# Prime Number Digit Frequencies

2019-05-05 BY GENE

Today I became curious about the relative frequencies of the digits that appear in prime numbers (in base 10).

Naturally, I wrote a little program to show the answer (digit-freq2):

```
#!/usr/bin/env perl
use strict;
use warnings;

use List::Util qw(sum0);
use Math::BigRat;
use Math::Prime::XS qw(primes);

my $min = shift // 10_000;
my $max = shift || 10_000_000;

my @primes = primes( $min, $max );
warn " $min to $max => ",scalar(@primes),"\n";

# Count the number of times a digit is seen
my %digits;
for my $prime ( @primes ) {
```

```
    my @parts = split //, $prime;
    $digits{$_}++ for @parts;
}

# Find the total of all digits seen
my $total = sum0 values %digits;

# Compute the proportional frequencies
for my $digit ( keys %digits ) {
    $digits{$digit} = Math::BigRat->new(
        Math::BigInt->new($digits{$digit}),
        Math::BigInt->new($total)
    );
}

# Output the frequencies in sorted order
for my $digit ( sort { $digits{$a} <=> $digits{$b} || $a <=> $b } keys %dig
#    print "$digit: ", $digits{$digit}->as_float, "\n";
    printf "%d: %.4f\n", $digit, $digits{$digit}->as_float;
}
```

For primes up to 10 (4 of them) we get this:

```
2: 0.2500
3: 0.2500
5: 0.2500
7: 0.2500
```

For primes up to 100 (25) we get:

```
6: 0.0435
8: 0.0435
2: 0.0652
4: 0.0652
5: 0.0652
9: 0.1304
1: 0.1957
3: 0.1957
7: 0.1957
```

For primes up to 1,000 (168) we get:

```
0: 0.0316
8: 0.0632
2: 0.0674
5: 0.0695
6: 0.0695
4: 0.0716
9: 0.1411
3: 0.1579
1: 0.1642
7: 0.1642
```

At 10,000 (1,229) we get:

```
0: 0.0492
8: 0.0744
4: 0.0763
5: 0.0763
6: 0.0782
2: 0.0829
9: 0.1369
7: 0.1382
3: 0.1435
1: 0.1443
```

At 100,000 (9,592) we get:

```
0: 0.0586
8: 0.0793
6: 0.0804
4: 0.0811
5: 0.0820
2: 0.0839
9: 0.1317
7: 0.1326
3: 0.1339
1: 0.1365
```

At 1,000,000 (78,498) we get:

```
0: 0.0660
8: 0.0838
6: 0.0842
5: 0.0846
```

```
4: 0.0848
2: 0.0860
9: 0.1264
7: 0.1268
3: 0.1278
1: 0.1296
```

For an increasing number of primes, the order of increasing frequency *appears* to stabilize at 0, 8, 6, 5, 4, 2, 9, 7, 3, 1.

So for instance, at 10,000,000 (664,579) we get:

```
0: 0.0710
8: 0.0864
6: 0.0867
5: 0.0868
4: 0.0871
2: 0.0878
9: 0.1225
7: 0.1228
3: 0.1238
1: 0.1250
```

For primes *between* 1,000,000 and 1,000,000,000 we get the same frequency order.

Fascinating.

(This link seems to indicate otherwise, but I haven't wrapped my head around it yet…)

FILED UNDER: MATHEMATICS
TAGGED WITH: FREQUENCY, PERL, PRIME NUMBERS

^ Top