

montecarlo.R

linanqiu

Mon Oct 12 14:09:51 2015

```
library(ggplot2)
library(foreach)
library(doMC)
```

```
## Loading required package: iterators
## Loading required package: parallel
```

```
registerDoMC(8)
# let's speed this up with parallelization
```

```
montecarlo = function(steps, delta_t, mu, sigma, s0) {
  s = s0
  for(i in 1 : steps) {
    random = runif(1, 0, 1);
    if(random > 0.5) {
      s = s + (mu / delta_t) * s + (sigma / delta_t) * s
    } else {
      s = s + (mu / delta_t) * s - (sigma / delta_t) * s
    }
  }
  return(s)
}
```

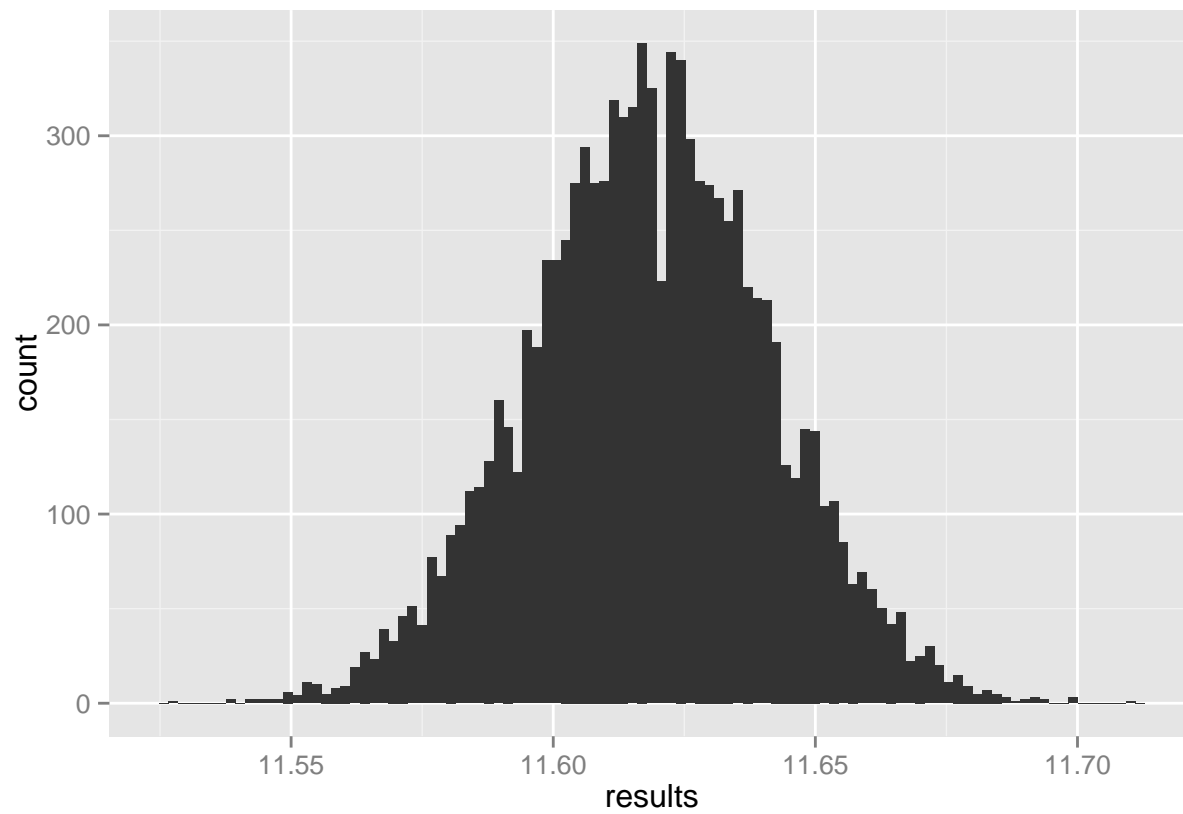
```
mu = 0.15
sigma = 0.2
n = 10000
simulations = 10000
s0 = 10
```

```
results = vector(mode="numeric", length=10000)
results = foreach(i = 1 : length(results)) %dopar% {
  montecarlo(simulations, n, mu, sigma, s0)
}
```

```
results = unlist(results)
data = data.frame(results)
```

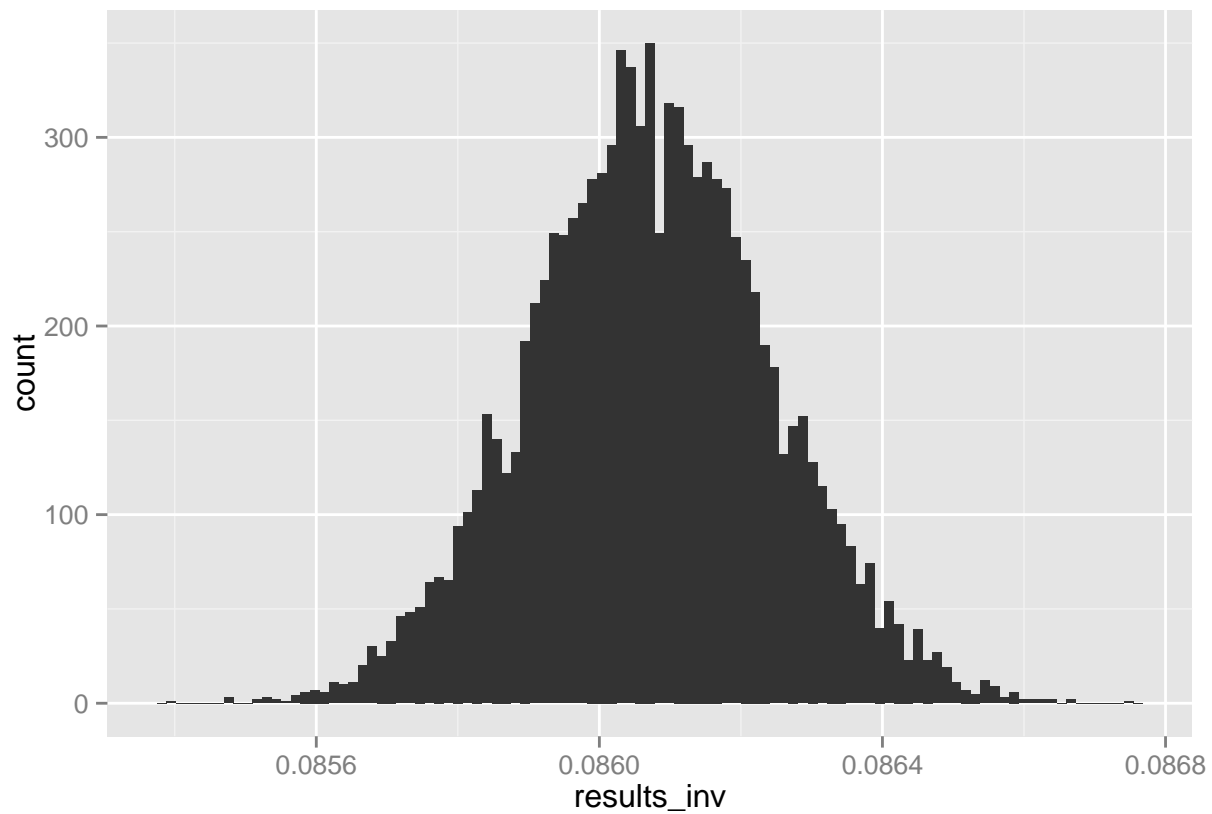
```
qplot(results, data=data, geom="histogram", binwidth=(max(results) - min(results))/100)
```

```
## Warning: position_stack requires constant width: output may be incorrect
```



```
results_inv = 1 / results
data_inv = data.frame(results_inv)

qplot(results_inv, data=data, geom="histogram", binwidth=(max(results_inv) - min(results_inv))/100)
```



```
# theoretical values
mean = s0 * exp(mu)
sd = sigma * s0 * sqrt(1/n)

mean_inv = (1/s0) * exp(sigma^2 - mu)
sd_inv = sigma * (1/s0) * sqrt(1/n)

# real values
mean(results)
```

```
## [1] 11.61821
```

```
mean
```

```
## [1] 11.61834
```

```
sd(results)
```

```
## [1] 0.02313361
```

```
sd
```

```
## [1] 0.02
```

```
mean(results_inv)
```

```
## [1] 0.08607212
```

```
mean_inv
```

```
## [1] 0.08958341
```

```
sd(results_inv)
```

```
## [1] 0.0001713906
```

```
sd_inv
```

```
## [1] 2e-04
```