Lab 6: Central Limit Theorem and Confidence Intervals

August 2, 2016

1 Goals

Today we will be using simulations to

- Examine convergence properties of the central limit theorem
- Examine coverage rates of confidence intervals for proportions
- Other multipliers for confidence levels not .95

2 Central Limit Theorem

We know from the Central Limit theorem, that $\bar{x}$ becomes more and more like a $\mathcal{N}(\mu, \sigma^2/n)$ as n increases. However, how "close" the distribution of $\bar{x}$ is to the normal distribution for any given n can vary widely depending on the underlying distribution of each X_i. Let's take a look at a few different types of distributions.

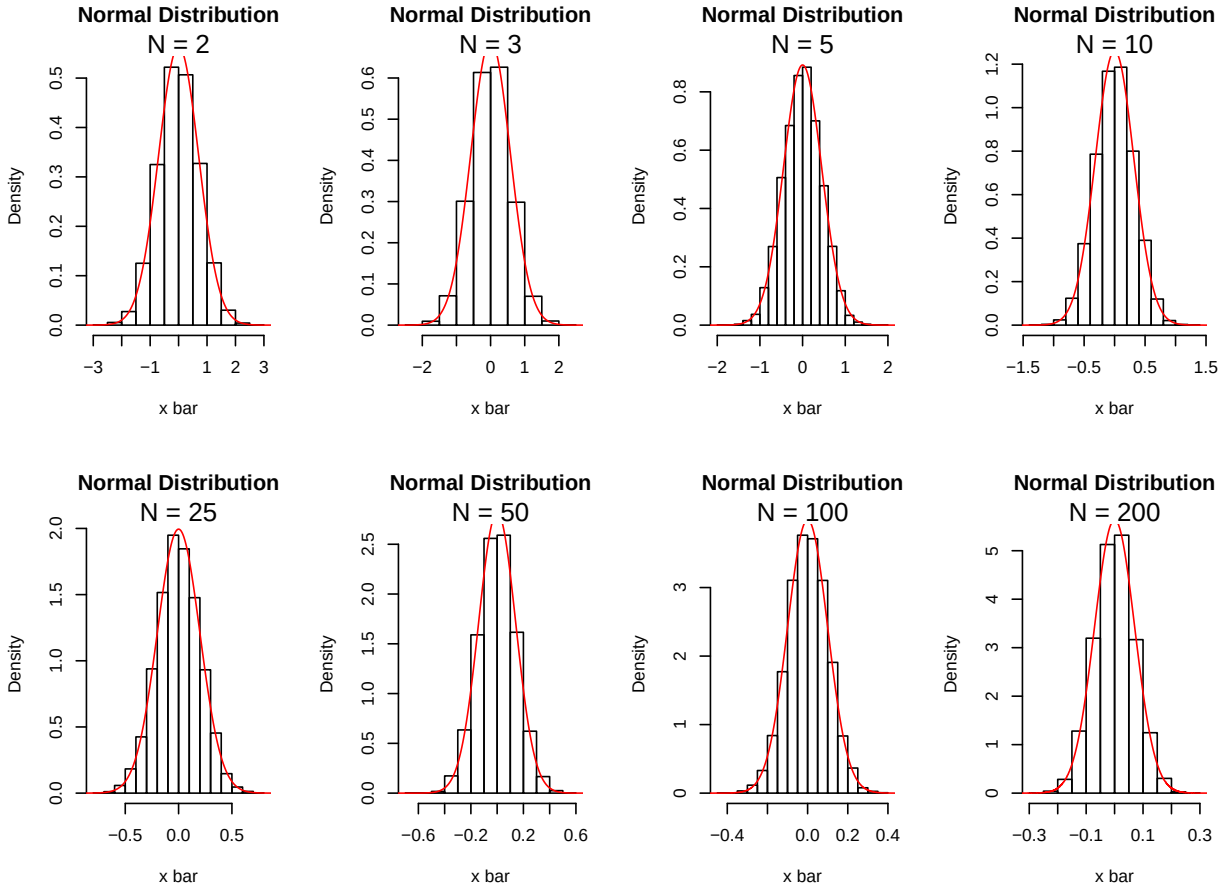
- Normal distribution
- Exponential Distribution
- Uniform Distribution
- Poisson Distribution
- T-Distribution

2.1 Normal Distribution

When we start out with each individual variable being normally distributed, we automatically have $\bar{x}_n$ being normally distirbuted with even $n = 1$. This is because the sum (and average) of normally distrbuted variables is still normally distributed. Let's look at how closely the distribution of $\bar{x}$ follows what we would expect from the Central Limit Theorem.

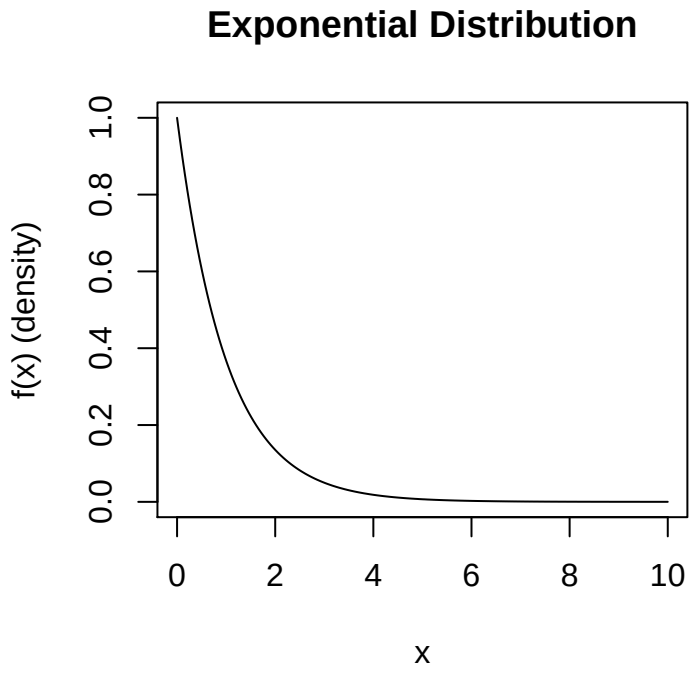
```
# arrange the plots in 2 rows with 4 columns
par(mfrow = c(2, 4))

#### Change this code when you use a different distribution ###
mu <- 0
sigma <- 1
title <- "Normal Distribution"
##################################################################
```

```r
n.list <- c(2, 3, 5, 10, 25, 50, 100, 200)
for(n in n.list){
  rec <- rep(0, 10000)
  for(i in 1:10000){

    # Change this function when you use a different distribution
    rec[i] <- mean(rnorm(n = n, mu, sigma))
  }
  hist(rec, main = title, freq = F, xlab = "x bar")
  mtext(paste("N = ", n, sep = ""))


  lines(seq(-5,5, by = 1/(n * 100)), dnorm(seq(-5,5, by = 1/(n * 100)), mean = mu,
  sd = sigma / sqrt(n)), col = "red")

}
```



2.2 Exponential Distribution

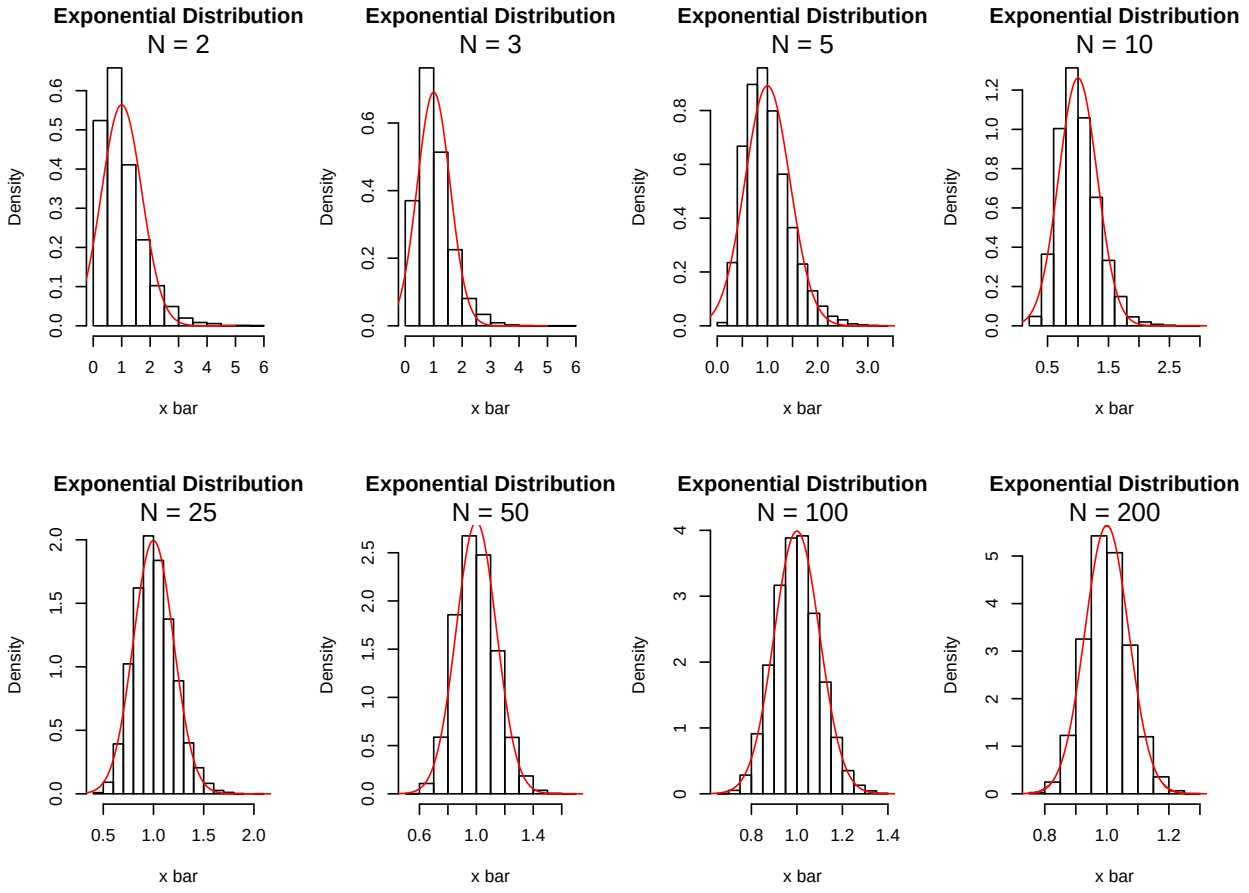
Let's examine an Exponential Distribution with $\lambda = 1$ (in R lambda is known as the rate parameter). We know that the Exponential Distribution is skewed right, so it is pretty far off from the normal distribution.



Let's take a look at how quickly the exponential distribution converges to the normal distribution specified by the Central Limit Theorem. Note that this is the same code as before. However, the mean of the exponential distribution becomes $\mu = 1/\lambda$ and $\sigma = 1/\lambda$ and we now are using `rexp`.

```
# arrange the plots in 2 rows with 4 columns
par(mfrow = c(2, 4))


#### Change this code when you use a different distribution ###
lambda <- 1
sigma <- 1 / lambda
mu <- 1 / lambda
title <- "Exponential Distribution"
###############################################################


n.list <- c(2, 3, 5, 10, 25, 50, 100, 200)
for(n in n.list){
  rec <- rep(0, 10000)
  for(i in 1:10000){

    # Change this function when you use a different distribution
    rec[i] <- mean(rexp(n = n, rate = lambda))
  }
  hist(rec, main = title, freq = F, xlab = "x bar")
  mtext(paste("N = ", n, sep = ""))


  lines(seq(-5,5, by = 1/(n * 100)),
        dnorm(seq(-5,5, by = 1/(n * 100)), mean = mu,
```

```
                sd = sigma / sqrt(n)), col = "red")
}
```

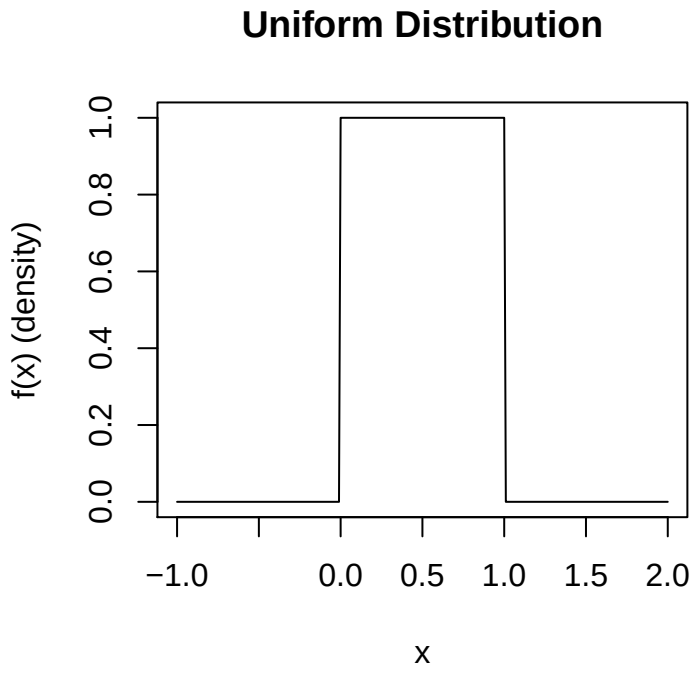


2.2.1 Questions

- At what size of n does the distribution start to look relatively normal?
- How does this compare to the previous distributions?

2.3 Uniform Distribution

Let's examine a uniform Distribution with min = 0 and max = 1 We know that the uniform Distribution is completely flat, so it is pretty far off from the normal distribution.



Let's take a look at how quickly the uniform distribution converges to the normal distribution specified by the Central Limit Theorem. Note that this is the same code as before. However, the mean of the uniform distribution becomes $\mu = 1/2$, the standard deviation $\frac{1}{\sqrt{12}}$ and we now are using `runif(n = n, min = 0, max = 1)`. Modify the code from above to produce the same plots for a uniform distribution.

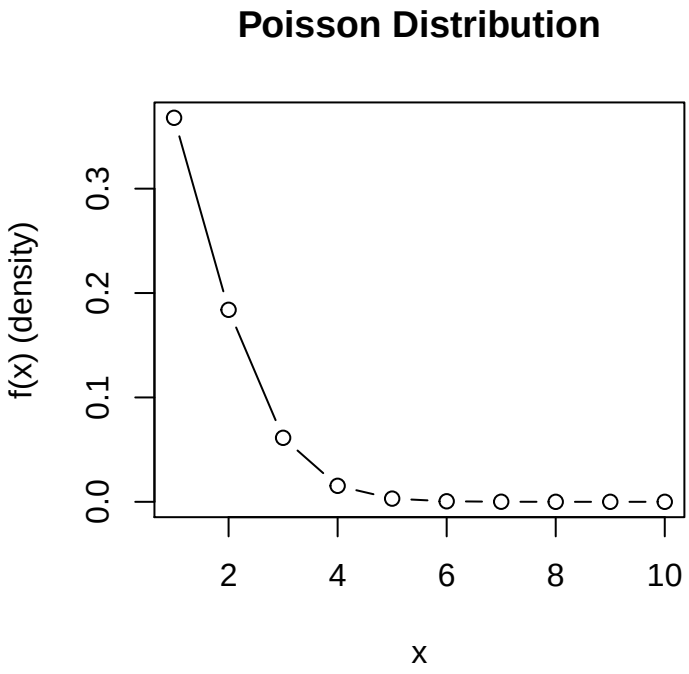
2.3.1 Questions

- At what size of n does the distribution start to look relatively normal?
- How does this compare to the previous distributions?

We can see that the uniform distribution converges quite quickly to a normal distribution. In fact, many many years ago, when computers had a tough time generating normal random variables, they used to simply take 12 uniform random variables (which were easier to generate) and then take the average.

2.4 Poisson Distribution

Let's examine a poisson istribution with $\lambda = 1$. We know that the Poisson Distribution is discrete and skewed, so it is pretty far off from the normal distribution.



Let's take a look at how quickly the Poisson distribution converges to the normal distribution specified by the Central Limit Theorem. Note that this is the same code as before. However, the mean of the poisson distribution has $\mu = \lambda$, the standard deviation of $\sigma = \sqrt{\lambda}$ and we now are using `rpois(n = n, lambda = 1)`. Modify the code from above to produce the same plots for a poisson distribution.

2.4.1 Questions

- At what size of n does the distribution start to look relatively normal?
- Does the fact that the Poisson is discrete affect the Central Limit Theorem?
- How does this compare to the previous distributions?

2.5 T Distribution

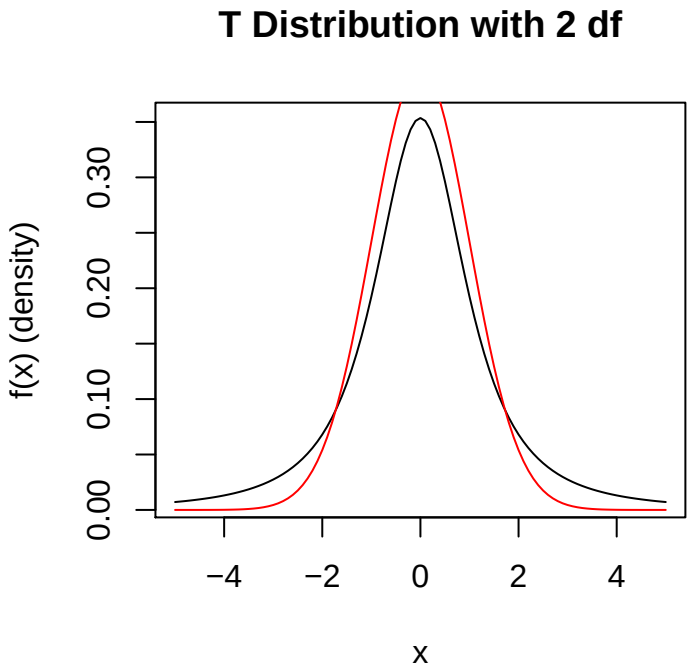
Finally, let's examine a new distribution, the T distribution. The T distribution does not look too different than the normal distribution. It is symmetric and is roughly bell shaped. The T distribution has a single parameter, called the degrees of freedom (or df).

We can form a random variable Y with a T distribution and d degrees of freedom in the following manner-

1. Let Z be a standard normal random variable
2. Let X_i be a standard normal random variable for $i = 1, \dots d$
3. $Y = \frac{Z}{\sqrt{\frac{\sum_i^d X_i^2}{d}}}$

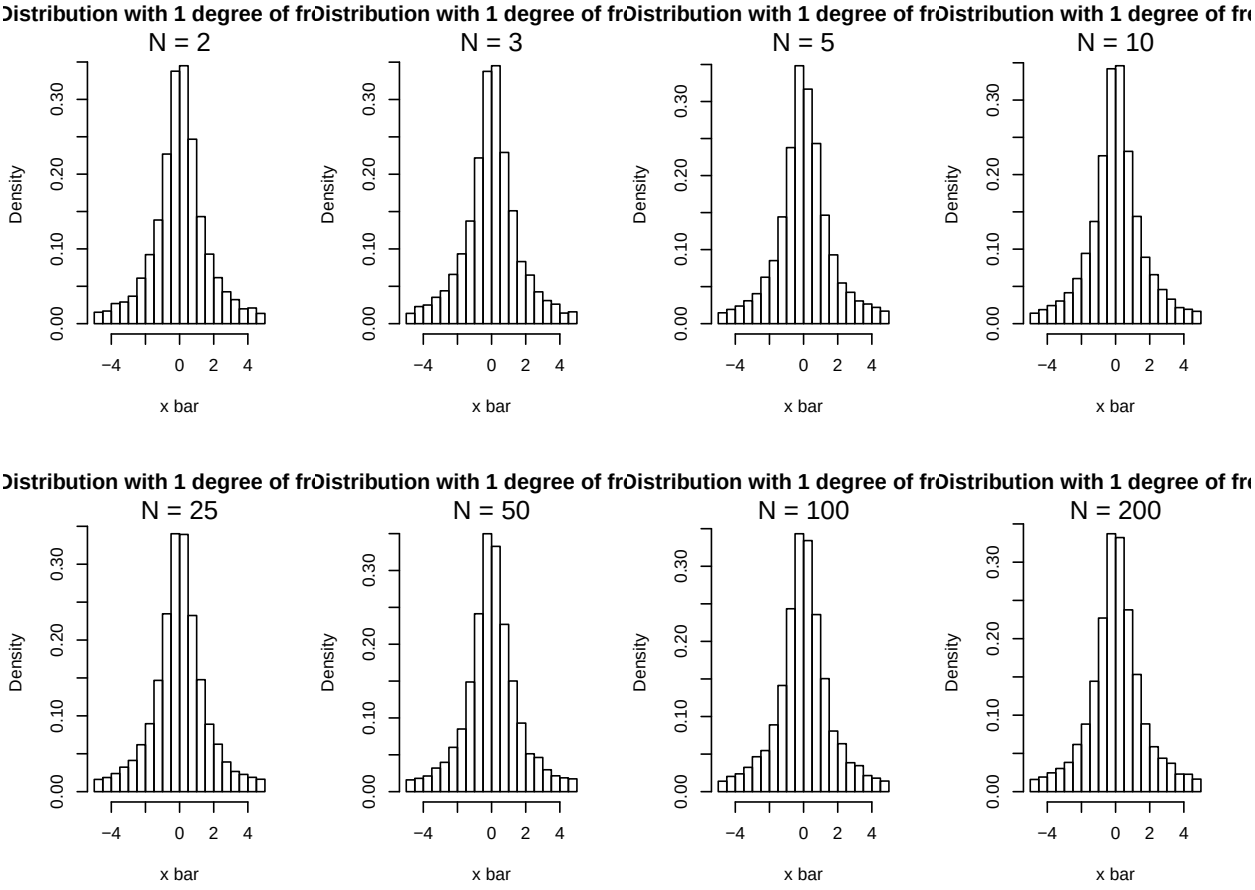
We often denote this distribution as t_d where d is the degrees of freedom. It turns out that for $d = 1$, Y has no mean or variance and for $d = 2$, Y has a mean of 0, but no variance. More technically, we know

that the integrals which define the variance (and mean for $d = 1$) diverge. Below we plot the density of a t distirbution with 2 degrees of freedom in black and a standard normal in red.



Let's take a look at whether the T distribution still converges to the normal distribution specified by the Central Limit Theorem. Use the code from below to see. Note that we can simulate values from a T_1 using `rt(n=n, df = 1)`

```
# arrange the plots in 2 rows with 4 columns
par(mfrow = c(2, 4))

# standard deviation of the underlying population
title <- "T Distribution with 1 degree of freedom"

n.list <- c(2, 3, 5, 10, 25, 50, 100, 200)
for(n in n.list){
  rec <- rep(0, 10000)
  for(i in 1:10000){

    # changed the function
    rec[i] <- mean(rt(n = n, df = 1))
  }
  hist(rec[rec < 5 & rec > -5], main = title, freq = F, xlab = "x bar",
       xlim = c(-5, 5))
  mtext(paste("N = ", n, sep = ""))

}
```



2.6 Questions

- At what size of n does the distribution start to look relatively normal?
- Does the variance of $\bar{x}$ seem to shrink as n increases?
- How does this compare to the previous distributions?

The CLT says that $\bar{x}$ goes to a normal distribution with mean of μ and variance σ^2/n where μ and σ^2 are the mean and variance of each individual in the sample. But in this case, since each individual in the sample does not have a proper mean or variance, things don't quite work out. This is a rare case of when the central limit theorem doesn't work out. Let's try this again with more degrees of freedom. Let $df = 5$ and try the code above again. In this case, set $\mu = 0$ and $\sigma = \sqrt{\frac{5}{3}}$.

When the degrees of freedom is greater than 2, we have both a mean and variance, so the CLT works in this case.

3 Confidence Intervals

Yesterday in class we saw that the coverage rate of 25 coin tosses wasn't quite 95%. The reason for this is because the confidence interval we formed is based on the CLT, which requires n to be large. In this case, it seems like n wasn't large enough. It turns out that how large n has to be also depends somewhat on the value of p.

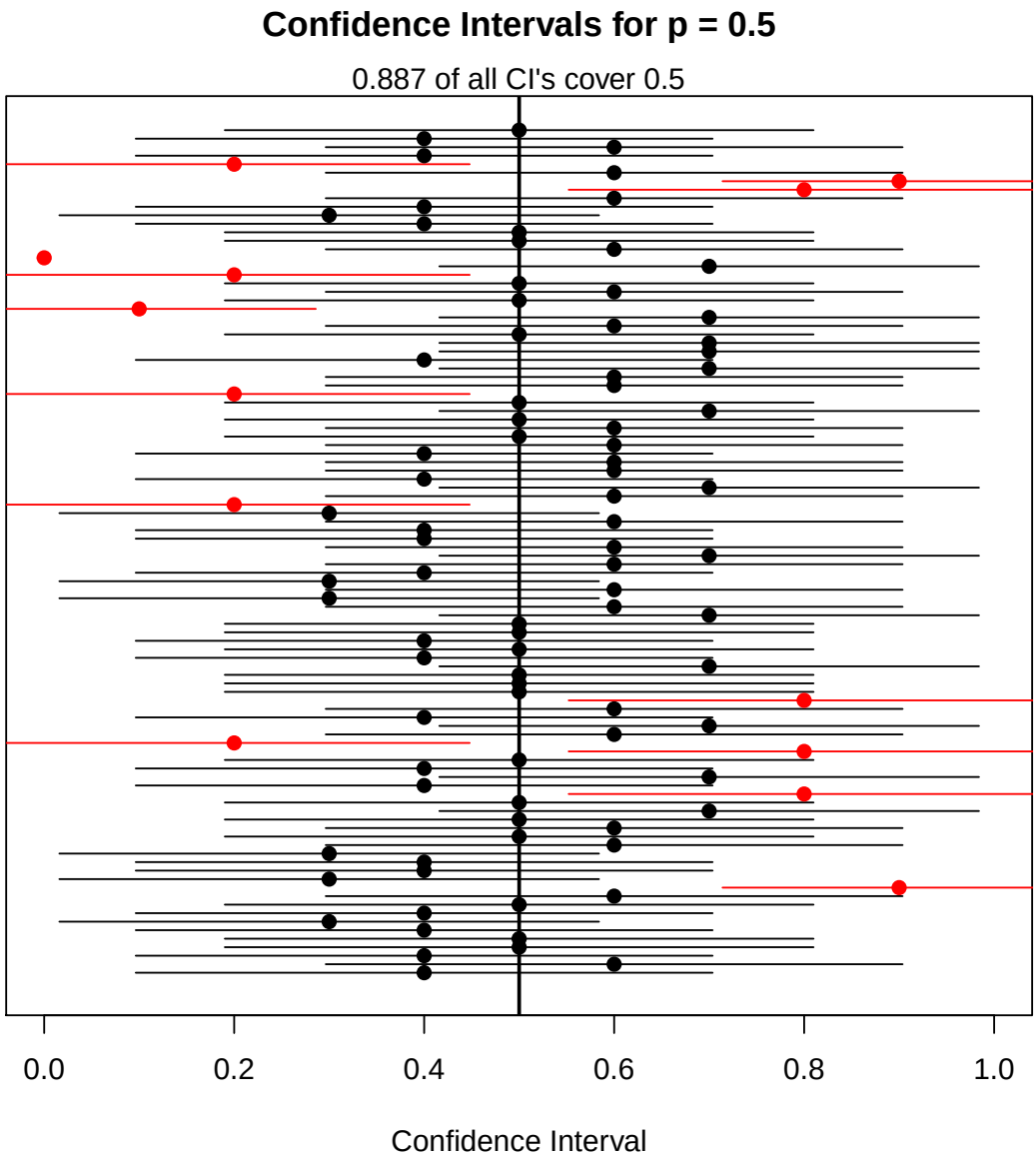
Below the code simulates 10000 samples from a binomial with specified n and p. It then calculates $\hat{p}$ by dividing the outcome of the binomial by $\hat{p}$ and constructs a confidence interval for each sample. We can then check to see whether each confidence interval contains the true parameter p or not. The percentange of the confidence intervals we created which actually contain the true parameter is also known as the **coverage rate**. Note that ideally the coverage rate and the confidence level would be the same, but as we saw in class the coverage rate can be much smaller than the confidence level if n is not big enough.

The code below only displays 100 of the 10000 confidence intervals because plotting all 10000 would be hard to visualize. The coverage rate displayed, however, accounts for all 10000 samples.

3.1 Question

- Run the code below and see how the coverage rate changes as n increases.
- Change p and rerun your experiments. Does p affect your coverage rate? Try extreme values of p, some very close to 1 or 0.
- How does the "extremeness" of p affect how large n has to be for accurate coverage?

```
### Change these ####
n <- 10
p <- .5

####################

mu <- p
sigma <-sqrt(p*(1-p))

# simulate binomials, then divide by n to get a p.hat
p.hat <- rbinom(n = 10000, size = n, prob = p)/ n

# Forming the lower nad upper
lower <- p.hat - 1.96 * sqrt(p.hat * (1-p.hat)/ n)
upper <- p.hat + 1.96 * sqrt(p.hat * (1-p.hat)/ n)

# Check to see if the confidence intervals cover the parameter
CI.cover <- (lower < p) & (upper > p)

# Randomly choose to plot 100 of the samples and CI's
random.plot <- sample(1:10000, 100)
plot(-1, -1, xlim = c(0,1), ylim = c(0, 100), ylab = "",
     yaxt = "n", xlab = "Confidence Interval",
     main = paste("Confidence Intervals for p = ", p, sep = ""))
abline(v = p, col = "black", lwd = 2)

segments(lower[random.plot], c(1:100),
         upper[random.plot], c(1:100),
         col = ifelse(CI.cover[random.plot], "black", "red"))

points(p.hat[random.plot],
       c(1:100), col = ifelse(CI.cover[random.plot], "black", "red"),
       pch = 19)
mtext(paste(round(mean(CI.cover), 3),
            " of all CI's cover ", p, sep = ""))
```



4 CI Multipliers

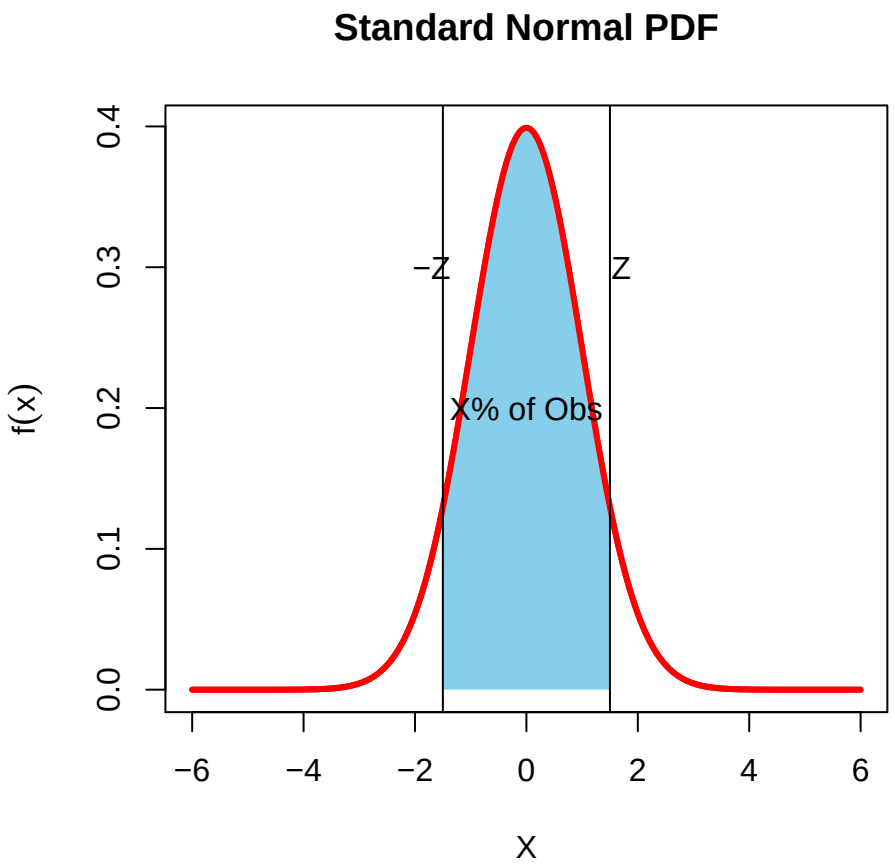
Recall the derivation of the Confidence Interval which started with a statement like

$$.95 = P(p - 1.96\sigma \leq \hat{p} \leq p + 1.96\sigma)$$

This implicitly assumes that we want a 95% confidence interval, so we use the multiplier of 1.96. What if we don't want a 95% confidence interval. We can easily look up the appropriate multiplier using R.

For an arbitrary X% confidence interval, we need to see what value of Z is such that when $\hat{p} \sim \mathcal{N}(p, \sigma^2)$,

$$X = P(p - Z\sigma \leq \hat{p} \leq p + Z\sigma)$$



If we want X% in between the Z's, then we need to find a Z such that (1- X)/2 is less than -Z. In particular, we can use the `qnorm` function to find the value of -Z which satisfies that property. If you don't remember the qnorm function, take a look back at the lab from last Thursday for a quick refresh.

```
confidence.level <- .95

# this is -Z
qnorm((1- confidence.level)/2)
```

```
## [1] -1.959964
```

5 Question

- What is the multiplier for a confidence level of 80%?
- What is the multiplier for a confidence level of 99.99%?
- What is the multiplier for a confidence level of 50%?