Approximation of the distribution function of $N(0,1)$

Lijiao Liu

2020/3/25

Abstract

This is a report that calculating the approximate value of the standard normal distribution by the Monte Carlo methods.

1 Math Equations

The distribution function of $N(0,1)$ is:

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

The equation for the Monte Carlo methods is:

$$\hat{\Phi}(t) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq t), \tag{2}$$

where X_i's are a random sample from $N(0,1)$, and $I(\cdot)$ is the indicator function.

2 Table

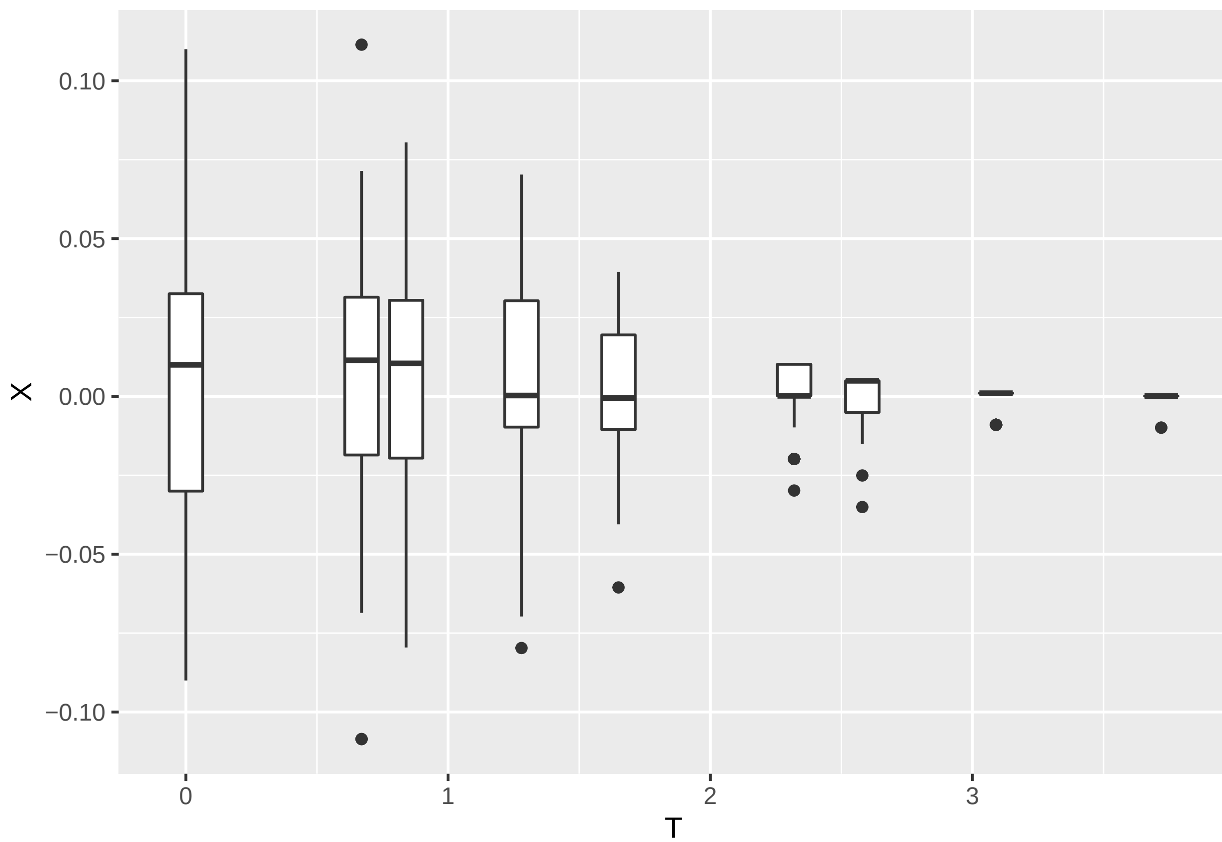
Experiment with the approximation at $n \in \{10^2, 10^3, 10^4\}$ at $t \in \{0.0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09, 3.72\}$ to form a table with the true value for comparison.

	100	1000	10000	true_value
0	0.53	0.517	0.503	0.500
0.67	0.74	0.748	0.746	0.749
0.84	0.84	0.802	0.800	0.800
1.28	0.93	0.899	0.903	0.900
1.65	0.93	0.960	0.951	0.951
2.32	0.99	0.988	0.991	0.990
2.58	1.00	0.996	0.994	0.995
3.09	0.99	0.999	0.998	0.999
3.72	1.00	1.000	1.000	1.000

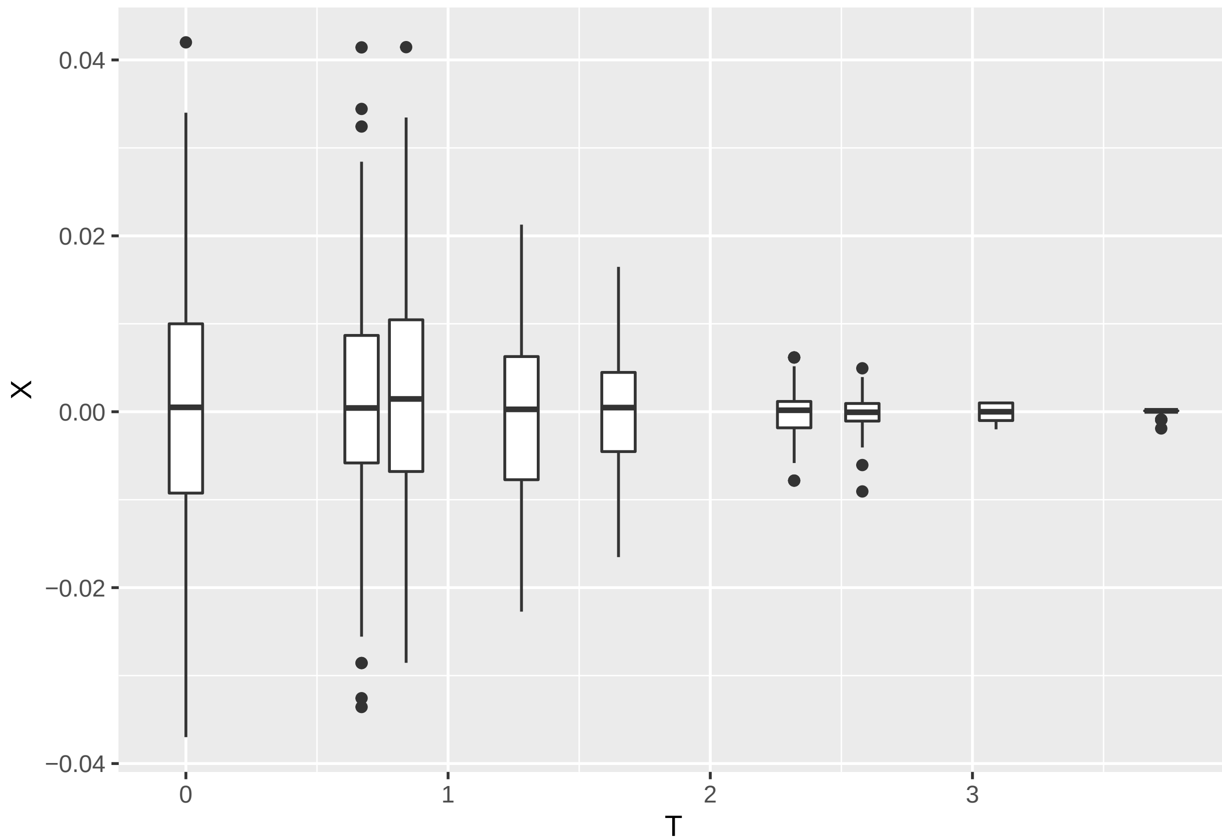
3 Figures

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

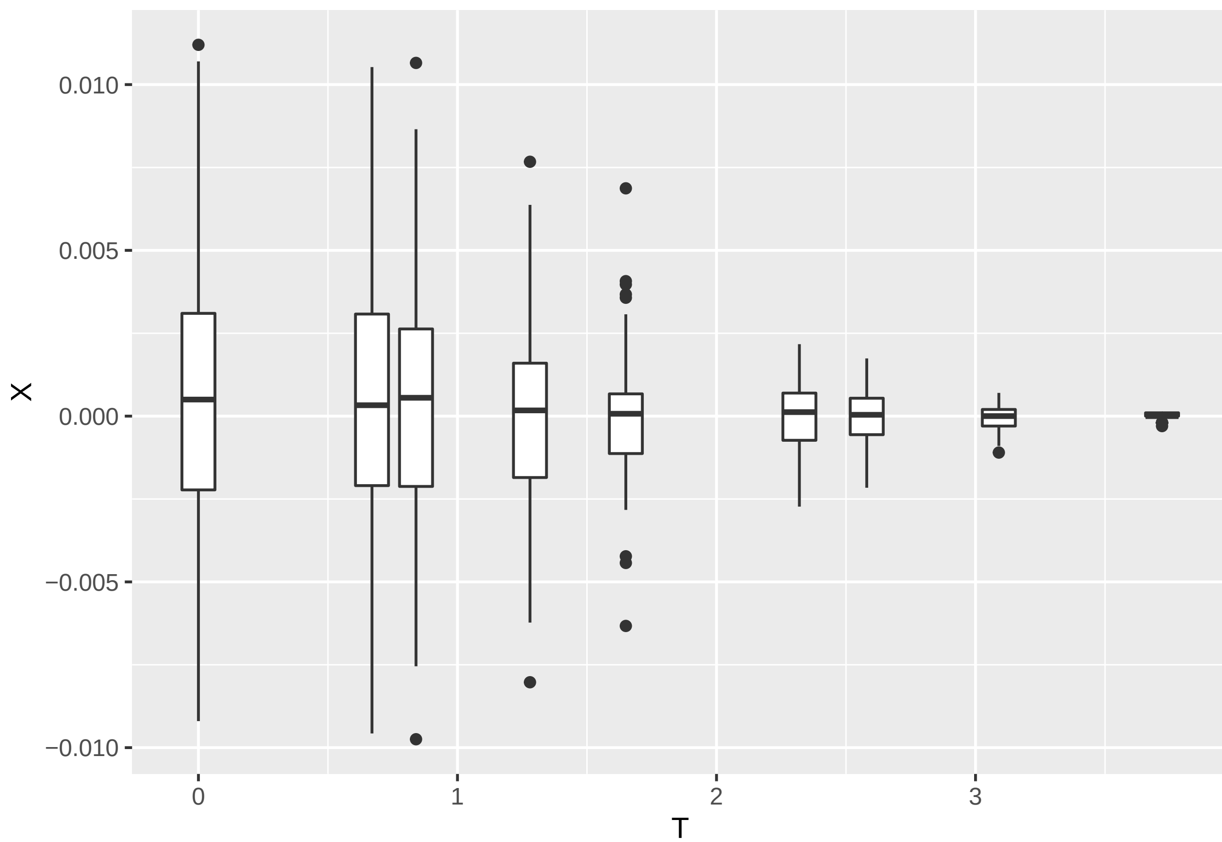
When n=100,the box plots of the 100 approximation errors at each t are:



When n=1000,the box plots of the 100 approximation errors at each t are:



When n=10000,the box plots of the 100 approximation errors at each t are:



4 Chunks of R Code

```
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)
  }
rownames(p)<-t
colnames(p)<-n
true_value<-c(pnorm(0),pnorm(0.67),pnorm(0.84),pnorm(1.28),pnorm(1.65),
           pnorm(2.32),pnorm(2.58),pnorm(3.09),pnorm(3.72))
p<-cbind(p,true_value)
p<-round(p,digits=3)
#make figure more beautiful
library(knitr)
library(magrittr)
```

```r
library(kableExtra)
  kable(p, booktabs=TRUE) %>%
  kable_styling(bootstrap_options = "striped",full_width = F) %>%
 column_spec(1,bold=T)#full_width=F, is or isn't full of screen
```

For the table

```r
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1)
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)
n=10^2
z=matrix(0,100,9)
w=matrix(0,9,n)
for(p in 1:100)
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
{q[s,1]=e[s]}
for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```

```r
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1)
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)
n=10^3
z=matrix(0,100,9)
w=matrix(0,9,n)
for(p in 1:100)
```

```
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
{q[s,1]=e[s]}
for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```

```
x=pnorm(  c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72), mean = 0, sd = 1)
t=c(0.0,0.67, 0.84,1.28,1.65,2.32,2.58,3.09,3.72)
n=10^4
z=matrix(0,100,9)
w=matrix(0,9,n)
for(p in 1:100)
{ y=c(rnorm(n,mean=0,sd=1))
  for(k in 1:9)
  {
   for(j in 1:n)
  {w[k,j]=sign(y[j]<=t[k])}
z[p,k]=sum(w[k,])/n}}
z=as.data.frame(z)
r=c(z$V1,z$V2,z$V3,z$V4,z$V5,z$V6,z$V7,z$V8,z$V9)
e=c(rep(0.0,100),rep(0.67,100),rep(0.84,100),rep(1.28,100),rep(1.65,100),rep(2.32,100),rep(2.58
q=data.frame(T=rep(0,100),X=0)
for(s in 1:900)
{q[s,2]=r[s]}
for(s in 1:900)
```

```r
{q[s,1]=e[s]}
for(a in 1:100)
 { q[a,2]=q[a,2]-x[1]
 q[a+100,2]=q[a+100,2]-x[2]
 q[a+200,2]=q[a+200,2]-x[3]
 q[a+300,2]=q[a+300,2]-x[4]
 q[a+400,2]=q[a+400,2]-x[5]
 q[a+500,2]=q[a+500,2]-x[6]
 q[a+600,2]=q[a+600,2]-x[7]
 q[a+700,2]=q[a+700,2]-x[8]
 q[a+800,2]=q[a+800,2]-x[9]}
library(ggplot2)
ggplot(q,aes(T,X,group=T)) + geom_boxplot()
```

For the figures

5 Reference

Refer to the code of other students in the process of compiling the code.