2014-01-10.sagews

January 13, 2014

Contents

0.1 Counting primes . . . 1
0.2 Theorems and Conjectures . . . 1
0.2.1 Example . . . 1
0.3 Many Related Questions . . . 2
0.4 Further accessible reading . . . 3

0.1 Counting primes

Prime counting function

$$\pi(x) = \#\{p : p \leq x \text{ is prime }\}$$

```
@interact
def f(t=10, PNT=False, Gauss=False):
    x = var('x')
    g = prime_pi.plot(0,t)
    if PNT:
        g += plot(x/(log(x)-1), (5,t), color='red')
    if Gauss:
        g += plot(Li, (2,t), color='darkgreen')
    show(g, gridlines=True, svg=True, frame=True, figsize=[10,3])
```

0.2 Theorems and Conjectures

The Prime Number Theorem: The functions $\pi(x)$ and $x/\log(x)$ are asymptotic to each other, i.e., the limit as $x \to \infty$ of their quotient is 1.

Conjecture (The Riemann Hypothesis): For every $x > 2.01$, we have

$$|\pi(x) - \mathrm{Li}(x)| \leq \sqrt{x}\log(x).$$

0.2.1 Example

```
1e3 == 1000 # scientific notation
```

```
True
```

```
%time prime_pi(1e10)
```

```
455052511

CPU time: 0.13 s, Wall time: 0.13 s
```

```
%time prime_pi(1e11)
```

```
4118054813

CPU time: 0.90 s, Wall time: 0.90 s
```

```
@interact
def f(x=100000):
    if x >= 1e12:
        print "This will probably take too long so refusing"
        return
    a = prime_pi(x)
    b = N(Li(x))
    c = N(x/(log(x)-1))
    md("""
\\begin{eqnarray}
  \\pi(x) &=& %s   \\\\
  \\text{Li}(x) &=& %s\\\\
  x/(\\log(x)-1) &=& %s\\\\
  \\pi(x) - \\text{Li}(x) &=& %s\\\\
  \\sqrt{x}\\log(x) &=& %s
\\end{eqnarray}
"""%(a, b, c, a-b, N(sqrt(x)*log(x)))  )
```

0.3 Many Related Questions

For example, you can ask:

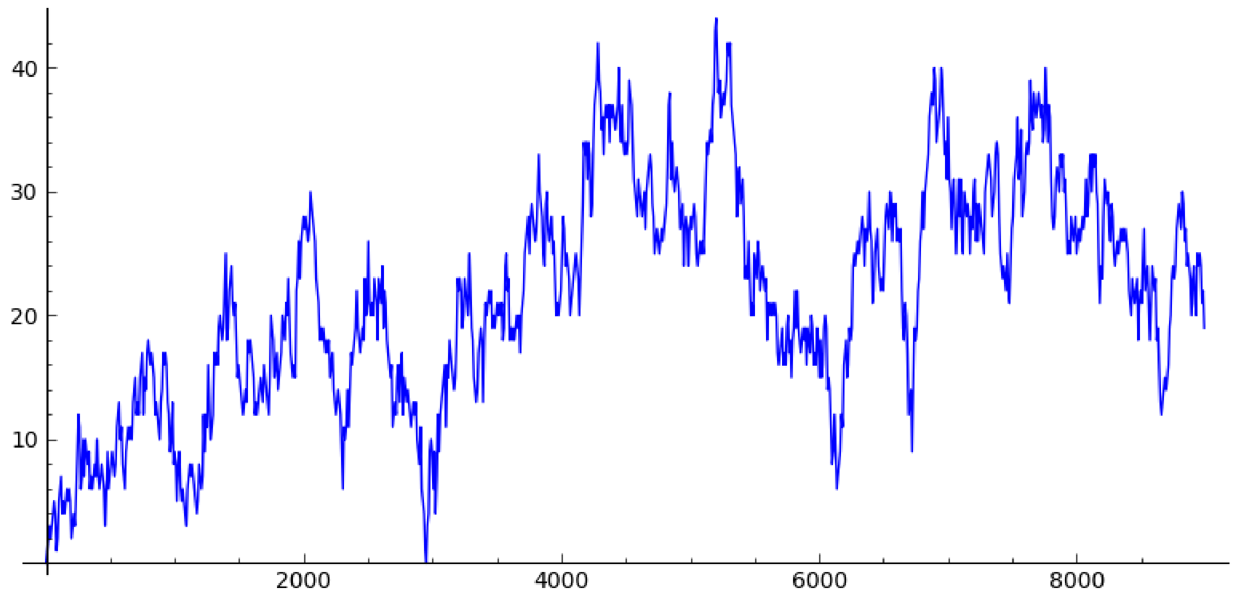
- are there infinitely many primes p that are congruent to 1 modulo 4 (if you divide p by 4 the remainder is 1)? YES
- you can race the primes that are 3 mod 4 versus the primes that are 1 mod 4
- you can replace 4 above by bigger numbers
- you can replace the primes with prime elements of the Gaussian integers
- and a million other things.

```
@interact
def f(bound=selector([10^i for i in [1..6]], buttons=True)):
    if bound > 1e7:
        print "way too big"
        return
    print "Up to %s"%bound
    p1 = len([p for p in prime_range(bound) if p%4 == 1])
```

```
    p3 = prime_pi(bound) - p1 - 1
    print "Primes p=1(mod 4): %s"%p1
    print "Primes p=3(mod 4): %s"%p3
```

```
v = [0]
for p in primes(3,100000):
    if p % 4 == 3:
        v.append(v[-1]+1)
    else:
        v.append(v[-1]-1)

finance.TimeSeries(v).plot()
```



0.4 Further accessible reading

See this book `http://wstein.org/rh`, which is currently free.