The Use of R Language in the Teaching of Central Limit Theorem

Cheang Wai Kwong
waikwong.cheang@nie.edu.sg
National Institute of Education
Nanyang Technological University
Singapore

Abstract: *The Central Limit Theorem (CLT) is probably the most important theorem in statistics. In the teaching of CLT, issues encountered include: (i) How large should the sample size n be for CLT to hold? Is the rule of thumb $n \geq 30$ adequate? (ii) How important is the condition of normal population for the convergence in distribution of the t-statistic to normal? (iii) Will CLT still hold if the sample observations are not independent? To provide students with better insight into these issues, we can perform empirical study by simulating samples from different distributions, and exploring the behaviours of the z-statistic and t-statistic as the sample size gets large. Statistical software packages such as SAS can be used to perform simulation. However, these packages are not free and may not be cost-effective to implement. The R language is a powerful software for data analysis within which many statistical procedures have been implemented. It is an official part of the Free Software Foundation's GNU Project (and so R is free). The strength of R derives from its many capabilities besides being a data analysis tool. In this paper, we explore the simulation capability of R in teaching the Central Limit Theorem, as well as its graphing capability in presenting the simulation results. With its ease of adaptability according to user's need, R has the potential to be an effective teaching tool for other topics of statistics.*

1. Introduction

The R language is a powerful software for data analysis within which many statistical procedures have been implemented. R can be considered as a "free" implementation of the S language which was originally developed at Bell Laboratories (of AT&T and now Alcatel-Lucent). A commercial implementation of S is S-PLUS from Insightful Corporation. R is an official part of the Free Software Foundation's GNU Project, and is distributed as Free Software under the terms of the GNU General Public License. The development of R is supported by the R Foundation seated in Vienna. R compiles and runs on a wide variety of Unix platforms (including Linux), Windows and MacOS. Precompiled R binaries as well as the R source code can be downloaded from http://www.r-project.org/.

The strength of R derives not only from being a data analysis software, but also from its powerful computing and flexible graphing capabilities. Cheang (2004) discussed some potential uses of R in mathematics teaching and computation. Another attractive feature of R is its simulation capability. We can easily simulate observations from common distributions like binomial and normal. For example, the following command in R generates a random sample of size $n = 30$ from $N(0, 1)$ distribution:

```
rnorm(n=30, mean=0, sd=1)
```

Using the flexible graphing capability of R, simulation results can easily be presented in publication-quality plots according to user's need. In this paper, we explore how the simulation and graphing capabilities of R can be used in the teaching of the Central Limit Theorem.

2. Approximation to Binomial: Normal versus Poisson

As a prelude to the Central Limit Theorem (CLT), we consider normal and Poisson approximations to binomial. In the teaching of these approximations, we want students to recognize that one is better than the other under certain condition.

Suppose X has a binomial(n, p) distribution. By writing $X = \sum_{i=1}^{n} Y_i$, where Y_1, Y_2, …, Y_n are Bernoulli trials, it can be justified using CLT that for large n, the distribution of X is approximately normal with mean np and variance npq ($q = 1 - p$). When n is large, another possible approximate distribution for binomial is Poisson. The usual rule of thumb for choosing normal approximation is $np > 5$ (and $nq > 5$). If p is "small" so that $np < 5$, we use Poisson approximation. How can we "justify" (at least empirically) such a rule of thumb to our students? One way is through simulating the distribution of X.

Using the R code given in Appendix A.1, for each selected pair of (n, p), 10000 replications of X are generated. Figure 1 shows the resulting density histogram, with the histogram cells set to intervals of the form [a, b). Since the common probability density functions and probability mass functions are inbuilt in R, we can easily demonstrate graphically the appropriateness of normal and Poisson approximations to binomial. The normal probability density function with mean np and variance npq is plotted as red dotted lines. For comparison, the Poisson probabilities with mean np are plotted as blue dots. This simulation suggests that for $np < 5$, Poisson would indeed provide a better approximation to binomial.

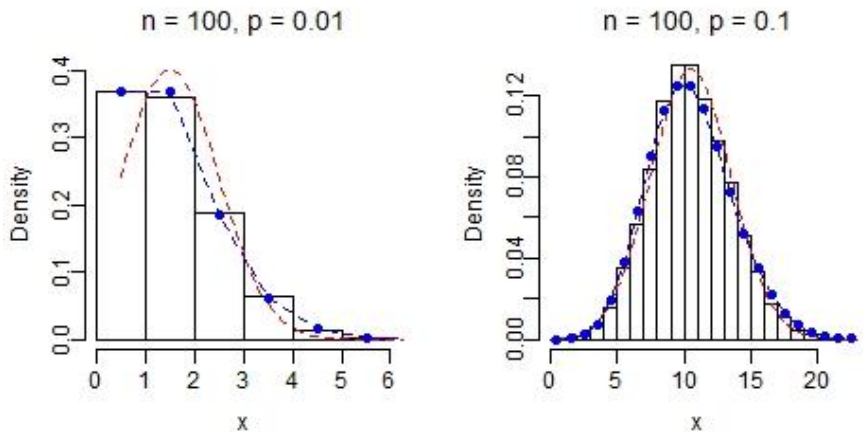

Figure 1 Normal and Poisson approximations to binomial

3. The Central Limit Theorem

The Central Limit Theorem is probably the most important theorem in statistics. Let X_1, X_2, …, X_n be a random sample from *any* distribution (not necessarily normal) with mean μ and variance σ^2. CLT says that for large sample size n, the sampling distribution of the sample mean $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ is approximately normal with mean μ and variance σ^2/n. More precisely, the z-statistic,

$$Z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}},$$

converges to $N(0, 1)$ in distribution as $n \to \infty$.

We may not want to prove CLT in an introductory statistics course, as a typical proof would require the use of more advanced machinery like moment generating function. Instead, we can demonstrate CLT by displaying graphically that the density histogram of sample means from any distribution approaches a bell-shaped (normal) density as *n* increases. Wild and Seber (2000, p. 284-287) presented simulation results for four distributions that do not look like normal: triangular, uniform, exponential and quadratic. Cheang (2004) showed how the simulation can be done using R for random samples from an exponential distribution, while Verzani (2005, p. 168) provided the R code for uniform distribution. Here, we further illustrate the simulation capability of R by using random samples from a gamma distribution with shape parameter α and scale parameter β, i.e., with probability density function (p.d.f.)

$$f(x) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}, \quad x > 0.$$

The choice of gamma distribution would allow us to investigate questions like:

- How large a sample size *n* is needed for CLT to hold? Is the rule of thumb $n \geq 30$ adequate?
- How would the shape of the distribution affect the sample size needed?

To answer the first question, for each selected pair of (α, β), we can simulate 10000 random samples of size *n*, for *n* = 1, 10, 20, etc. The R code is given in Appendix A.2. By plotting the density histogram of the 10000 sample means one-at-a-time as *n* increases, and superimposing the p.d.f. of $N(\alpha\beta, \alpha\beta^2)$, we can check whether the sampling distribution of $\bar{X}$ is closely approximated by the normal distribution if $n < 30$.

To answer the second question, Figure 2 compares that the density histograms of the sample means from gamma distributions with (α, β) = (0.5, 1) and (2, 1). The R code in Appendix A.2 can easily be modified for this purpose. The simulation results suggest that one factor affecting the sample size needed for $\bar{X}$ to be approximately normal is the "shape" of the underlying distribution. A larger sample size is needed for a less symmetrical distribution.

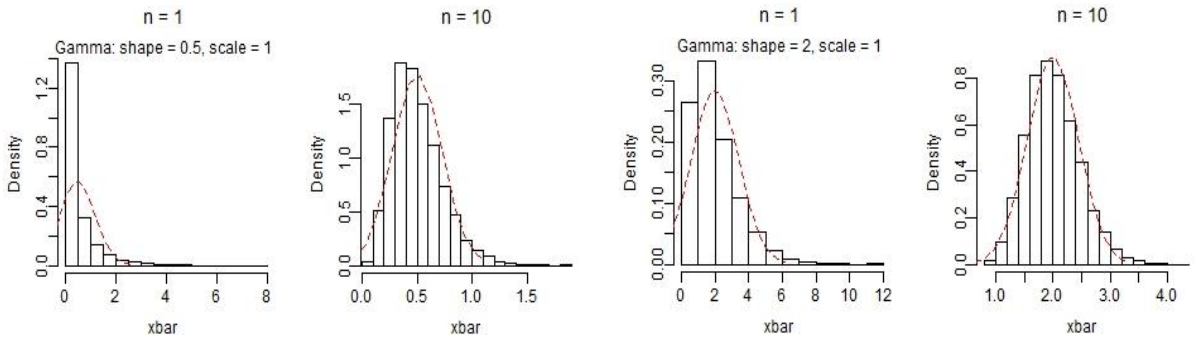

Figure 2 CLT for random samples from gamma distribution of varying shapes

4. Sampling Distribution of *t*-statistic

Let $X_1, X_2, \ldots, X_n$ be a random sample from a $N(\mu, \sigma^2)$ distribution. If we replace σ in the *z*-statistic by the sample standard deviation $S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$, we know that the resulting *t*-statistic,

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}},$$

has a t_{n-1} distribution.

We can stimulate students' thinking with questions like: Can we use $N(0, 1)$ to approximate the distribution of T when n is "small"? To answer this question, we can simulate 10000 random samples from $N(\mu, \sigma^2)$ of size n, for $n = 10, 20$, etc. If the density histogram of the *t*-statistic and the p.d.f. of $N(0, 1)$ are too crude to "see" whether the sampling distribution of T is approximately normal, we can use a normal Q-Q plot to detect any deviation from normality of the *t*-statistic, as shown in Figure 3.

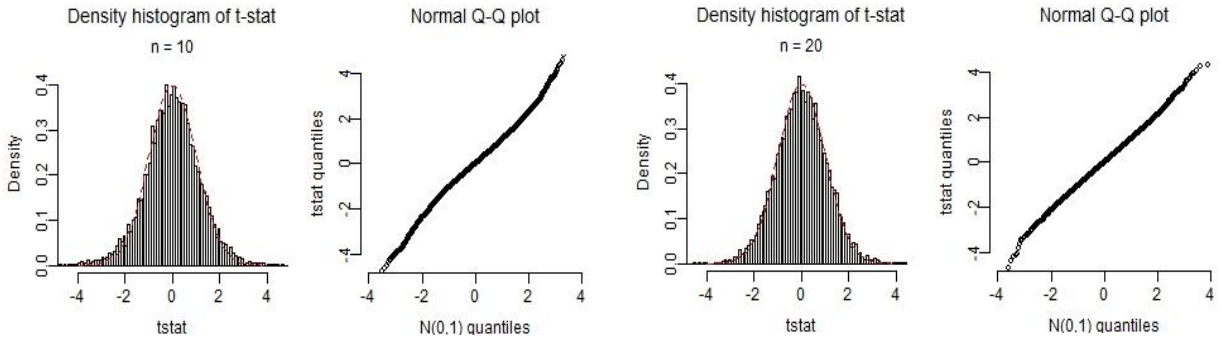

Figure 3 Sampling distribution of *t*-statistic with samples from normal

What happens if the underlying population is not normal? By CLT (and the convergence in probability of S^2 to σ^2), it can be shown that T is approximately $N(0, 1)$ when n is large. How large must n be for this to happen? Is $n \geq 30$ an over-conservative rule of thumb? Again, we can investigate such questions through simulating the sampling distribution of the *t*-statistic. The R code is given in Appendix A.3. Figure 4 displays the density histograms of the *t*-statistic and *z*-statistic for 10000 random samples from gamma distribution with $(\alpha, \beta) = (2, 1)$. The Q-Q plot indicates a deviation from $N(0, 1)$ for the *t*-statistic when $n = 20$.

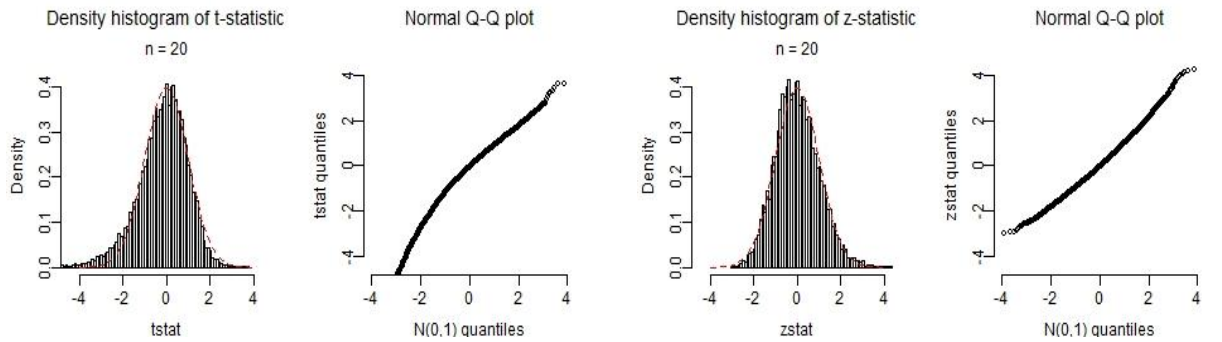

Figure 4 Sampling distributions of *t*-statistic and *z*-statistic with samples from gamma

5. Dependent Observations

A standard assumption for CLT to hold is random sample, i.e., independent and identically distributed (i.i.d.) observations. Is CLT valid for dependent observations? Consider a stationary autoregressive process $\{X_i\}$ of order 1 [AR(1)] with zero mean, defined by

$$X_i = \phi X_{i-1} + \varepsilon_i ,$$

where $-1 < \phi < 1$ and the errors ε_i's are i.i.d. $N(0, \sigma^2)$. For large n, the sample mean $\overline{X}$ is approximately normal, as it can be shown that

$$\frac{(1-\phi)\overline{X}}{\sigma/\sqrt{n}} \rightarrow N(0,1) \text{ in distribution.}$$

So, the usual z-statistic,

$$Z = \frac{\overline{X}}{\sigma/\sqrt{n}} ,$$

does not converge to $N(0, 1)$ in distribution, but rather $Z \sim N(0, 1/(1-\phi)^2)$ approximately when n is large. Proof of such a result is probably beyond the scope of an introductory statistics course, as it requires knowledge of time series analysis [e.g., Box et al. (1994)]. Nevertheless, we can still demonstrate the result by simulating the sampling distribution of Z from AR(1) series. As R has inbuilt function (arima.sim) to generate AR(1) series, the R code in Appendix A.3 can easily be modified to perform such as a simulation.

How about the sampling distribution of the usual t-statistic,

$$T = \frac{\overline{X}}{S/\sqrt{n}} \ ?$$

Will T be approximately normal for large n? If so, what is the impact of the AR parameter ϕ on the sample size needed to achieve normality? Figure 5 shows the density histograms of the z-statistic and t-statistic for $n = 30$ and $\phi = 0.5$. Superimposing the p.d.f. of $N(0, 1/(1-\phi)^2)$ suggests that $n > 30$ is needed for T to be approximately normal. For students to investigate these questions further on their own, we can provide the R code in Appendix A.3 as a template for them to make appropriate modification. So the R language can serve as a tool for students to explore and verify theoretical results that may be not easily established without advanced knowledge.

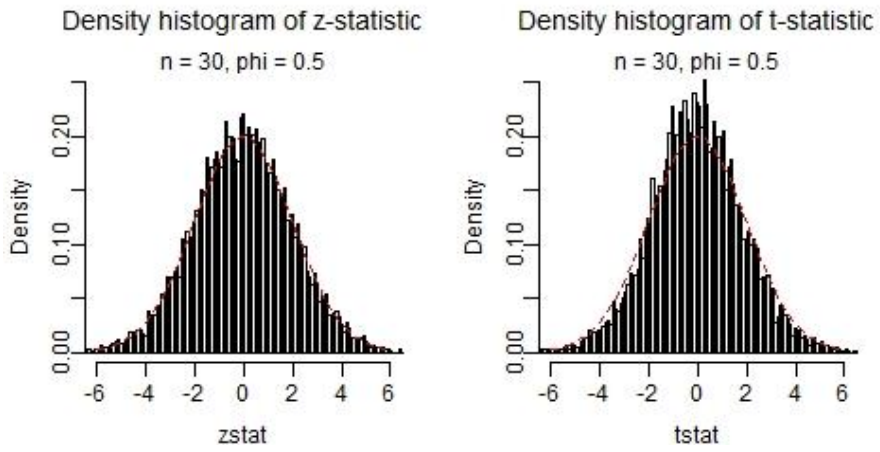

Figure 5 Sampling distribution of z-statistic and t-statistic from AR(1) series

6. Conclusion

We see that R can simulate random samples from distributions like binomial, normal and gamma, and time series from processes like AR(1). We can further exploit this simulation capability of R in teaching other topics of statistics, such as binomial approximation to hypergeometric. Hopefully, through simulation, we can encourage students to become more critical of the various rules of thumb in statistics. A useful reference of how to use R in introductory statistics is Verzani (2005).

With its simulation capability and ease of adaptability, R has the potential to be an effective teaching and research tool in statistics. As a free software, the R language provides a platform for educators and researchers to "freely" explore how technology can be applied into their teaching and research.

References

[1] Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control, 3rd ed.*, Prentice Hall, Englewood Cliffs, New Jersey.

[2] Cheang, W. K. (2004). The use of R language in mathematics teaching and computation. *Proceedings of the 9th Asian Technology Conference in Mathematics*, 402–409, December 2004, Singapore.

[3] Verzani, J. (2005). *Using R for Introductory Statistics*, Chapman & Hall/CRC.

[4] Wild, C. J., and Seber, G. A. F. (2000). *Chance Encounters: A First Course in Data Analysis and Inference*, John Wiley, New York.

Appendix: R codes

In this appendix, we give the R codes for some of the examples discussed. A good introduction to the R language is *The R Manual* edited by the R Development Core Team and downloadable from http://www.r-project.org/.

A.1 Normal and Poisson approximations to binomial

```
jpeg("C:/Users/plot.jpg",width=480,height=240,quality=100)
par(mfrow=c(1,2),oma=c(0,0,0,0),mar=c(3,3,3,2),bty="n")

# To put the random number generator in a reproducible state
set.seed(321)

# Binomial(n,p)
n <- c(100,100)
p <- c(0.01,0.1)
r <- 10000  # No. of replications

for (i in 1:length(n))
{ x <- rbinom(r,n[i],p[i])  # x[1], ..., x[r] are i.i.d. Binomial(n,p)

  mu <- n[i]*p[i]
```

```
  sigma2 <- n[i]*p[i]*(1-p[i])
  Npdf <- dnorm(seq(0,n[i],0.1),mu,sqrt(sigma2))  # pdf of N(mu,sigma2)

  lambda <- n[i]*p[i]
  Ppf <- dpois(0:n[i],lambda)  # pf of Poisson(lambda)

# Histogram cells are set to intervals of the form [a,b)
  hist(x,breaks=seq(0,max(x),1),prob=T,right=F,main="",xlim=c(0,max(x)),
  ylim=c(0,max(Npdf,Ppf)),mgp=c(2,0.5,0),cex=1.0)
  mtext(side=3,line=1,outer=F,paste("n = ",n[i],", p = ",p[i],sep=""),cex=1.2)

# Plot pdf of N(mu,sigma2)
  lines(seq(0,n[i],0.1)+0.5,Npdf,lty=2,col="red")

# Plot pf of Poisson(lambda)
  points(c(0:n[i])+0.5,Ppf,pch=19,col="blue")
  lines(c(0:n[i])+0.5,Ppf,lty=2,col="blue")
}
```

A.2 Central Limit Theorem for gamma distribution

```
par(ask=T)
set.seed(321)

n <- c(1,10,20)    # n = vector of increasing sample sizes
r <- 10000         # r = no. of replications of a given sample size

# Gamma(alpha,beta)
alpha <- 0.5
beta <- 1

mu <- alpha*beta
sigma <- sqrt(alpha)*beta

for (i in 1:length(n))
{ xbar <- rep(NA,r)
  sxbar <- sigma/sqrt(n[i])

  for (j in 1:r)
  {
# Generation of random sample of size n from Gamma(alpha,beta)
    x <- rgamma(n[i],shape=alpha,scale=beta)
    xbar[j] <- mean(x)
  }

# Histogram cells are set to intervals of the form [a,b)
  hist(xbar,breaks=15,prob=T,right=F,main="",xlim=c(min(xbar),max(xbar)),
  mgp=c(2,0.5,0),cex=1.0)
  mtext(side=3,line=1.5,outer=F,paste("n =",n[i]),cex=1.0)

  Npdf <- dnorm(seq(mu-3*sxbar,mu+3*sxbar,0.1),mu,sxbar)
  lines(seq(mu-3*sxbar,mu+3*sxbar,0.1),Npdf,lty=2,col="red")

  if (i==1) mtext(side=3,line=0,outer=F,paste("Gamma: shape = ",alpha,", scale =
",beta,sep=""),cex=1.0)
}
```

A.3 Sampling distribution of *z*-statistic and *t*-statistic from gamma distribution

```
jpeg("C:/Users/plot.jpg",width=480,height=240,quality=100)
par(mfrow=c(1,2),oma=c(0,0,0,0),mar=c(3,3,3,2),bty="n")
set.seed(321)

n <- 20
alpha <- 2  # Gamma(alpha,beta)
beta <- 1

mu <- alpha*beta
sigma <- sqrt(alpha)*beta

r <- 10000  # No. of replications
zstat <- rep(NA,r)
tstat <- rep(NA,r)

for (i in 1:r)
{
# x is a random sample of size n from Gamma(alpha,beta)
  x <- rgamma(n,shape=alpha,scale=beta)

  xbar <- mean(x)     # sample mean
  sx <- sqrt(var(x))  # sample st. dev.

  zstat[i] <- (xbar-mu)/(sigma/sqrt(n))  # z-statistic
  tstat[i] <- (xbar-mu)/(sx/sqrt(n))     # t-statistic
}

# Density histogram of t-statistic, with cell intervals of the form [a,b)
hist(tstat,breaks=seq(min(tstat),max(tstat)+0.1,0.1),prob=T,right=F,
main="",xlim=c(-4.5,4.5),ylim=c(0,0.45),mgp=c(2,0.5,0),cex=0.8)
mtext(side=3,line=1.5,outer=F,"Density histogram of t-statistic",cex=1.0)

# Plot pdf of N(0,1)
zpdf <- dnorm(seq(-4,4,0.1),0,1)
lines(seq(-4,4,0.1),zpdf,lty=2,col="red")
mtext(side=3,line=0,outer=F,paste("n =",n),cex=1.0)

# Normal probability (Q-Q) plot
prob <- (1:r - 0.5)/r
qqx <- qnorm(prob,0,1)
qqy <- quantile(tstat,prob)

plot(qqx,qqy,xlab="N(0,1) quantiles",ylab="tstat quantiles",
xlim=c(-4,4),ylim=c(-4.5,4.5),mgp=c(2,0.5,0),cex=0.8)
mtext(side=3,line=1.5,outer=F,"Normal Q-Q plot",cex=1.0)
```