

Monte Carlo Inference

Su2016 - Dr. Junvie Pailden

July 21, 2016

Contents

Why the need for simulation?	1
Monte Carlo methods for estimation	1
Estimating the Confidence Level	6
Hypothesis Tests	13
Empirical Type I error rate	14
Power of a Test	16

Why the need for simulation?

- In statistical inference there is uncertainty in an estimate.
- The methods covered in this topic used repeated sampling from a given probability model.
- We also call this methods *parameteric bootstrap*.
- If we can simulate the stochastic process that generated our data, repeatedly drawing samples under identical conditions, then ultimately we hope to have a close replica of the process itself reflected in the samples.

Monte Carlo methods for estimation

Suppose X_1, \dots, X_n is a random sample from the distribution of X . An estimator of $\hat{\theta}$ for a parameter θ is an n variate function

$$\hat{\theta} = \hat{\theta}(X_1, \dots, X_n)$$

of the sample.

Random variates from the sampling distribution of $\hat{\theta}$ can be generated by repeatedly drawing independent random samples $x^{(j)}$ and computing

$$\hat{\theta}^{(j)} = \hat{\theta}(x_1^{(j)}, \dots, x_n^{(j)})$$

for each sample.

The estimator $\hat{\theta}$ of θ can be written as $\hat{\theta} = \frac{1}{m} \sum_{j=1}^m \hat{\theta}^{(j)}$.

The Monte Carlo estimate of $MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$ is

$$\hat{MSE} = \frac{1}{m} \sum_{j=1}^m (\hat{\theta}^{(j)} - \hat{\theta})^2.$$

The corresponding estimate of the standard error of $\hat{\theta}$ is

$$\hat{se}(\hat{\theta}) = \frac{1}{\sqrt{m}} \left\{ \frac{1}{m} \sum_{j=1}^m (\hat{\theta}^{(j)} - \hat{\theta})^2 \right\}^{1/2} = \frac{1}{m} \left\{ \sum_{j=1}^m (\hat{\theta}^{(j)} - \hat{\theta})^2 \right\}^{1/2}.$$

Example 1: Expected absolute distance

Suppose that X_1, X_2 are IID from χ_1^2 which is a chi-square distribution with 1 degree of freedom. Estimate the mean difference

$$\theta = E[g(X_1, X_2)] = E|X_1 - X_2|.$$

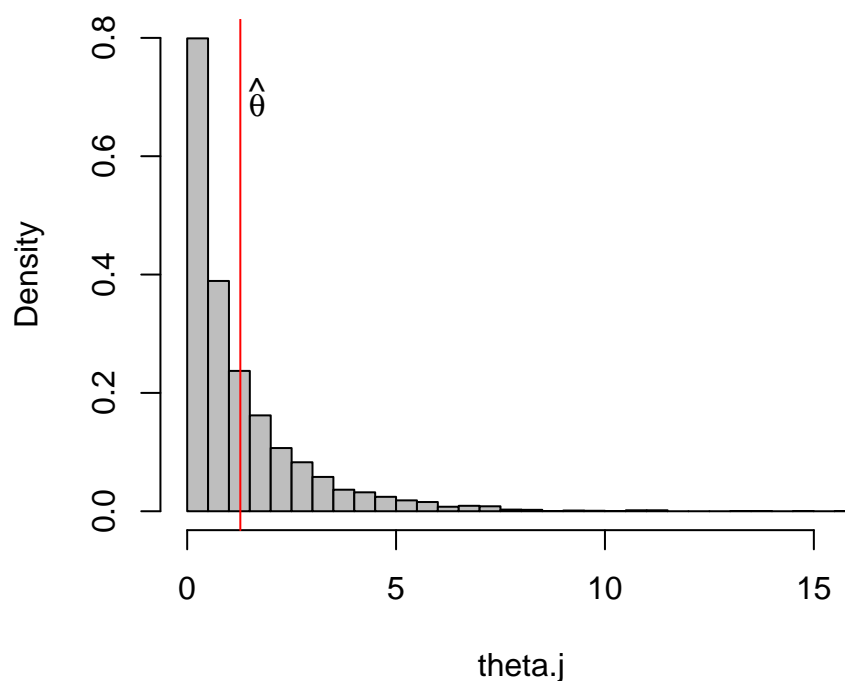
To estimate θ using Monte Carlo approach, we generate random samples $x^{(j)} = (x_1^{(j)}, x_2^{(j)})$ of size 2 from $\chi_1^2, j = 1, \dots, m$. Then compute

$$\hat{\theta} = \frac{1}{m} \sum_{j=1}^m \hat{\theta}^{(j)} = g(X_1^-, X_2) = \frac{1}{m} \sum_{j=1}^m |x_1^{(j)} - x_2^{(j)}|.$$

```
set.seed(1234)
m <- 10000
x <- matrix(rchisq(m,df=1), ncol=2, byrow = TRUE)
g <- function(u) abs(u[1] - u[2])
theta.j <- apply(x, 1, g)
data.frame(theta.hat = mean(theta.j),
            theta.mse = var(theta.j)*m/(m-1))

##   theta.hat theta.mse
## 1  1.272102  2.37895

hist(theta.j, prob = T, col="gray",
      breaks = 30, main = "")
abline(v = mean(theta.j), col="red")
text(mean(theta.j) + 0.4, .7, expression(hat(theta)))
```



Trimmed Mean and LIBOR

Suppose X_1, \dots, X_n is a random sample and $X_{(1)}, \dots, X_{(n)}$ is the corresponding ordered sample. The k^{th} level trimmed sample mean is defined by

$$\bar{X}_{[-k]} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} X_{(i)}.$$

- Trimmed mean is used in many areas specially sports and finance. See [<http://goo.gl/Glnxoq>]
- Less sensitive to outliers than the untrimmed mean but will still give a reasonable estimate of the central tendency or location parameter.
- Unbiased estimator of the population mean when the distribution is symmetric.

Libor Interest rate

Libor interest rate is an index that measures the cost of funds to large global banks operating in London financial markets.

Each day, the British Bankers Association (BBA) surveys a panel of banks (18 major global banks for the USD Libor), asking the question,

“At what rate could you borrow funds, were you to do so by asking for and then accepting inter-bank offers in a reasonable market size just prior to 11 am?”

The BBA throws out the highest 4 and lowest 4 responses, and averages the remaining middle 10, yielding a 23% trimmed mean. The average is reported at 11:30 a.m.

See [<http://goo.gl/MJtGT>] , [<http://goo.gl/A3W0yC>], and [<http://goo.gl/nfzTe>]

The Libor benchmark interest rate is a trimmed mean with $k = 4$, $\bar{X}_{[-4]}$.

Example 2: Estimating the Libor (Trimmed Mean)

Interest rates are usually modelled as lognormal processes.

For our purposes, suppose X has mean -0.7 and standard deviation 1 on the log scale, i.e. $E(\log X) = -0.7$ and $Var(\log X) = 1$.

We select these parameter values so that the Libor rate will yield an average value of 0.56 which is the current 12 months USD Libor rate.

Goal: Obtain a Monte Carlo estimate of the MSE of the Libor, $\bar{X}_{[-4]}$.

1. Generate the replicates $T^{(j)}$, $j = 1, \dots, m$ by repeating :

- (a) Generate $x_1^{(j)}, \dots, x_n^{(j)}$, IID from the distribution of X .
- (b) Sort $x_1^{(j)}, \dots, x_n^{(j)}$ increasing order, to obtain $x_{(1)}^{(j)} \leq \dots \leq x_{(n)}^{(j)}$.
- (c) Compare $T^{(j)} = \frac{1}{n-8} \sum_{i=5}^{n-4} x_{(i)}^{(j)}$.

2. Compute $MSE(T) = \frac{1}{m} \sum_{j=1}^m (T^{(j)} - \bar{T})^2$, where $\bar{T} = \frac{1}{m} \sum_{j=1}^m T^{(j)}$.

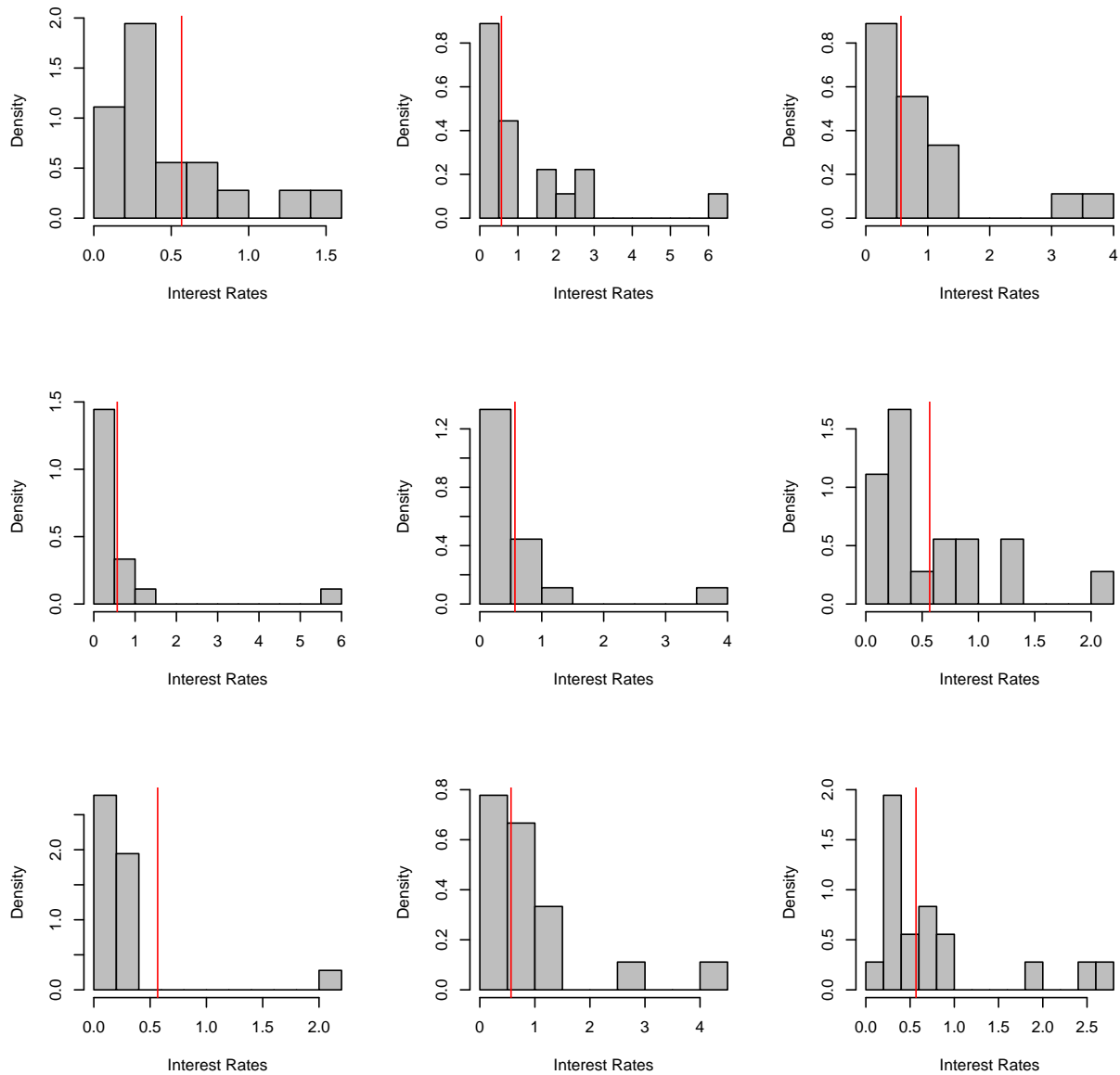
Generic Function: Trimmed Mean

```
# computes any kth level trimmed mean
tmean_fn <- function(u,k) {
  n <- length(u)
  u <- sort(u)      # sort values
  sum(u[(k+1):(n-k)])/(n-2*k)
}

n <- 18           # 18 banks
m <- 10^4         # no of replicates
# generate interest rates from lognormal
# with mean and sd on the log scale
set.seed(1234)
dat <- rlnorm(n*m, meanlog=-0.7, sdlog=1)
x <- matrix(dat,ncol=18,byrow=T)
# libor is 4th level TM
tmean <- apply(x,1,tmean_fn, k=4)
data.frame(mean = mean(tmean),
            mse = mean((tmean-mean(tmean))^2))
```

```
##          mean      mse
## 1 0.5669937 0.02230814
```

```
par(mfcol = c(3,3))
for (s in 1:9) {
  hist(x[s,], prob = TRUE,col="gray",
       breaks = 10, main = "",
       xlab = "Interest Rates")
  abline(v = mean(tmean), col="red")}
```



Lets look at interest estimates and its MSE when we vary the trimmed mean from $k = 0$ (ordinary mean) to $k = 4$ (Libor Rate).

```

MSE.fn <- function(k){
  tmean <- apply(x, 1, tmean_fn, k)
  data.frame(mean = mean(tmean),
             mse = mean((tmean - mean(tmean))^2))
}
res <- sapply(0:4, MSE.fn)
print(res,4)

```

```

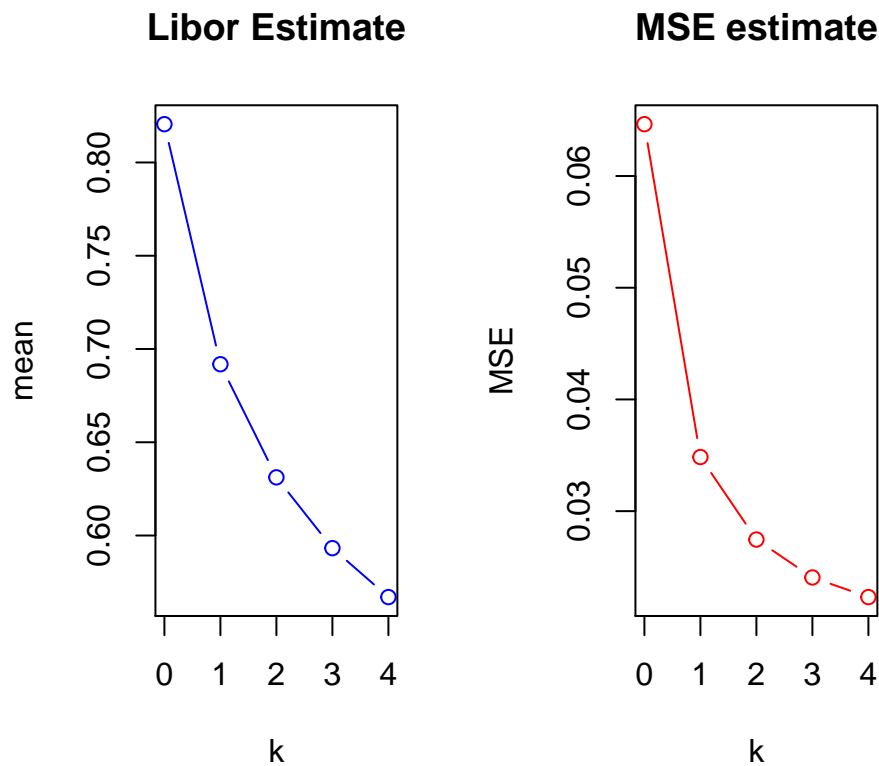
##      [,1]    [,2]    [,3]    [,4]    [,5]
## mean 0.8205 0.6918 0.6312 0.5932 0.567
## mse  0.06464 0.03484 0.02746 0.02406 0.02231

```

```

par(mfcol = c(1,2))
plot(0:4,res[1,], type = "b", col="blue",
     ylab = "mean", main = "Libor Estimate", xlab = "k")
plot(0:4,res[2,], type = "b", col="red",
     ylab = "MSE", main = "MSE estimate", xlab = "k")

```



Estimating the Confidence Level

If (U, V) is a confidence interval estimate for an unknown parameter θ , then U and V are statistics with distribution that depend on the distribution F_X of the sampled population X .

The confidence level is the probability that the interval (U, V) covers the true value of the parameter θ .

Evaluating the confidence level is therefore an integration problem.

In statistical applications, the integrand $g(x)$ is often not specified in estimating integral $\int g(x)dx$, but only that the variable $g(X)$ is easily generated.

Confidence Interval for Variance

If X_1, \dots, X_n is a random sample (IID) from $N(\mu, \sigma^2)$, $n \geq 2$, and S^2 is the sample variance, then

$$V = \frac{(n-1)S^2}{\sigma^2} \sim \chi_{(n-1)}^2.$$

A one side $100(1 - \alpha)\%$ CI for σ^2 is $(0, (n-1)S^2/\chi_\alpha^2)$, where χ_α^2 is the α -quantile of the $\chi_{(n-1)}^2$ distribution.

If the sampled pop'n is indeed $N(\mu, \sigma^2)$, then

$$P\left(\frac{(n-1)S^2}{\chi_\alpha^2} > \sigma^2\right) = P\left(S^2 > \frac{\sigma^2 \chi_\alpha^2}{n-1}\right) = 1 - \alpha.$$

MC approach to estimate a confidence level

Suppose that $X \sim F_X$ is the random variable of interest and that θ is the target parameter to be estimated.

1. For each replicate, indexed $j = 1, \dots, m$:

- (a) Generate the j^{th} random sample, $X_1^{(j)}, \dots, X_n^{(j)}$.
- (b) Compute the confidence interval C_j for the j^{th} sample.
- (c) Compute $y_j = I(\theta \in C_j)$ for the j^{th} sample, where I is the indicator set function.

2. Compute

$$\bar{y} = \frac{1}{m} \sum_{j=1}^m y_j.$$

The estimator \bar{y} is a sample proportion estimating the true confidence level $1 - \alpha^*$, so

$$Var(\bar{y}) = (1 - \alpha^*)\alpha^*/m$$

$$\hat{se}(\bar{y}) = \sqrt{(1 - \bar{y})\bar{y}/m}.$$

Example 3: MC estimate of confidence level example

Consider calculation of the 95% upper confidence limit (UCL) for random sample of size $n = 20$ from $N(0, \sigma^2 = 4)$.

```
n <- 20
alpha <- .05
set.seed(1234)
x <- rnorm(n, mean=0, sd=2)
UCL <- (n-1) * var(x) / qchisq(alpha, df=n-1)
UCL
```

```
## [1] 7.720979
```

The sample proportion of intervals that contain $\sigma^2 = 4$ is a MC estimate of the true confidence level $1 - \alpha^* = 0.95$.

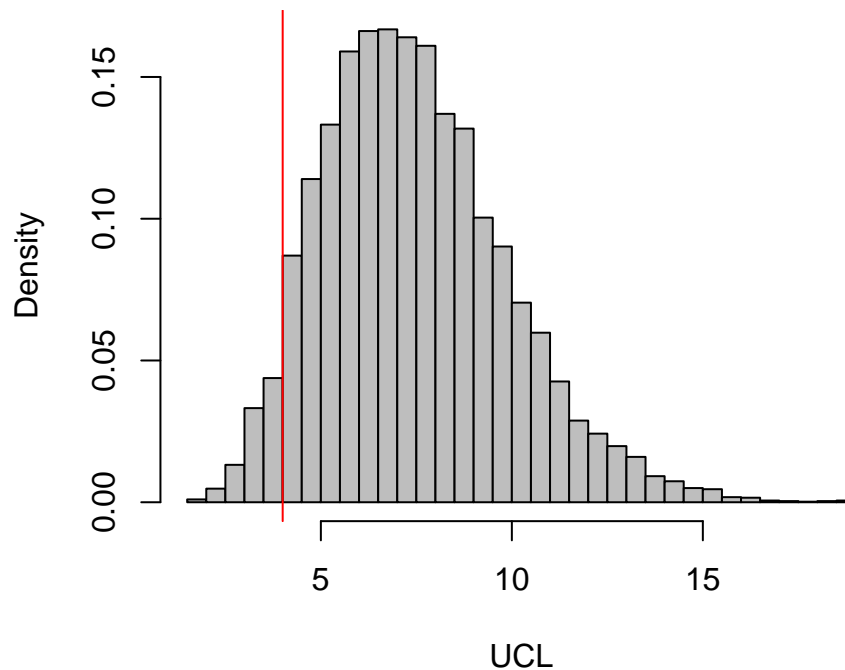
```
myCI <- function(n,alpha){
  x <- rnorm(n, mean = 0, sd = 2)
  (n-1) * var(x) / qchisq(alpha, df = n-1)
}
n <- 20
alpha <- .05
m <- 10^4
set.seed(1234)
# use replicate function to repeatedly execute myCI above
UCL <- replicate(m, expr = myCI(n,alpha))

# compute the mean to get the confidence level and
# estimate the standard error
data.frame(level.est = mean(UCL>4),
  level.se = sqrt((1-mean(UCL>4))*mean(UCL>4)/m))

##      level.est      level.se
## 1      0.952 0.002137662
```

```
# density histogram
hist(UCL, prob = TRUE, col="gray", breaks = 30)
abline(v = 4, col="red")
text(4 + 0.4, .4, expression(sigma^2))
```


Histogram of UCL



What happens if we sample from a non-normal, say $\chi^2(2)$ population?

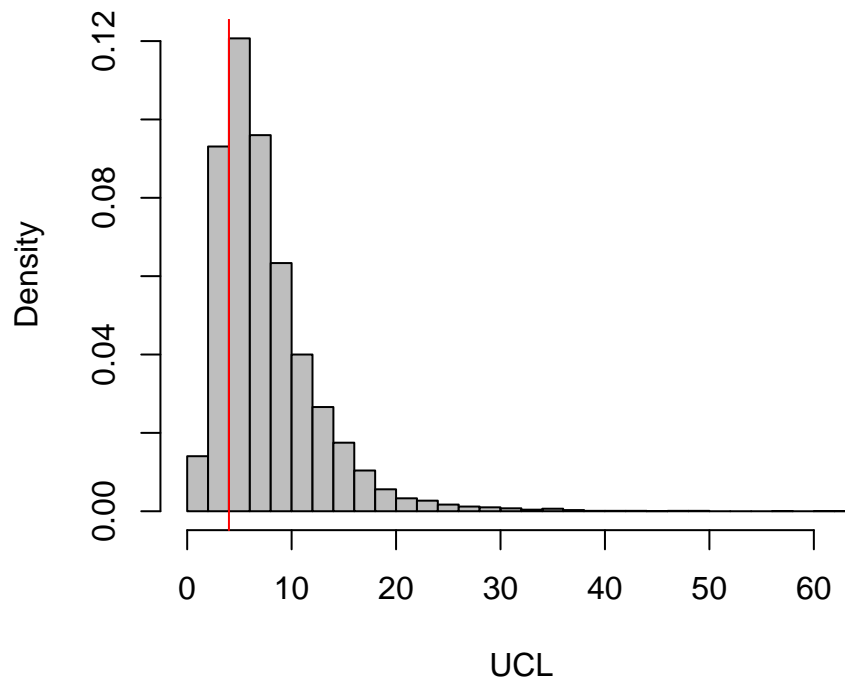
```
myCI.chi <- function(n,alpha){  
  x <- rchisq(n, df=2)  
  (n-1) * var(x) / qchisq(alpha, df = n-1)  
}  
set.seed(1234)  
UCL <- replicate(m, expr = myCI.chi(n,alpha))  
mean(UCL > 4)
```

```
## [1] 0.7857
```

The estimated confidence level is below the 95% nominal confidence level when the sample pop'n is not normal.

```
# plot  
hist(UCL, prob = TRUE, col="gray", breaks = 30)  
abline(v = 4, col="red")  
text(4+0.4, .4, expression(sigma^2))
```

Histogram of UCL



Example 4: CI Simulation for Binomial Proportion

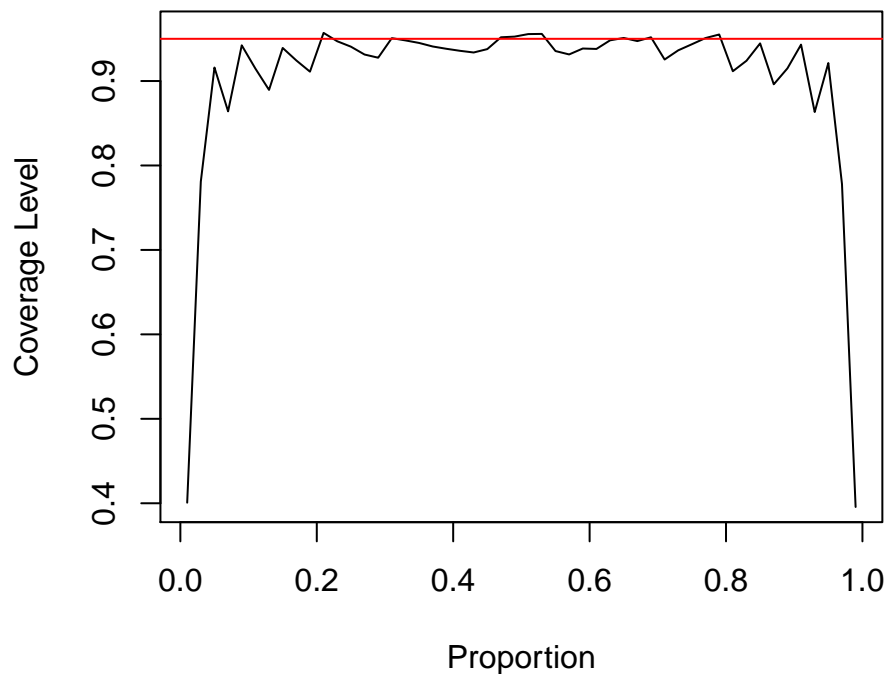
Use the Wald Formula in Constructing CI's for proportion p in distribution $Binomial(n, p)$.

```
myCI_p <- function(x, n, z) {  
  p.hat <- mean(x)  
  LCL <- p.hat - z*sqrt(p.hat*(1-p.hat)/n)  
  UCL <- p.hat + z*sqrt(p.hat*(1-p.hat)/n)  
  c(p.hat, LCL, UCL)  
}  
  
myCov_p <- function(p, n, m, z){  
  # each column is a random sample  
  data <- replicate(m, rbinom(n,1,p))  
  CL_p <- apply(data, 2, myCI_p,n,z)  
  # this proportion gives the empirical coverage.  
  mean( CL_p[,2] <= p & p <= CL_p[,3])  
}  
  
p <- .5 # we pretend that unknown value is .5  
n <- 50 # the sample size  
m <- 10^4 # number of iterations  
set.seed(1234)  
z <- qnorm(.975) # we simulate 95% confidence intervals
```

```
myCov_p(p, n, m, z) # estimated coverage level (p=0.5)
```

```
## [1] 0.934
```

```
p <- seq(0.01, 0.99, by = 0.02) # sequence of p's  
covp <- sapply(p, myCov_p, n, m, z)  
plot(p, covp, type = "l", ylim = c(0.4, 0.96),  
      xlab = "Proportion", ylab = "Coverage Level")  
abline(h = 0.95, col = "red")
```



Note that the Wald's CI have very low coverage levels when p is close to either 0 and 1.

Example 5: CI for Linear Regression Slope

Evaluate the confidence (coverage) level of 95% CI for regression slope in

$$y = 3x + \epsilon, \quad \epsilon \sim_{IID} N(0, 1).$$

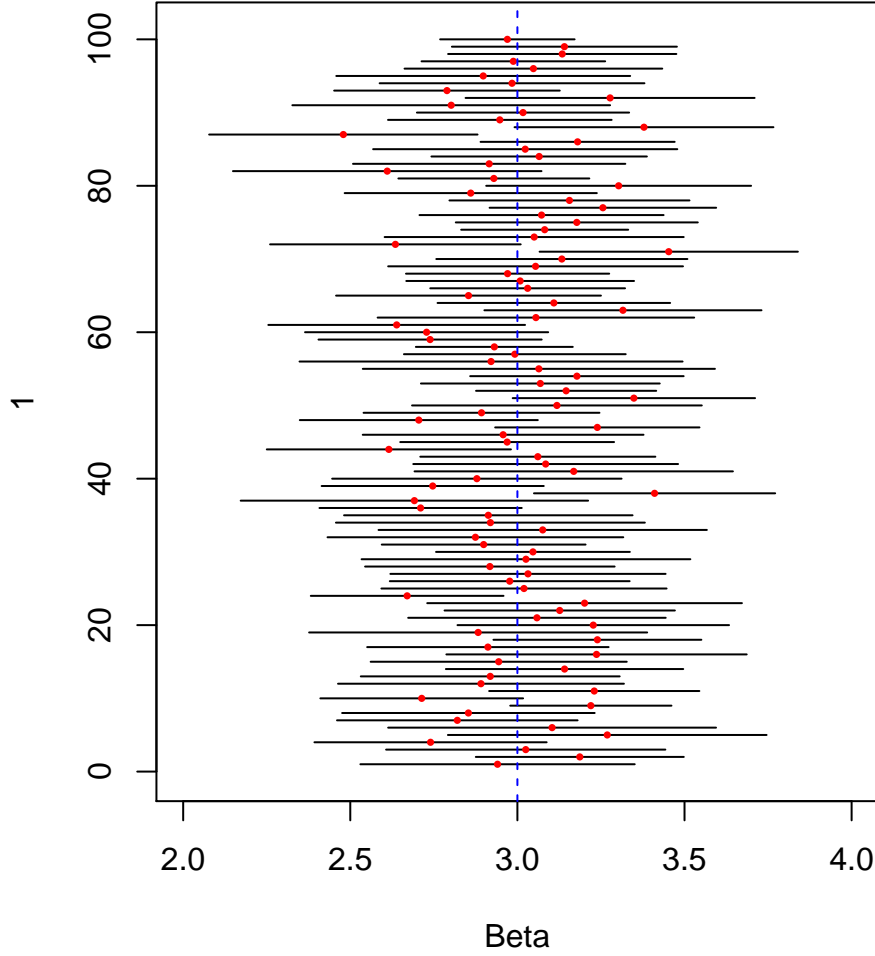
```

myCI_LR <- function(n, alpha){
  x <- rnorm(n)
  y <- 3*x + rnorm(n)
  fit <- lm(y ~ x)
  confint(fit, level = 1 - alpha)[2, ]
}
alpha <- 0.05
n <- 30 # sample size
m <- 1000
set.seed(1234)
CL <- replicate(m, expr = myCI_LR(n, alpha))
mean(CL[,1] < 3 & 3 < CL[,2])

## [1] 0.952

# plot the first 100 CI estimates of beta = 3
plot(1, 1, xlim=c(2,4), ylim=c(0,101), type='n', xlab = "Beta")
for (i in 1:100) {
  # we'll plot CI schematics from bottom up
  lines(CL[, i], c(i, i))
  points(mean(CL[, i]), i, pch=16, cex=.5, col = "red")
}
abline(v=3, lty=2, col = "blue")

```



Hypothesis Tests

We want to test a hypothesis on parameter θ in the parameter space Θ . The hypotheses are

$$H_0 : \theta \in \Theta_0 \text{ vs } H_1 : \theta \in \Theta_1$$

where Θ_0 and Θ_1 are partitions of the space Θ .

A *Type I error* occurs if H_0 is rejected when in fact the H_0 is true.

A *Type II error* occurs if the H_0 is not rejected when in fact the H_0 is false.

For a given test procedure, let $\pi(\theta)$ denote the probability of rejecting H_0 and is a function of θ .

The *significance level* α of a test is an upper bound on the probability of Type I error,

$$\alpha = \sup_{\theta \in \Theta_0} \pi(\theta).$$

If the test procedure is replicated a large number of times under the conditions of the null hypothesis, the observed Type I error rate should be at most (approximately) α .

If T is the test statistic and T^* is the observed value of the test statistic, then T^* is significant if the test decision based on T^* is the observed value of the test statistic.

The significance probability or p-value is the smallest possible value of α such that the observed test statistic would be significant.

Empirical Type I error rate

An empirical Type I error rate can be computed by a Monte Carlo experiment and is the sample proportion of significant test statistics among the large number of replicates.

1. For each replicate, indexed $j = 1, \dots, m$:
 - (a) Generate the j^{th} random sample, $X_1^{(j)}, \dots, X_n^{(j)}$ from the null distribution.
 - (b) Compare the test statistic T_j from the j^{th} sample.
 - (c) Record the test decision $I_j = 1$ if H_0 is rejected at significance level α and otherwise $I_j = 0$.
2. Compute the proportion of significant tests

$$\hat{p} = \frac{1}{m} \sum_{j=1}^m I_j.$$

This proportion is the *observed Type I error rate*.

An estimate of the standard error $se(\hat{p})$ is

$$\hat{se}(\hat{p}) = \sqrt{\frac{\hat{p}(1 - \hat{p})}{m}} \leq \frac{0.5}{\sqrt{m}}.$$

Example 6: Empirical Type I error rate

Suppose X_1, \dots, X_{20} is a random sample from $N(\mu, \sigma^2)$. Test $H_0 : \mu = 500$ vs $H_1 : \mu > 500$ at $\alpha = 0.05$.

Under H_0 ,

$$T^* = \frac{\bar{X} - 500}{S/\sqrt{20}} \sim t(19),$$

where $t(19)$ denotes the Student t dist'n with 19 degree of freedom.

Use a MC method to compute an empirical probability of Type I error when $\sigma = 100$, and check that it is approximate equal to $\alpha = 0.05$.

Use `t.test` in R to compute T^* and based the test decision on the reported p-values returned by `\tt t.test`.

```

myHT <- function(n, mu0, sigma){
  x <- rnorm(n, mu0, sigma)
  ttest <- t.test(x, alternative = "greater", mu = mu0)
  return(ttest$p.value)
}
n <- 20
alpha <- .05
mu0 <- 500
sigma <- 100
m <- 1000
set.seed(1234)
p <- replicate(m, expr = myHT(n, mu0, sigma))
p.hat <- mean(p < alpha)
se.hat <- sqrt(p.hat * (1 - p.hat) / m)
data.frame(p.hat, se.hat)

```

```

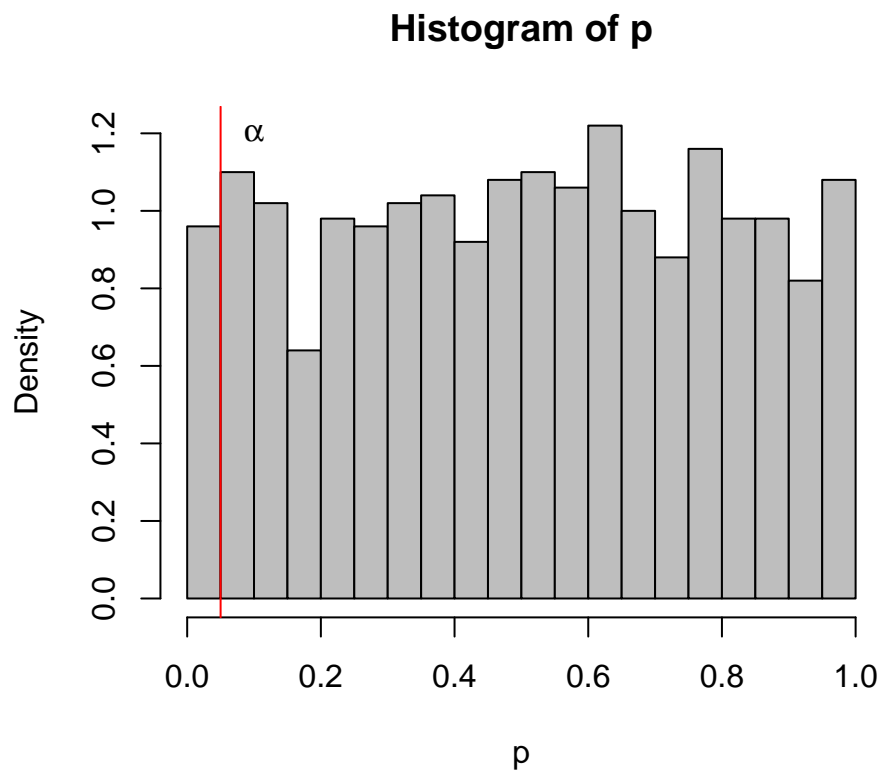
##   p.hat      se.hat
## 1 0.048 0.006759882

```

```

hist(p, prob = TRUE, col="gray", breaks = 20)
abline(v = alpha, col="red")
text(alpha + 0.05, 1.2, expression(alpha))

```



Power of a Test

For a given test procedure, recall that $\pi(\theta)$ denote the probability of rejecting H_0 and is a function of θ . $\pi(\theta)$ is called the power of a test given that θ is the true value of the parameter. π is called the power function

$$\pi : \Theta \rightarrow [0, 1].$$

For a given $\theta_1 \in \Theta_1$, the prob'y of Type II error is $1 - \pi(\theta_1)$.

Ideally, we prefer a test with low probability of error where.

Type I error is controlled by choice of the significance level α usually set at 0.05.

Low Type II error corresponds to high power under the alternative hypothesis H_1 .

Thus, when comparing test procedures for the same hypotheses at the same significance level, we compare the power of the tests.

In general, the comparison is not one but many; the power $\pi(\theta_1)$ of a test under the alternative hypothesis depends on the particular value of the alternative θ_1 .

For the t-test in Example 6, $\Theta_1 = (500, \infty)$.

In general, Θ_1 is more complicated.

If the power of the test cannot be defined analytically, the power of a test against a fixed alternative $\theta_1 \in \Theta_1$ can be estimated by Monte Carlo methods.

Note that the power function is defined for all $\theta \in \Theta$, but the significance level α controls $\pi(\theta) \leq \alpha$ for all $\theta \in \Theta_0$.

Algorithm: Empirical Power Simulation

1. Select a particular value of the parameter $\theta_1 \in \Theta$.
2. For each replicate, indexed by $j = 1, \dots, m$;
 - (a) Generate the j^{th} random sample $x_1^{(j)}, \dots, x_n^{(j)}$ under conditions of the alternative $\theta = \theta_1$.
 - (b) Compute the test statistic T_j from the j^{th} sample.
 - (c) Record the test decision: set $I_j = 1$ if H_0 is rejected at significance level α , and otherwise set $I_j = 0$.
3. Compute the proportion of significant tests

$$\hat{\pi}(\theta_1) = \frac{1}{m} \sum_{j=1}^m I_j.$$

Example 7: Empirical power of a t-test


```

pval <- function(mu1, sigma, n) {
  #simulate under alternative mu1
  x <- rnorm(n, mean = mu1, sd = sigma)
  ttest <- t.test(x, alternative = "greater", mu = mu0)
  ttest$p.value
}

pvalues <- function(mu1, sigma, n, m) {
  val <- replicate(m, expr = pval(mu1, sigma, n))
  mean(val <= .05)
}

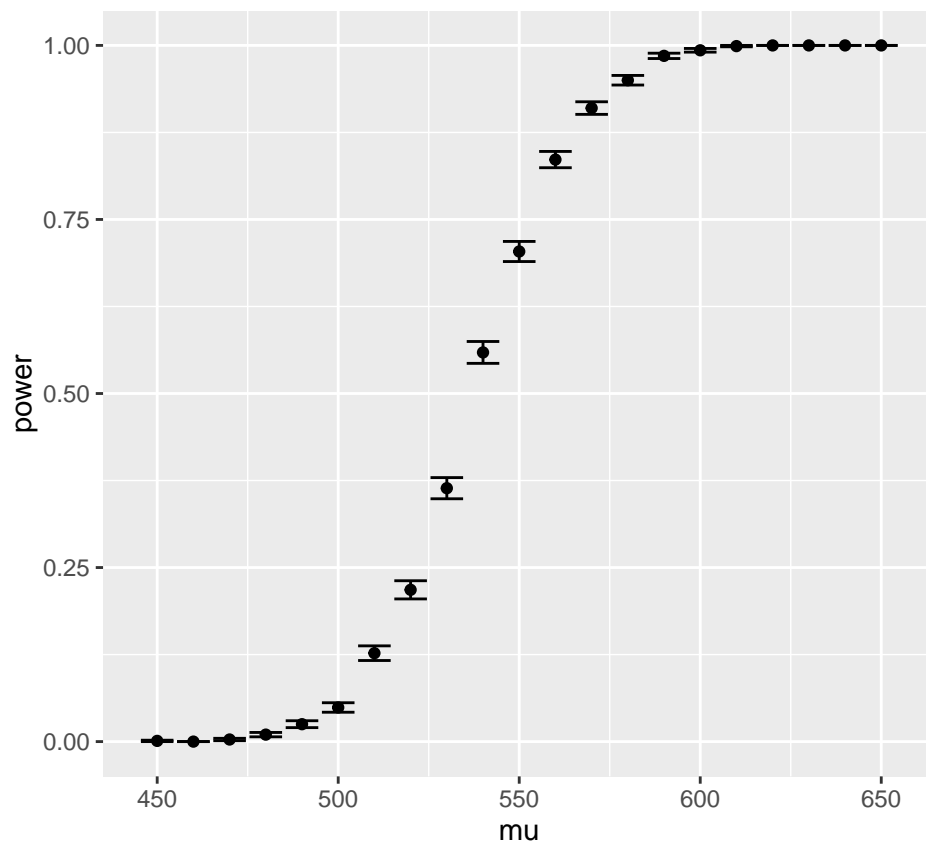
n <- 20
m <- 1000
mu0 <- 500
sigma <- 100
mu <- c(seq(450, 650, 10)) #alternatives

power <- sapply(mu, pvalues, sigma, n, m)

se <- sqrt(power * (1-power) / m)
yplus = power+se
yminus = power-se
df <- data.frame(mu,power,yminus,yplus)

# let's graph these empirical power values with error bars
library(ggplot2)
ggplot(data = df, mapping = aes(x = mu, y = power)) +
  geom_point() + ## add the points
  geom_errorbar(mapping = aes(x = mu, ymin = yminus, ymax = yplus) ## add the error bars
)

```



Note the empirical power $\pi(\hat{\theta})$ is small when θ is close to $\theta_0 = 500$, and increase as θ move farther away from θ_0 , approaching 1 as $\theta \rightarrow \infty$.

In R, there is a function `power.t.test` that do the same power calculation as above but with a single command.

```
df[which(mu == 530),]
```

```
##      mu power   yminus   yplus
## 9 530 0.364 0.3487847 0.3792153
```

```
# delta = mu1 - mu0
power.t.test(n = n, delta = 30, sd = sigma, sig.level = 0.05,
             type = "one.sample", alternative = "one.sided")
```

```
##
##      One-sample t test power calculation
##
##              n = 20
##            delta = 30
##              sd = 100
##      sig.level = 0.05
##        power = 0.3627796
## alternative = one.sided
```

This is a powerful command that can do much more than just calculate the power of a test.

For example it can also be used to calculate the number of observations necessary to achieve a given power. For more information check out the help page, `help(power.t.test)`.