```
> with(numtheory):
> with(plots):
```

First 1000 primes:

```
> big := [seq(ithprime(i),i=1..1000)]:
```

First 10000 primes:

```
> bigger := [seq(ithprime(i),i=1..10000)]:
```

First 100000 primes:

```
> biggest := [seq(ithprime(i),i=1..100000)]:
```

Given list of primes and modulus, counts number of primes congruent to 1, 2, 3, ..., m (mod m):

```
> modstats := proc(primelist,m)
  local i,statlist,p;
  statlist := [seq(0,i=1..m)];
  for p in primelist do
      i := p mod m;
      if i=0 then i:=m end if;
      statlist[i] := statlist[i]+1;
  end do;
  statlist;
  end proc:
> currentm := 42;
```

$$currentm := 42 \quad (1)$$

```
> modstats(big,currentm);
```

[80, 1, 1, 0, 85, 0, 1, 0, 0, 0, 84, 0, 82, 0, 0, 0, 81, 0, 85, 0, 0, 0, 89, 0, 80, 0, 0, 0, 85, 0, 86, 0, 0, 0, 0, 0, 76, 0, 0, 0, 84, 0] (2)

```
> modstats(bigger,currentm);
```

[822, 1, 1, 0, 833, 0, 1, 0, 0, 0, 842, 0, 835, 0, 0, 0, 837, 0, 841, 0, 0, 0, 831, 0, 824, 0, 0, 0, 839, 0, 840, 0, 0, 0, 0, 0, 825, 0, 0, 0, 828, 0] (3)

```
> modstats(biggest,currentm);
```

[8328, 1, 1, 0, 8334, 0, 1, 0, 0, 0, 8338, 0, 8357, 0, 0, 0, 8340, 0, 8339, 0, 0, 0, 8348, 0, 8292, 0, 0, 0, 8349, 0, 8344, 0, 0, 0, 0, 0, 8300, 0, 0, 0, 8328, 0] (4)

As # primes -> infinity, expect proportion in each r.p. congruence class to approach 1/phi(m), so following should give roughly the right # of the first 100000 primes in each congruence class r.p. to *currentm*:

```
> evalf(100000/phi(currentm));
```

$$8333.333333 \quad (5)$$

1000th prime:

```
> p := big[1000];
```

$$p := 7919 \quad (6)$$

By PNT, expected proportion of primes at most that size:

```
> evalf(1/ln(p));
```

$$0.1113955384 \quad (7)$$

Actual proportion of primes at most that size:

```
> evalf(1000/p);
```

$$0.1262785705 \quad (8)$$

Ratio:

```
> evalf((1000/p)/(1/ln(p)));
```

$$1.133605280 \quad (9)$$

10000th prime:

```
> p := bigger[10000];
```

$$p := 104729 \qquad (10)$$

By PNT, expected proportion of primes at most that size:

```
> evalf(1/ln(p));
```

$$0.08651169111 \qquad (11)$$

Actual proportion of primes at most that size:

```
> evalf(10000/p);
```

$$0.09548453628 \qquad (12)$$

Ratio:

```
> evalf((10000/p)/(1/ln(p)));
```

$$1.103718296 \qquad (13)$$

100000th prime:

```
> p := biggest[100000];
```

$$p := 1299709 \qquad (14)$$

By PNT, expected proportion of primes at most that size:

```
> evalf(1/ln(p));
```

$$0.07103457839 \qquad (15)$$

Actual proportion of primes at most that size:

```
> evalf(100000/p);
```

$$0.07694029971 \qquad (16)$$

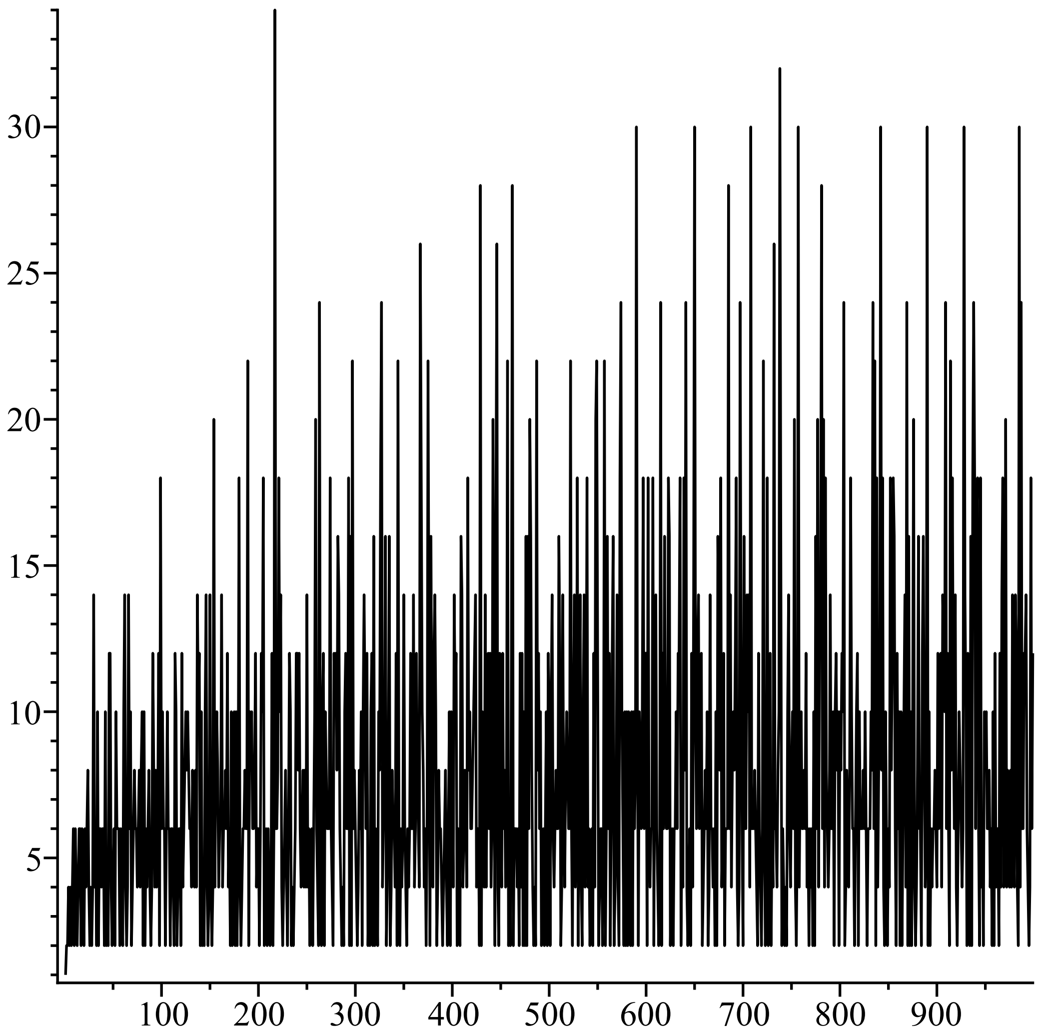
Ratio:

```
> evalf((100000/p)/(1/ln(p)));
```

$$1.083138683 \qquad (17)$$

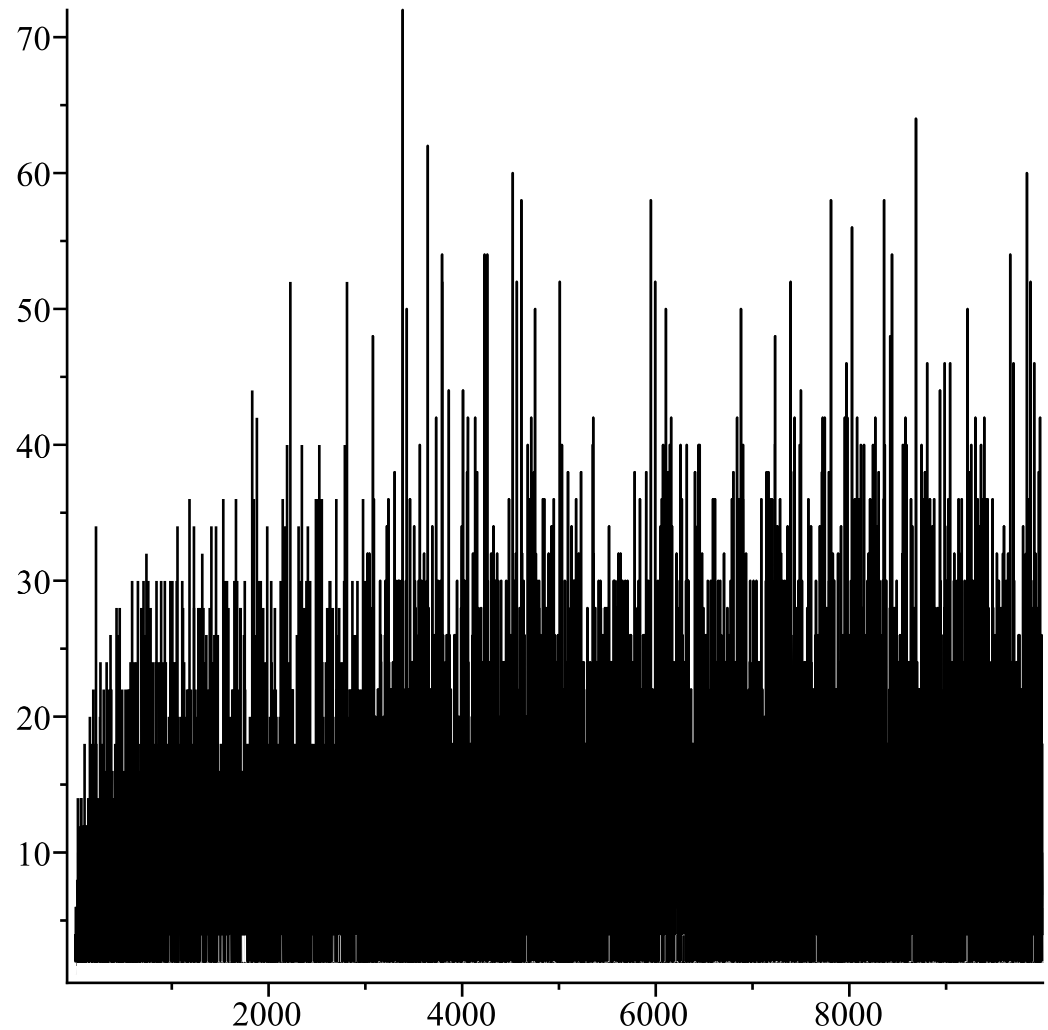
First 999 prime gaps:

```
> listplot ([seq(big[n+1]-big[n],n=1..999)]);
```



First 9999 prime gaps:

```
> listplot ([seq(bigger[n+1]-bigger[n],n=1..9999)]);
```


70
60
50
40
30
20
10
2000
4000
6000
8000