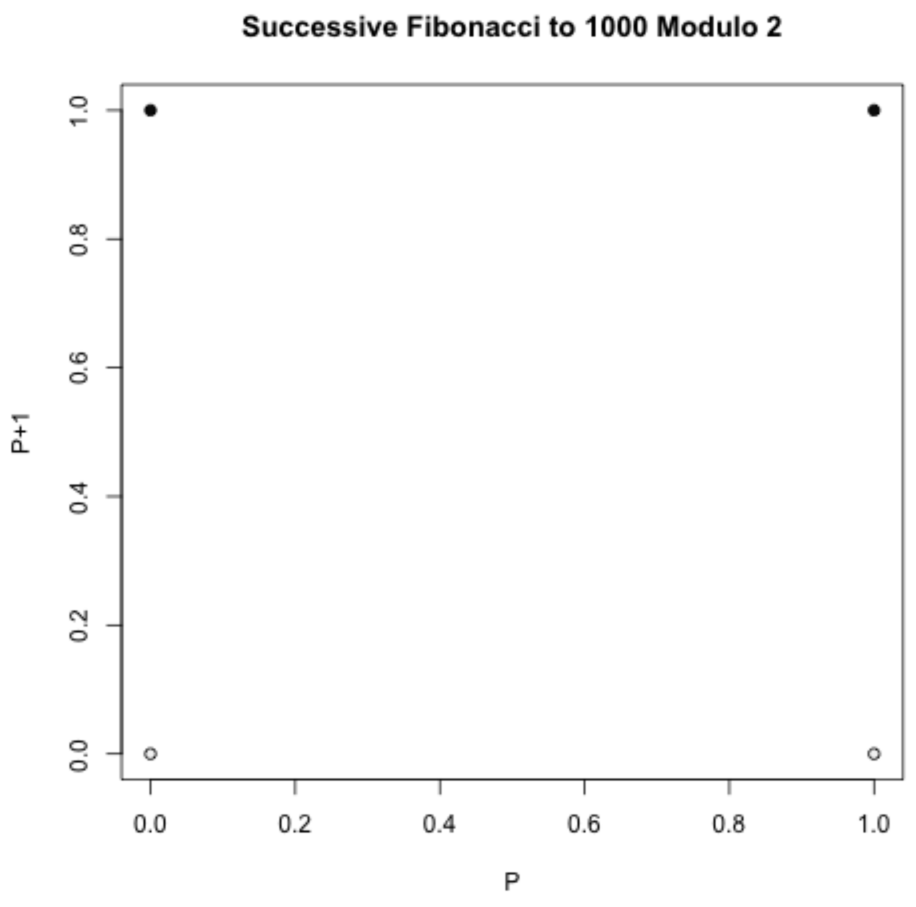
What is z, is an exercise left to the reader…

This runs for each sequence of "modulo primes." For the animation, I chose to show 100 frames. I used the following ImageMagick command to stitch together the frames into an animated GIF:

```
convert -delay 40 -loop 0 sequence-pairs-mod-*.png sequence-pairs-animaion.
```



To show a contrasting plot, here is the Fibonacci sequence plotted in the same way (i.e. by successive modulo):



FILED UNDER: MATHEMATICS, SOFTWARE
TAGGED WITH: PERL, PRIME NUMBERS, R, VISUALIZATION

Epistemologist-at-large

^ Top