week9 exercise

Xiang Li

2023/12/29

```
library(rbenchmark)
```

Exercise 1

```
set.seed(9)
x1 = rbinom(n = 20000, size = 5, prob = 0.3)
x2 = rgamma(n = 30000, shape = 2, rate = 1)
```

1

```
mean(x1)
```

```
## [1] 1.5097
```

```
mean(x2)
```

```
## [1] 1.998389
```

`rbinom` generates ramdom samples that follow the specified binomial distribution with parameter `size` and `prob`.

`rgamma` generates ramdom samples that follow the specified gamma distribution with parameter `shape` and `rate`.

2

```
my_mean = function (x) {
  n = length(x)
  sum_x = 0
  for (i in 1:n){
    sum_x = sum_x + x[i]
  }
  mean_x = sum_x/n
  return(mean_x)
}
my_mean(x1)
```

```
## [1] 1.5097
```

3

```
ben1 = benchmark('mean_x1' = {mean(x1)}, 'my_mean_x1' = {my_mean(x1)}, replications = 500)
ben1
```

```
##         test replications elapsed relative user.self sys.self user.child
## 1    mean_x1          500   0.021    1.000     0.021    0.000          0
## 2 my_mean_x1          500   0.327   15.571     0.322    0.002          0
##   sys.child
## 1         0
## 2         0
```

```
ben2 = benchmark('mean_x2' = {mean(x2)}, 'my_mean_x2' = {my_mean(x2)}, replications = 500)
ben2
```

```
##         test replications elapsed relative user.self sys.self user.child
## 1    mean_x2          500   0.028    1.000     0.028    0.000          0
## 2 my_mean_x2          500   0.429   15.321     0.425    0.002          0
##   sys.child
## 1         0
## 2         0
```

Using `mean()` is faster. The execution time of `my_mean()` is more than 16 times longer than `mean()`.

Exercise 2

```
set.seed(9)
n = 2000
p = 500
m1 = matrix(rnorm(n*p, mean = 4.7, sd = 0.5), ncol = p)
```

1

```
ben = benchmark(
        'apply' = {apply(m1, 2, mean)},
        'colMeans' = {colMeans(m1)},
        'pre_for' = {
          c_n = ncol(m1)
          c_mean_v = rep(NA, c_n)
          for (c in 1:c_n){
            c_mean_v[c] = mean(m1[,c])
          }
        },
        'for' = {
          c_n = ncol(m1)
          c_mean_v = c()
          for (c in 1:c_n){
            c_mean_v[c] = mean(m1[,c])
          }
        },
        replications = 100
)
ben
```

```
##       test replications elapsed relative user.self sys.self user.child
## 1    apply          100   1.722   14.718     1.474    0.128          0
## 2 colMeans          100   0.117    1.000     0.094    0.002          0
## 4      for          100   1.251   10.692     1.129    0.110          0
## 3  pre_for          100   1.370   11.709     1.187    0.119          0
##   sys.child
## 1         0
## 2         0
## 4         0
## 3         0
```

`colMeans()` is fastest and `apply()` is slowest.

Exercise 3

```
1 - pbinom(44, size = 127, prob = 0.38)
```

```
## [1] 0.7527784
```

Exercise 4

1

```
dpois(7, lambda = 6)
```

```
## [1] 0.137677
```

2

```
0
```

```
## [1] 0
```

Gamma distribution is a continuous distribution. Thus, for one point, the probability is equal to 0.

3

```
ppois(4, lambda = 6) - ppois(2, lambda = 6)
```

```
## [1] 0.2230877
```

4

```
pgamma(3, shape = 3, rate = 2) - pgamma(1, shape = 3, rate = 2)
```

```
## [1] 0.6147076
```

5

```
ppois(5, lambda = 6)
```

```
## [1] 0.4456796
```

6

```
ppois(3, lambda = 6) + pgamma(10, shape = 3, rate = 2)
```

```
## [1] 1.151203
```

Exercise 5

1

```
set.seed(123)
x = rnorm(1e4, mean = 3, sd = 1.4)
y = rbeta(1e4, shape1 = 2, shape2 = 2)
```

2

```
z = x/y
```

3

```
mean(z)
```

```
## [1] 9.133582
```

```
var(z)
```

```
## [1] 226.961
```