Assignment 9.b for **STATGR6103**

submitted to Professor Andrew Gelman

Advait Rajagopal

9 November 2016

1 Question 1

Suppose the scalar variable θ is approximately normally distributed in a posterior distribution that is summarized by n independent simulation draws. How large does n have to be so that the 2.5% and 97.5% quantiles of θ are specified to an accuracy of 0.1*sd($\theta|y$)?

1.1 Part A

Figure this out mathematically, without using simulation.[1][2]
If $y_1, ..., y_n$ are independently and identically distributed random variables with a cumulative distribution function F_y, then let $\psi_n(\alpha)$ be a random variable defined for a fixed α in such a way that;

$$\psi_n(\alpha) = \frac{1}{n}\sum_{i=1}^{n} I\{y_i \leq \alpha\} = \frac{1}{n}\sum_{i=1}^{n} Z_i$$

where $Z_i = I\{y_i \leq \alpha\} = 1$ if $y \leq \alpha$ and zero otherwise. Now Z_i has and expected value given by $E(Z_i) = F_y(\alpha)$ and variance $F_y(\alpha) * [1 - F_y(\alpha)]$, thus by the Central Limit Theorem we obtain the result;

$$F_y^{-1}(\psi_n(x)) - F_y^{-1}(F_y(\alpha)) \xrightarrow{d} y \sim \mathcal{N}\left(0, \frac{F_y(\alpha) * [1 - F_y(\alpha)]}{n(f_y(F_y^{-1}(F_y(\alpha))))^2}\right)$$

on simplifying this yields;

$$\hat{\alpha} - \alpha \xrightarrow{d} y \sim \mathcal{N}\left(0, \frac{p(1-p)}{nf_y^2(\alpha)}\right)$$

for $0 < p < 1$ and a large n, where $\hat{\alpha}$ is our estimate of the quantile value and α is the true value of the quantile. The difference between the sample quantile and the true value of the quantile is normally distributed with the mean and variance as described in the equation above where $p = F_y(\alpha)$ and f_y is the probability density function.

My goal is to find out approximately how large n has to be in order to get the value $|\hat{\alpha} - \alpha|$ to be less than 0.1*sd($\theta|y$). Therefore I solve for n as follows.

Given $\alpha_{0.025} = -1.96$ and $\alpha_{0.975} = 1.96$ for the standard normal distribution. We can use the standard normal distribution effectively here because the mean of the distribution of $\hat{\alpha} - \alpha$ is zero and thus standardizing we normalize this distribution to a standard normal. So we draw on the variance of the distribution of $\hat{\alpha} - \alpha$ to find the value of n.

$$\frac{p(1-p)}{nf_y^2(\alpha)} = \frac{0.025 * 0.975}{n * f_y^2(1.96)} = 0.01$$

[1] http://www.math.mcgill.ca/ dstephens/OldCourses/556-2006/Math556-Median.pdf

[2] http://www.utstat.utoronto.ca/keith/papers/subsample.pdf

Therefore

$$\frac{7.136}{n} = 0.01$$
$$n = 714$$

1.2 Part B

Check your answer using simulation and show your results.
I simulate a normal distribution 1000 times of size 714 and calculate the distance between the true value $\alpha_{0.025} = -1.96$ and $\alpha_{0.975} = 1.96$ and calculate the absolute value of the distance between the estimated value of the quantiles (2.5% and 97.5%) and the true value (+/- 1.96). The average accuracy of the distance is approximately 0.08 in both cases which is close to 0.1. I plot a histogram of the accuracy of absolute value of distances below.

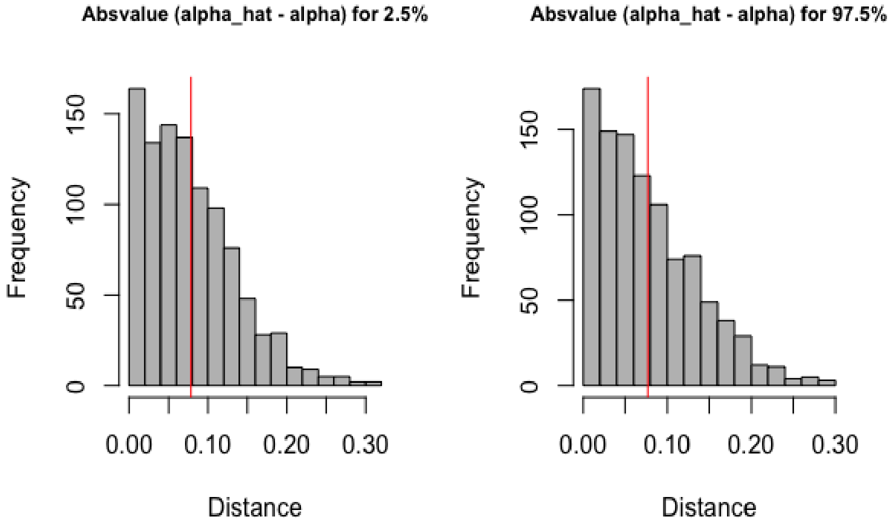

Figure 1: Accuracy of $|\hat{\alpha} - \alpha|$ distances plotted with the average shown by the red line. The average is close to 0.8.

2 Code

2.1 R Code

Listing 1: R Code

```r
f.alpha <-dnorm(qnorm(0.025,0,1), 0, 1)
n <-(0.025*0.975/f.alpha^2)/0.01
sqrt(0.025*0.975/(714*f.alpha^2))
###2-b
set.seed(8)
theta.gen1 <- replicate(1000,rnorm(714))
rep_fun1 <-function(x){
  abs(qnorm(0.025) - quantile(x, c(0.025)))
}
rep1 <- apply(theta.gen1,2,rep_fun1)
mean(rep1)
hist(rep1 )
abline(v = mean(rep1))
#
set.seed(8)
theta.gen2 <- replicate(1000,rnorm(714))
rep_fun2 <-function(x){
  abs(qnorm(0.975) - quantile(x, c(0.975)))
}
rep2 <- apply(theta.gen2,2,rep_fun2)
mean(rep2)
hist(rep2 )
abline(v = mean(rep2))

par(mfcol = c(1,2))
hist(rep1, col = "gray", breaks  = 20,
main = "Absvalue␣(alpha_hat␣-␣alpha)␣for␣2.5\%",
cex.main = 0.8, xlab = "Distance" )
abline(v = mean(rep1), col = "red")
#
hist(rep2, col = "gray", breaks  = 20,
main = "Absvalue␣(alpha_hat␣-␣alpha)␣for␣97.5\%",
cex.main = 0.8 , xlab = "Distance")
abline(v = mean(rep2), col = "red")
```