Monte Carlo method

tian47720

2020/3/27

1 Abstract

Monte Carlo method is also known as Random Sampling method or Statistical expertiment method. It is a method based on probailistic statistical thory, the observed values of some experimental results are obtained on the basis of kowned distribution sampling, and the approximate solution of the problem is obtained according to the discribed process.

2 Math Equation

2.1 The standard normal distribution

The equation for the standard normal distribution $N(0,1)$ is

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy$$

2.2 The Monte Carlo method

The equation for the Monte Carlo method is

$$\hat{\Phi}(t) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq t)$$

where X_i is a random sample from $N(0,1)$, $I(\cdot)$ is the indicator function. Then, experiment with the approximation at $n \in \{10^2, 10^3, 10^4\}$ at $t \in \{0.0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09, 3.72\}$ to form a table.

3 Table

The approximation results are shown in the table.

```
t = c(0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09, 3.72)
n = c(100, 1000, 10000)
p = matrix(0, nrow=9, ncol=3)
a = 0
for(i in 1:9)
  for(j in 1:3){
    a = a+1
    num = rnorm(n[j], 0, 1)
    p[i, j] = mean(num<t[i])
    print(a)
  }
```

```
## [1] 1
## [1] 2
## [1] 3
```

Table 1: (ref:norm)

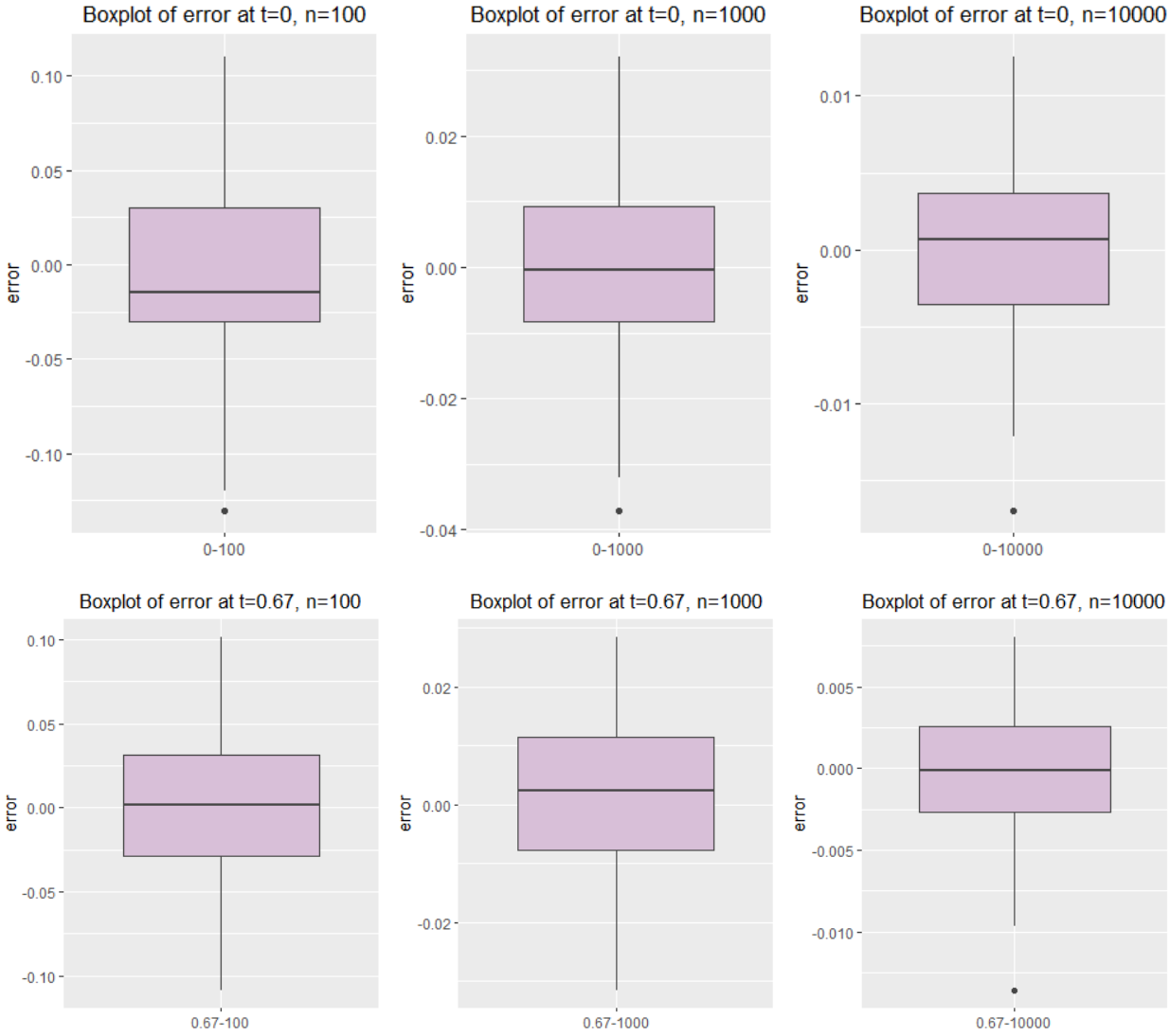
	100	1000	10000
0	0.52	0.498	0.5118
0.67	0.70	0.752	0.7444
0.84	0.80	0.798	0.7957
1.28	0.96	0.896	0.9064
1.65	0.91	0.941	0.9463
2.32	0.98	0.989	0.9897
2.58	1.00	0.992	0.9933
3.09	1.00	1.000	0.9993
3.72	1.00	1.000	0.9999

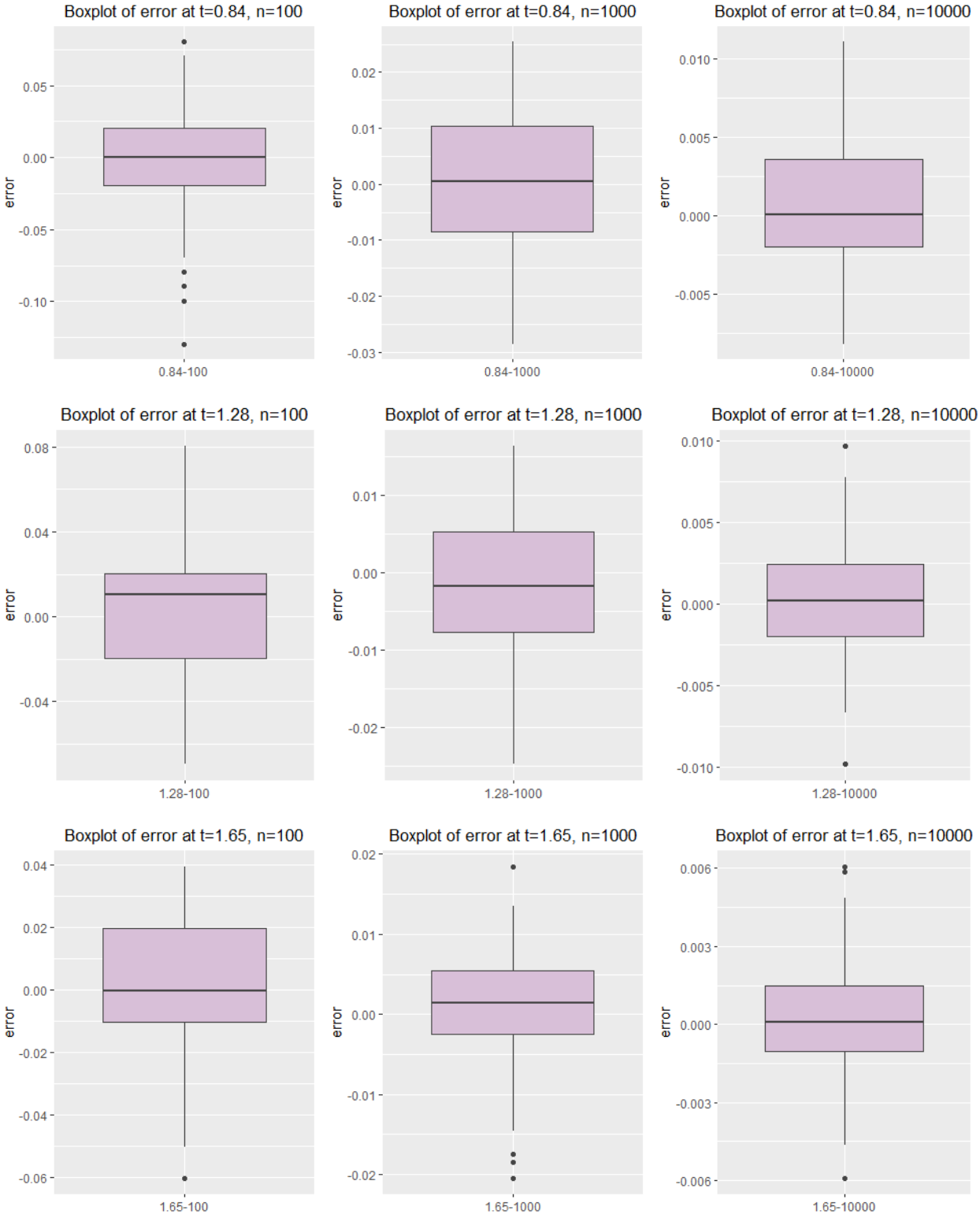
```
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
```

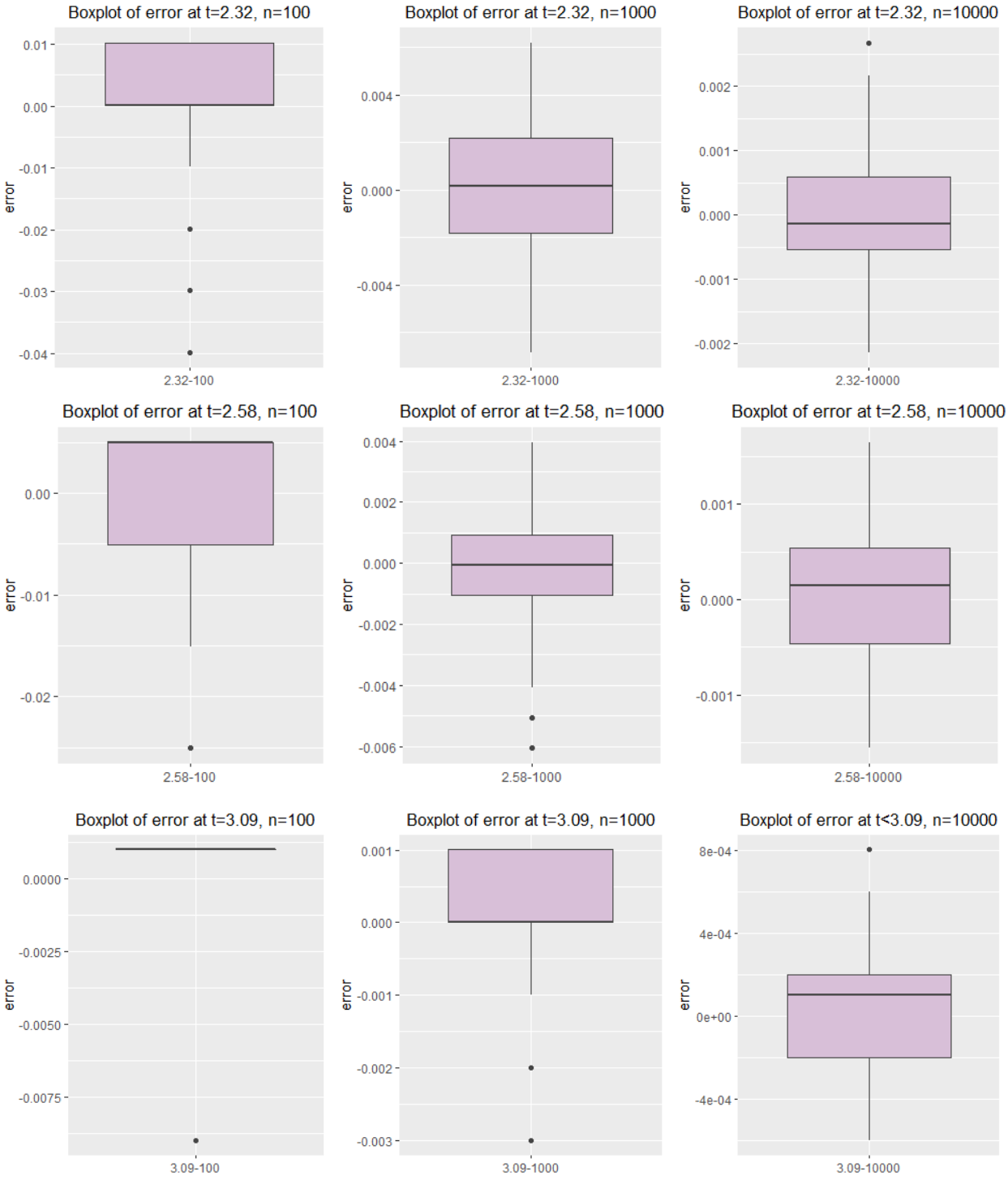
```
rownames(p)<-t
colnames(p)<-n

library(knitr)
library(magrittr)
library(kableExtra)
library(callr)
library(webshot)

kable(p, booktabs=TRUE, caption='(ref:norm)') %>%
 kable_styling(bootstrap_options = "striped",full_width = F) %>%
 column_spec(1,bold=T)
```

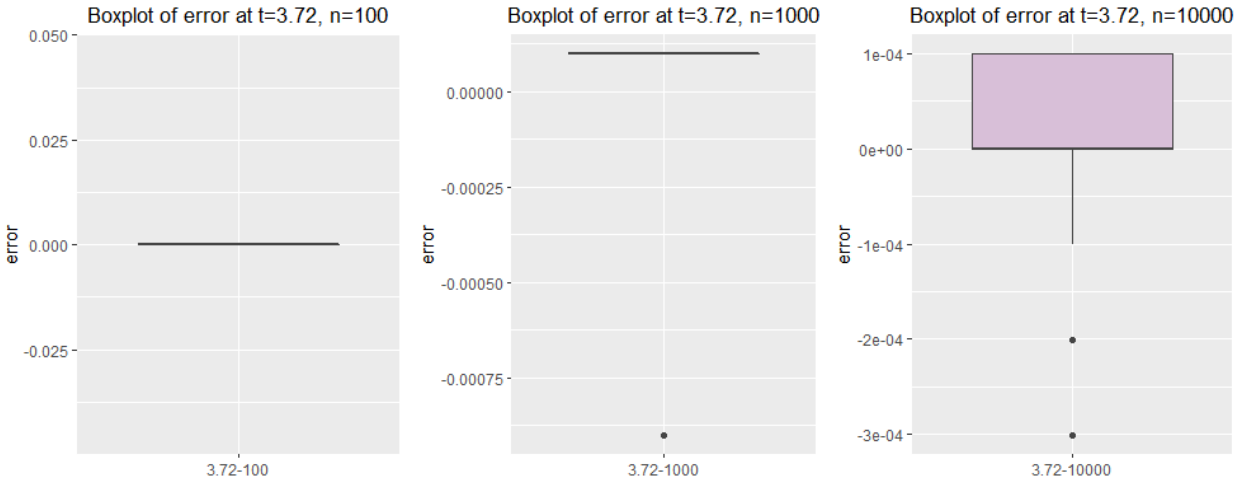
4 figure

After repeat the experiment 100 times, I draw box plots of the 100 approximation errors at each t using **ggplot2** for each n.




Boxplot of error at t=0.84, n=100
error
0.05
0.00
-0.05
-0.10
0.84-100
Boxplot of error at t=0.84, n=1000
error
0.02
0.01
0.00
-0.01
-0.02
-0.03
0.84-1000
Boxplot of error at t=0.84, n=10000
error
0.010
0.005
0.000
-0.005
0.84-10000
Boxplot of error at t=1.28, n=100
error
0.08
0.04
0.00
-0.04
1.28-100
Boxplot of error at t=1.28, n=1000
error
0.01
0.00
-0.01
-0.02
1.28-1000
Boxplot of error at t=1.28, n=10000
error
0.010
0.005
0.000
-0.005
-0.010
1.28-10000
Boxplot of error at t=1.65, n=100
error
0.04
0.02
0.00
-0.02
-0.04
-0.06
1.65-100
Boxplot of error at t=1.65, n=1000
error
0.02
0.01
0.00
-0.01
-0.02
1.65-1000
Boxplot of error at t=1.65, n=10000
error
0.006
0.003
0.000
-0.003
-0.006
1.65-10000


Boxplot of error at t=2.32, n=100
error
0.01
0.00
-0.01
-0.02
-0.03
-0.04
2.32-100
Boxplot of error at t=2.32, n=1000
error
0.004
0.000
-0.004
2.32-1000
Boxplot of error at t=2.32, n=10000
error
0.002
0.001
0.000
-0.001
-0.002
2.32-10000
Boxplot of error at t=2.58, n=100
error
0.00
-0.01
-0.02
2.58-100
Boxplot of error at t=2.58, n=1000
error
0.004
0.002
0.000
-0.002
-0.004
-0.006
2.58-1000
Boxplot of error at t=2.58, n=10000
error
0.001
0.000
-0.001
2.58-10000
Boxplot of error at t=3.09, n=100
error
0.0000
-0.0025
-0.0050
-0.0075
3.09-100
Boxplot of error at t=3.09, n=1000
error
0.001
0.000
-0.001
-0.002
-0.003
3.09-1000
Boxplot of error at t<3.09, n=10000
error
8e-04
4e-04
0e+00
-4e-04
3.09-10000


Boxplot of error at t=3.72, n=100
0.050
0.025
0.000
-0.025
error
3.72-100
Boxplot of error at t=3.72, n=1000
0.00000
-0.00025
-0.00050
-0.00075
error
3.72-1000
Boxplot of error at t=3.72, n=10000
1e-04
0e+00
-1e-04
-2e-04
-3e-04
error
3.72-10000