ST 501 R Project

Bruce Campbell

November 26, 2016

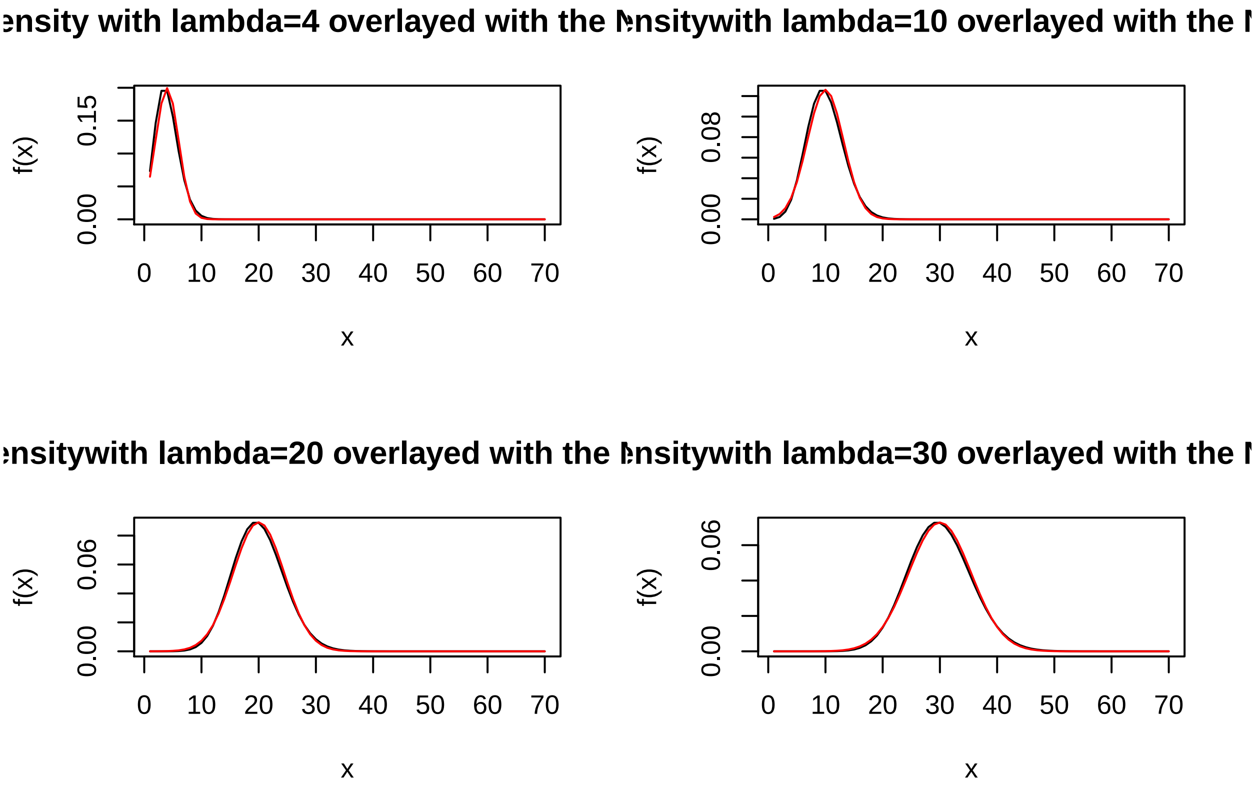
```r
library(knitr)
#Code wrap
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Normal Apporximation to Poisson with large rate parameter

Plot the PMF of a Poisson distribution with mean 4. Overlay the 'large-sample'

normal approximation on this graph.(c) Repeat for a mean of 10, 20, and 30.

```r
par(mfrow = c(2, 2))

x <- seq(from = 1, to = 70, by = 1)
lambdaParameter <- 4
plot(x, y = dpois(x, lambda = lambdaParameter), type = "l", ylab = "f(x)",
    main = "Plot of Poisson density with lambda=4 overlayed with the Normal Approximation")
lines(x, dnorm(x, lambdaParameter, sqrt(lambdaParameter)), type = "l",
    col = "red")


lambdaParameter <- 10
plot(x, y = dpois(x, lambda = lambdaParameter), type = "l", ylab = "f(x)",
    main = "Plot of Poisson densitywith lambda=10 overlayed with the Normal Approximation")
lines(x, dnorm(x, lambdaParameter, sqrt(lambdaParameter)), type = "l",
    col = "red")


lambdaParameter <- 20
plot(x, y = dpois(x, lambda = lambdaParameter), type = "l", ylab = "f(x)",
    main = "Plot of Poisson densitywith lambda=20 overlayed with the Normal Approximation")
lines(x, dnorm(x, lambdaParameter, sqrt(lambdaParameter)), type = "l",
    col = "red")


lambdaParameter <- 30
plot(x, y = dpois(x, lambda = lambdaParameter), type = "l", ylab = "f(x)",
    main = "Plot of Poisson densitywith lambda=30 overlayed with the Normal Approximation")
lines(x, dnorm(x, lambdaParameter, sqrt(lambdaParameter)), type = "l",
    col = "red")
```



For each of the settings for the mean parameter, use R to find $P(Y \geq \lambda + 2\sqrt{\lambda})$ using the Poisson distribution and the normal approximation.

```
parameterData <- c(4, 10, 20, 30)

calculatePoissonProb <- function(lambda) {
    y <- lambda + 2 * sqrt(lambda)
    result <- 1 - dpois(floor(y), lambda = as.integer(lambda))
    return(result)
}

calculateNormalProb <- function(lambda) {
    y <- lambda + 2 * sqrt(lambda)
    result <- 1 - dnorm(y, mean = lambda, sd = sqrt(lambda))
    return(result)
}

DF <- data.frame(lambda = as.numeric(), poisson = as.numeric(),
    normal = as.numeric())

for (i in 1:4) {
    parameterValue <- parameterData[i]
    row = c(parameterValue, calculatePoissonProb(parameterValue),
        calculateNormalProb(parameterValue))
```

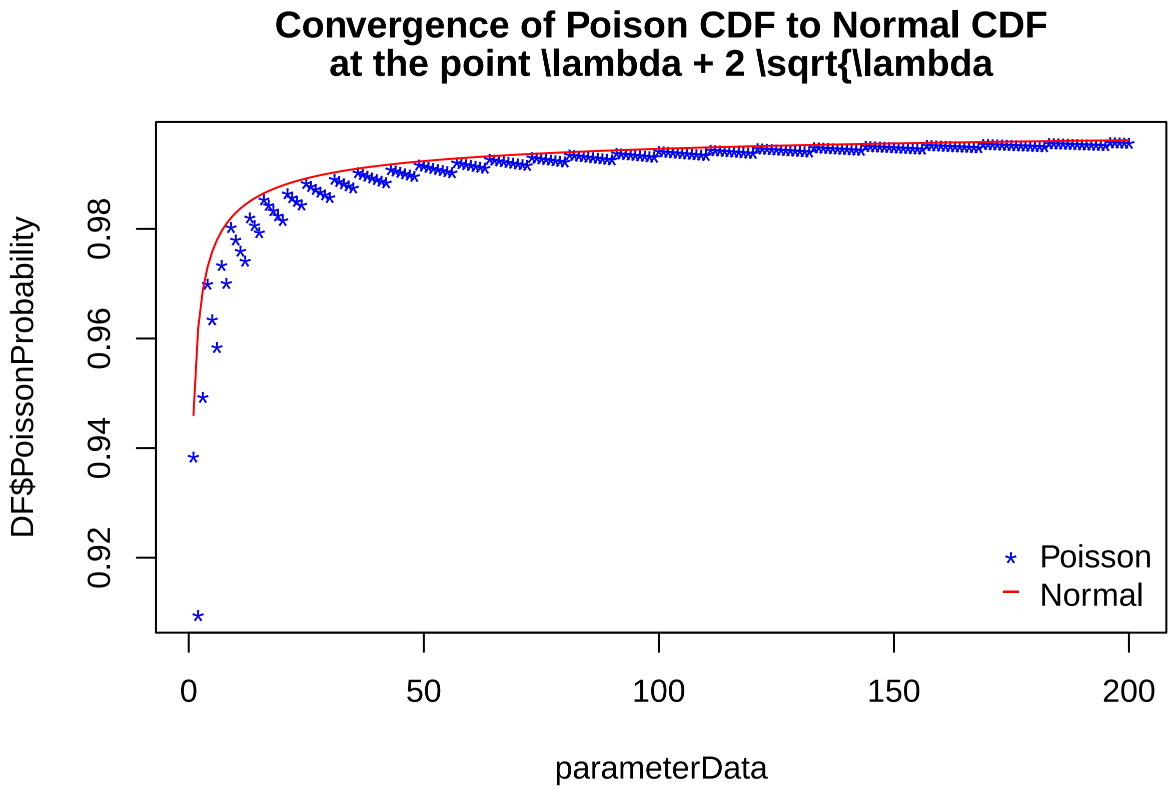
```r
    DF <- rbind(DF, row)
}

library(pander)
names(DF) <- c("Parameter", "PoissonProbability", "NormalProbability")
pander(DF, caption = "Comparing Poisson versus Normal Probabilities")
```

Table 1: Comparing Poisson versus Normal Probabilities

Parameter	PoissonProbability	NormalProbability
4	0.9702	0.973
10	0.9783	0.9829
20	0.9819	0.9879
30	0.9861	0.9901

Create a single plot that graphs the both of these probabilities for values of ranging from 1 to 200

```r
parameterData <- seq(1, 200, 1)

DF <- data.frame(lambda = as.numeric(), poisson = as.numeric(),
    normal = as.numeric(), y = as.numeric())

for (i in 1:length(parameterData)) {
    parameterValue <- as.numeric(parameterData[i])
    y <- y <- parameterValue + 2 * sqrt(parameterValue)
    row = c(parameterValue, calculatePoissonProb(parameterValue),
        calculateNormalProb(parameterValue), y)
    DF <- rbind(DF, row)
}

names(DF) <- c("Parameter", "PoissonProbability", "NormalProbability")

plot(parameterData, DF$PoissonProbability, col = "blue", pch = "*")

lines(parameterData, DF$NormalProbability, col = "red")

# title(c('Convergence of Poison CDF to Normal CDF at the
# point ',expression('\\lambda + 2 \\sqrt{\\lambda')))
title(c("Convergence of Poison CDF to Normal CDF", "at the point \\lambda + 2 \\sqrt{\\lambda"))

legend("bottomright", pch = c("*", "-"), col = c("blue", "red"),
    bty = "n", legend = c("Poisson", "Normal"))
```

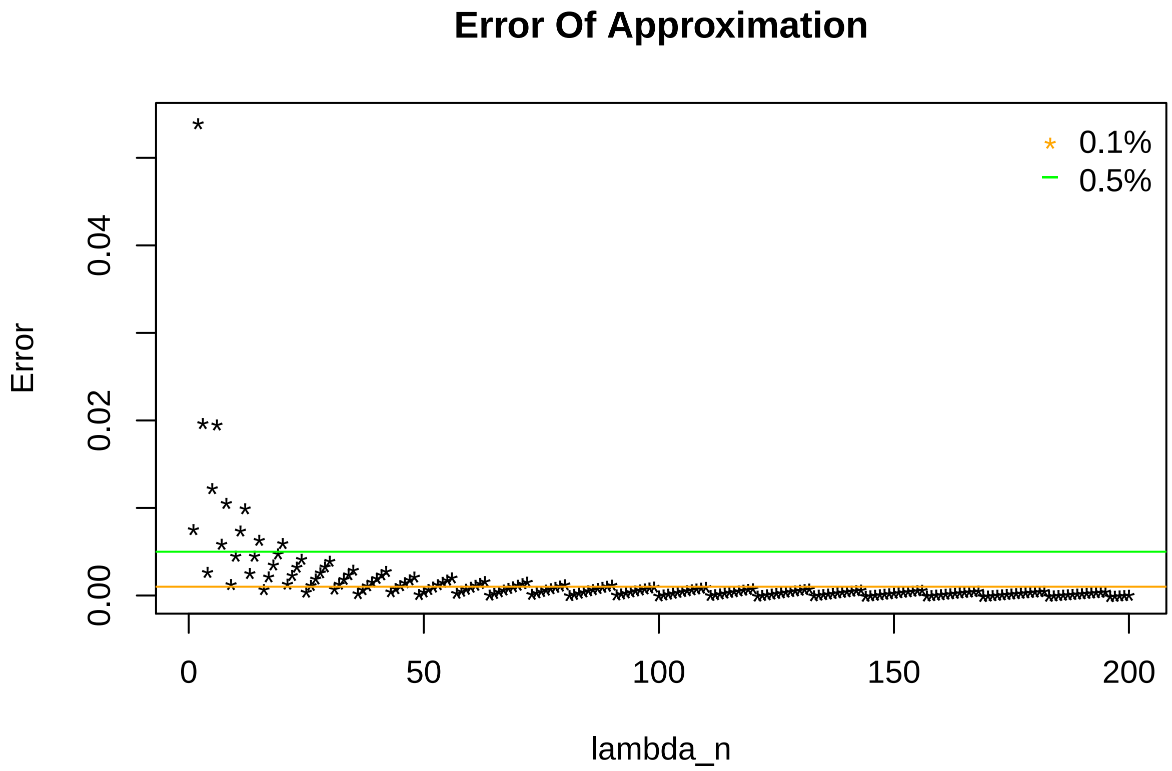


f) What concept we are investigating with the above question/graphs?

We are investigating the Central Limit Theorem. Let $F_n(y)$ be the CDF of $Poisson(\lambda_n)$ we're looking at $F_n \to Normal(\lambda, \sqrt(\lambda))$ as $\lambda_n \to \infty$ The convergence is demonstrated for a different point at each n, so this is not a visualization of pointwise convergence at a fixed y.

g) Recommendations for using the normal approximation to the poisson

```r
plot((DF$NormalProbability - DF$PoissonProbability)/DF$NormalProbability,
    pch = "*", xlab = "lambda_n", ylab = "Error", main = "Error Of Approximation")
abline(h = 0.001, col = "orange")
abline(h = 0.005, col = "green")


legend("topright", pch = c("*", "-"), col = c("orange", "green"),
    bty = "n", legend = c("0.1%", "0.5%"))
```



We see that the error is about 0.1% above 60 and that the error is less than 0.2% around $\lambda_n = 25$.

Estimating the mean of Weibull distribution.

Generate N = 50000 data sets of size n = 5 from a Weibull distribution with a = 0.75 and b = 1. For each sample, calculate the observed sample mean. Save these sample means for later use. Repeat this process for n = 15 and n = 30.

```
numberOfSamples <- 50000
a <- 0.75
b <- 1

sampleSize <- 5

sampleMeans5 <- matrix(data = NA, nrow = numberOfSamples, ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(5, shape = a, scale = b)
    sampleMeans5[i] <- mean(sample)
}



sampleSize <- 15
```

```
sampleMeans15 <- matrix(data = NA, nrow = numberOfSamples, ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(15, shape = a, scale = b)
    sampleMeans15[i] <- mean(sample)
}

sampleSize <- 30

sampleMeans30 <- matrix(data = NA, nrow = numberOfSamples, ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(30, shape = a, scale = b)
    sampleMeans30[i] <- mean(sample)
}
```

Repeat the above two steps except generate the data using a = 1.75

```
a <- 1.75
b <- 1

sampleSize <- 5

sampleMeans5a2 <- matrix(data = NA, nrow = numberOfSamples, ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(5, shape = a, scale = 1)
    sampleMeans5a2[i] <- mean(sample)
}

sampleSize <- 15

sampleMeans15a2 <- matrix(data = NA, nrow = numberOfSamples,
    ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(15, shape = a, scale = 1)
    sampleMeans15a2[i] <- mean(sample)
}

sampleSize <- 30

sampleMeans30a2 <- matrix(data = NA, nrow = numberOfSamples,
    ncol = 1)
for (i in 1:numberOfSamples) {
    sample <- rweibull(30, shape = a, scale = 1)
    sampleMeans30a2[i] <- mean(sample)
}
```

histogram of the sample mean values with an overlay the normal approximation to the distribution given by the CLT.

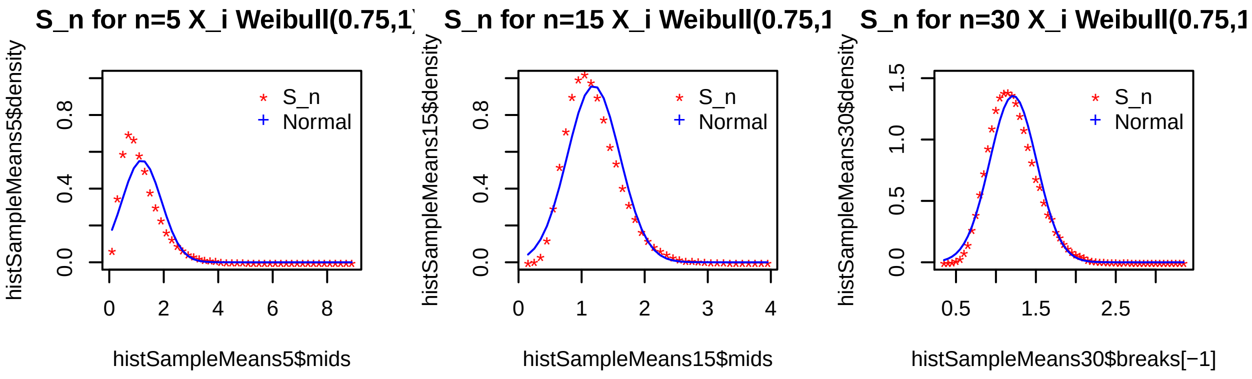
To do this we will need the mean and variance of the Weibull $E(X) = b\Gamma(1 + 1/a)$ $Var(X) = b^2 * (\Gamma(1 + 2/a) - (\Gamma(1 + 1/a))^2)$

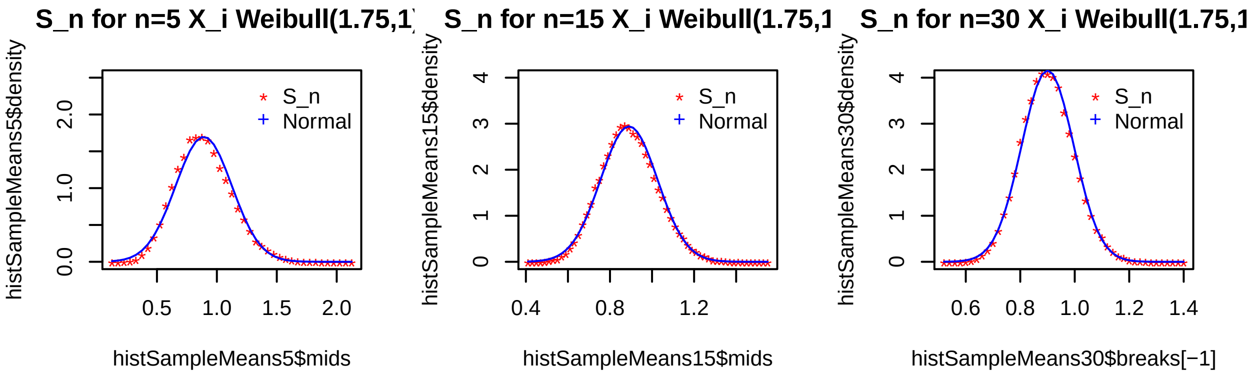
```r
par(mfrow = c(2, 3))

mean_a1 <- gamma(1 + 1/0.75)
var_a1 <- b^2 * (gamma(1 + 2/0.75) - gamma(1 + 1/0.75)^2)
sd_a_1 <- sqrt(var_a1)

# Sanity check above install.packages('mixdist')
library(mixdist)
```

```
## Warning: package 'mixdist' was built under R version 3.2.5
```

```r
weibullParams <- weibullparinv(0.75, 1)

histSampleMeans5 <- hist(sampleMeans5, plot = FALSE, 50)
plot(histSampleMeans5$mids, histSampleMeans5$density, col = "red",
    pch = "*", ylim = c(0, 1))
lines(histSampleMeans5$mids, dnorm(histSampleMeans5$mids, mean = mean_a1,
    sd = sqrt(var_a1)/sqrt(5)), col = "blue", pch = "+")
title(c("S_n for n=5 X_i Weibull(0.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))

histSampleMeans15 <- hist(sampleMeans15, plot = FALSE, 50)
plot(histSampleMeans15$mids, histSampleMeans15$density, col = "red",
    pch = "*", ylim = c(0, 1))
lines(histSampleMeans15$mids, dnorm(histSampleMeans15$mids, mean = mean_a1,
    sd = sqrt(var_a1)/sqrt(15)), col = "blue", pch = "+")
title(c("S_n for n=15 X_i Weibull(0.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))

histSampleMeans30 <- hist(sampleMeans30, plot = FALSE, 50)
plot(histSampleMeans30$breaks[-1], histSampleMeans30$density,
    col = "red", pch = "*", ylim = c(0, 1.5))
lines(histSampleMeans30$breaks[-1], dnorm(histSampleMeans30$mids,
    mean = weibullParams$mu, sd = weibullParams$sigma/sqrt(30)),
    col = "blue", pch = "+")
title(c("S_n for n=30 X_i Weibull(0.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))


mean_a2 <- gamma(1 + 1/1.75)
var_a2 <- b^2 * (gamma(1 + 2/1.75) - gamma(1 + 1/1.75)^2)
sd_a_2 <- sqrt(var_a1)

# Sanity check above
weibullParams <- weibullparinv(1.75, 1)


histSampleMeans5 <- hist(sampleMeans5a2, plot = FALSE, 50)
plot(histSampleMeans5$mids, histSampleMeans5$density, col = "red",
    pch = "*", ylim = c(0, 2.5))
```

```r
lines(histSampleMeans5$mids, dnorm(histSampleMeans5$mids, mean = mean_a2,
    sd = sqrt(var_a2)/sqrt(5)), col = "blue", pch = "+")
title(c("S_n for n=5 X_i Weibull(1.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))

histSampleMeans15 <- hist(sampleMeans15a2, plot = FALSE, 50)
plot(histSampleMeans15$mids, histSampleMeans15$density, col = "red",
    pch = "*", ylim = c(0, 4))
lines(histSampleMeans15$mids, dnorm(histSampleMeans15$mids, mean = mean_a2,
    sd = sqrt(var_a2)/sqrt(15)), col = "blue", pch = "+")
title(c("S_n for n=15 X_i Weibull(1.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))


histSampleMeans30 <- hist(sampleMeans30a2, plot = FALSE, 50)
plot(histSampleMeans30$breaks[-1], histSampleMeans30$density,
    col = "red", pch = "*", ylim = c(0, 4))
lines(histSampleMeans30$breaks[-1], dnorm(histSampleMeans30$mids,
    mean = mean_a2, sd = sqrt(var_a2)/sqrt(30)), col = "blue",
    pch = "+")
title(c("S_n for n=30 X_i Weibull(1.75,1) "))
legend("topright", pch = c("*", "+"), col = c("red", "blue"),
    bty = "n", legend = c("S_n", "Normal"))
```





Discussion

Wat we're seeing in the plot above is evidence of the convergence in distribution of the sampling distribution of the mean for the Weibull distribution. 2 shape parameters are presented with three different sample sizes for S_n. S_n is the sum of a set of 50000 sample means. As $n \to \infty$ we expect convergence of the normalized sum to the Standard Normal distribution.

Estimate the probability the sample mean is more than 2 standard deviations greater than the population mean.

We calculate the proportion of samples above the exact values

```r
rho <- mean_a1 + 2 * sqrt(var_a1/5)
proportion_A1_5 <- sum(sampleMeans5 > rho)/numberOfSamples

rho <- mean_a1 + 2 * sqrt(var_a1/15)
proportion_A1_15 <- sum(sampleMeans15 > rho)/numberOfSamples

rho <- mean_a1 + 2 * sqrt(var_a1/30)
proportion_A1_30 <- sum(sampleMeans30 > rho)/numberOfSamples

rho <- mean_a2 + 2 * sqrt(var_a2/5)
proportion_A2_5 <- sum(sampleMeans5a2 > rho)/numberOfSamples

rho <- mean_a2 + 2 * sqrt(var_a2/15)
proportion_A2_15 <- sum(sampleMeans15a2 > rho)/numberOfSamples

rho <- mean_a2 + 2 * sqrt(var_a2/30)
proportion_A2_30 <- sum(sampleMeans30a2 > rho)/numberOfSamples

library(pander)
pander(data.frame(samplingDistribution = c("Scale 0.75 n=5",
    "Scale 0.75 n=15", "Scale 0.75 n=30", "Scale 1.75 n=5", "Scale 1.75 n=15",
    "Scale 1.75 n=30"), estimate = c(proportion_A1_5, proportion_A1_15,
    proportion_A1_30, proportion_A2_5, proportion_A2_15, proportion_A2_30)),
    caption = "Estimate of the proportion greatther than 2 standard deviations to the right of the mean
```

Table 2: Estimate of the proportion greatther than 2 standard deviations to the right of the mean

samplingDistribution	estimate
Scale 0.75 n=5	0.0457
Scale 0.75 n=15	0.03898
Scale 0.75 n=30	0.03612
Scale 1.75 n=5	0.03188
Scale 1.75 n=15	0.02752
Scale 1.75 n=30	0.0276

$(1 - pnorm(2))/2$ should approximate the probabilities above. This is the probability of a realization of the sampling distribution of the mean to be greater than 2 standard deviations away from the true value.

```r
(1 - pnorm(2))/2
```

```
## [1] 0.01137507
```

We see that with a larger α shape parameter we have a better approximation to the limiting value. Also, for a fixed shape parameter we have a better approximation as the sample size for the mean increases. This is expected.

To come up with a general rule of thumb we'd want to run a larger experiment and set some performance parameters expressed as a function of the shape and rate parameters. Both parameters α and λ of the Weibull distribution appear in the variance, hence both appear in the standard deviation σ. The CLT describes convergence in distribution of a sum scaled by $\frac{1}{\sigma}$. We would run a series of simulations with a large number of sample means.

The following parameters would be used - n the number of samples in each calculation of a sample mean - α the scale parameter - λ the shape parameter

We'd create a nested loop covering a good range of these parameter values and for each we'd calculate the some measure of the fit of the empirical distribution to the the standard normal. There are several ways to do calculate the fit - Anderson Darling - Kolmogorov Smirnov

With the fit data we'd first visualize the fit over the three dimensional parameter space and then try to make some general statements about how the fit is better with larger n or larger α. We'd also think about how we might be able to formaulate a more precise staement in terms of the results of Kolmogorov-Smirnov or Anderson-Darling fit values.