Information Science 19 Dec. 2016

No.11. Monte Carlo Simulation

ID 580012-F Name Alex Taniguchi

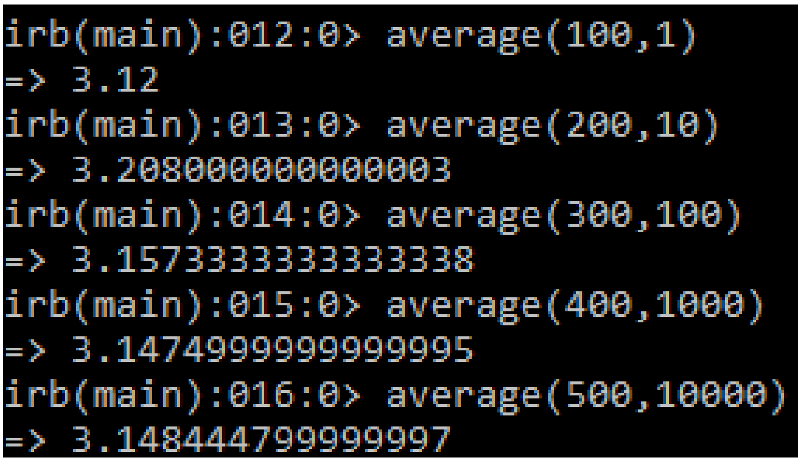
（１）Make a Ruby function average(t, n) that computes the average when performing the function montecalro(n) t times. Observe the performance of the Monte-Carlo Method taking a large t. Write anything you observed below.

```
def montecarlo(n)
    m = 0
    for i in 1..n
        x = rand() # random number in [0,1)
        y = rand()
        if x*x + y*y < 1.0 # (*)
            m = m + 1
        end
    end
    4*m*1.0/n
end
def average(t, n)
# repeat t times montecarlo(n)
# write here






end
```

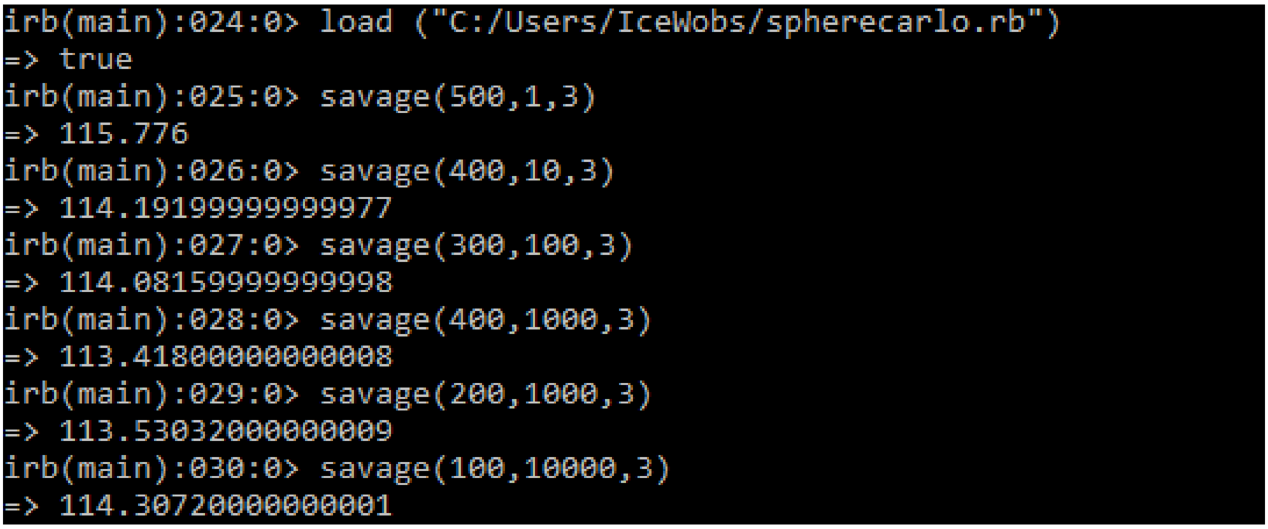
n	t	Average	# correct digits
1	100	3.12	2
10	200	3.208000000000003	1
100	300	3.1573333333333338	2
1000	400	3.1474999999999995	3
10000	500	3.148444799999997	3

2）(optional) Make a program that computes the volume of the 3-dimentional unit sphere, and do the same thing as in Question 1. I assume r (radius) = 3 units for the chart below. (Based off the expected answer from Google being "113.097335529)

n	t	Average ("savage" value)	# correct digits
1	500	115.776	2
10	400	114.19199999999977	2
100	300	114.08159999999998	2
1000	200	113.53032000000009	3
10000	100	114.30720000000001	2



```
irb(main):012:0> average(100,1)
=> 3.12
irb(main):013:0> average(200,10)
=> 3.2080000000000003
irb(main):014:0> average(300,100)
=> 3.1573333333333338
irb(main):015:0> average(400,1000)
=> 3.1474999999999995
irb(main):016:0> average(500,10000)
=> 3.148444799999997
```



```
irb(main):024:0> load ("C:/Users/IceWobs/spherecarlo.rb")
=> true
irb(main):025:0> savage(500,1,3)
=> 115.776
irb(main):026:0> savage(400,10,3)
=> 114.19199999999977
irb(main):027:0> savage(300,100,3)
=> 114.08159999999998
irb(main):028:0> savage(400,1000,3)
=> 113.41800000000008
irb(main):029:0> savage(200,1000,3)
=> 113.53032000000009
irb(main):030:0> savage(100,10000,3)
=> 114.30720000000001
```