Approximation of The Distribution Function of N(0,1) by Using The Monte Carlo Methods

Jiang Bian & Haoyu Yin

30 January 2018

Abstract

This is an approximation of the distribution function of N(0,1) by using the Monte Carlo Methods. To do the experiment, we set up a function with specific values of two parameters, and for each pair of those parameters, we repeat the experiment 100 times to compare the results with the true values of the function that we want to approximate.

Math Equations

Original Function

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

Approximated Function

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

Methodology

We do the experiment with the approximation at $n \in \{10^2, 10^3, 10^4\}$

at $t \in \{0.0, 0.67, 0.84, 1.28, 1.65, 2.32, 2.58, 3.09, 3.72\}$

```
n <- c(100, 1000, 10000)
t <- c(0.0,0.67,0.84,1.28,1.65,2.32,2.58,3.09,3.72)
```

Then, we use for loop function to generate 27 different combinations of n and t and each repeat 100 times.

We receive:

When n = 100,

```
knitr::kable(head(result[1,,]), digits = 4, col.names = t)
```

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
0.48	0.80	0.81	0.89	0.99	1.00	0.99	1	1.00
0.44	0.64	0.84	0.91	0.93	0.97	1.00	1	1.00
0.53	0.82	0.78	0.85	0.96	0.99	1.00	1	1.00
0.47	0.76	0.85	0.92	0.96	0.99	0.99	1	1.00
0.52	0.76	0.76	0.88	0.98	0.98	0.97	1	1.00
0.47	0.68	0.80	0.88	0.97	0.99	0.99	1	0.99

When n = 1000,

```
knitr::kable(head(result[2,,]), digits = 4, col.names = t)
```

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
0.471	0.747	0.823	0.903	0.955	0.992	0.992	0.998	0.999
0.510	0.750	0.809	0.906	0.955	0.996	0.996	0.995	0.999
0.492	0.728	0.809	0.887	0.956	0.982	0.992	0.998	1.000
0.504	0.732	0.785	0.887	0.961	0.991	0.997	0.999	1.000
0.515	0.743	0.771	0.893	0.951	0.992	0.994	0.999	1.000
0.514	0.731	0.809	0.897	0.948	0.991	0.993	0.998	0.999

When n = 10000,

```
knitr::kable(head(result[3,,]), digits = 4, col.names = t)
```

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
0.5010	0.7496	0.7980	0.8996	0.9532	0.9901	0.9959	0.9990	0.9999
0.5036	0.7550	0.7965	0.9025	0.9497	0.9905	0.9950	0.9990	0.9998
0.5018	0.7429	0.7977	0.8993	0.9532	0.9888	0.9950	0.9987	0.9999
0.4991	0.7552	0.7971	0.9029	0.9475	0.9910	0.9945	0.9993	0.9998
0.5020	0.7527	0.8009	0.9016	0.9527	0.9906	0.9955	0.9986	1.0000
0.4983	0.7441	0.8002	0.9022	0.9490	0.9880	0.9940	0.9993	1.0000

For comparision, the difference between valuations and true values are:

When n = 100,

```
knitr::kable(head(diff[1,,]), digits = 4, col.names = t)
```

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
-0.02	0.0514	0.0105	-0.0097	0.0395	0.0102	-0.0051	0.001	0.0001
-0.06	-0.1086	0.0405	0.0103	-0.0205	-0.0198	0.0049	0.001	0.0001
0.03	0.0714	-0.0195	-0.0497	0.0095	0.0002	0.0049	0.001	0.0001
-0.03	0.0114	0.0505	0.0203	0.0095	0.0002	-0.0051	0.001	0.0001
0.02	0.0114	-0.0395	-0.0197	0.0295	-0.0098	-0.0251	0.001	0.0001
-0.03	-0.0686	0.0005	-0.0197	0.0195	0.0002	-0.0051	0.001	-0.0099

When n = 1000,

```
knitr::kable(head(diff[2,,]), digits = 4, col.names = t)
```

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
-0.029	-0.0016	0.0235	0.0033	0.0045	0.0022	-0.0031	-0.001	-9e-04
0.010	0.0014	0.0095	0.0063	0.0045	0.0062	0.0009	-0.004	-9e-04
-0.008	-0.0206	0.0095	-0.0127	0.0055	-0.0078	-0.0031	-0.001	1e-04

0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
0.004	-0.0166	-0.0145	-0.0127	0.0105	0.0012	0.0019	0.000	1e-04
0.015	-0.0056	-0.0285	-0.0067	0.0005	0.0022	-0.0011	0.000	1e-04
0.014	-0.0176	0.0095	-0.0027	-0.0025	0.0012	-0.0021	-0.001	-9e-04

When n = 10000,

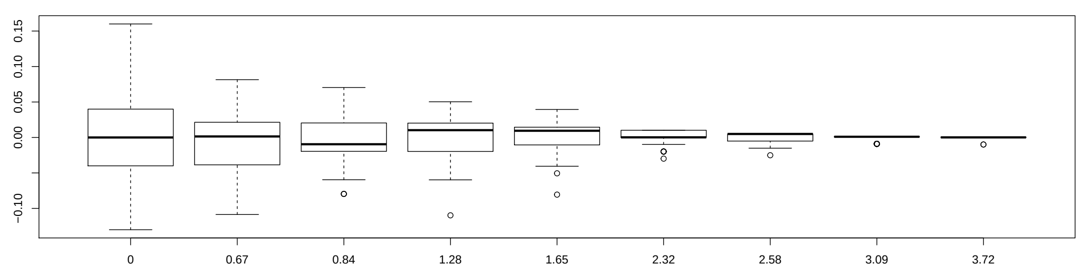
```
knitr::kable(head(diff[3,,]), digits = 4, col.names = t)
```

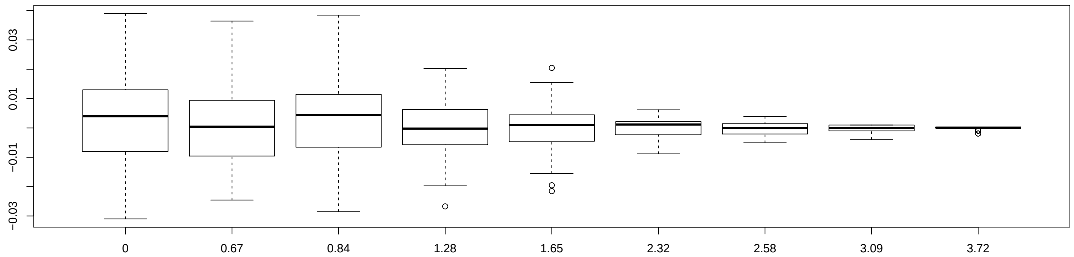
0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
0.0010	0.0010	-0.0015	-0.0001	0.0027	0.0003	0.0008	0e+00	0e+00
0.0036	0.0064	-0.0030	0.0028	-0.0008	0.0007	-0.0001	0e+00	-1e-04
0.0018	-0.0057	-0.0018	-0.0004	0.0027	-0.0010	-0.0001	-3e-04	0e+00
-0.0009	0.0066	-0.0024	0.0032	-0.0030	0.0012	-0.0006	3e-04	-1e-04
0.0020	0.0041	0.0014	0.0019	0.0022	0.0008	0.0004	-4e-04	1e-04
-0.0017	-0.0045	0.0007	0.0025	-0.0015	-0.0018	-0.0011	3e-04	1e-04

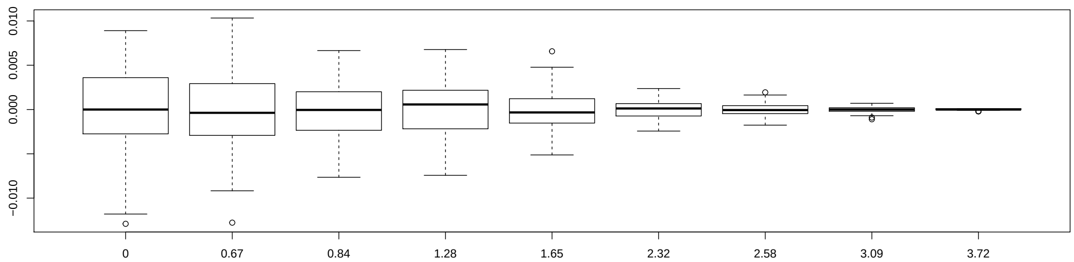
Plots

We draw the box plots to show the bias at all t.

```
m <- 1
for (m in 1:length(n)){
  boxplot(diff[m,,])
  m <- m + 1
}
```







Conclusion

In conclusion, we found that the approximation of the function that we set up is ideal and when n becomes bigger, the result is more accurate.