区間推定_尤度比

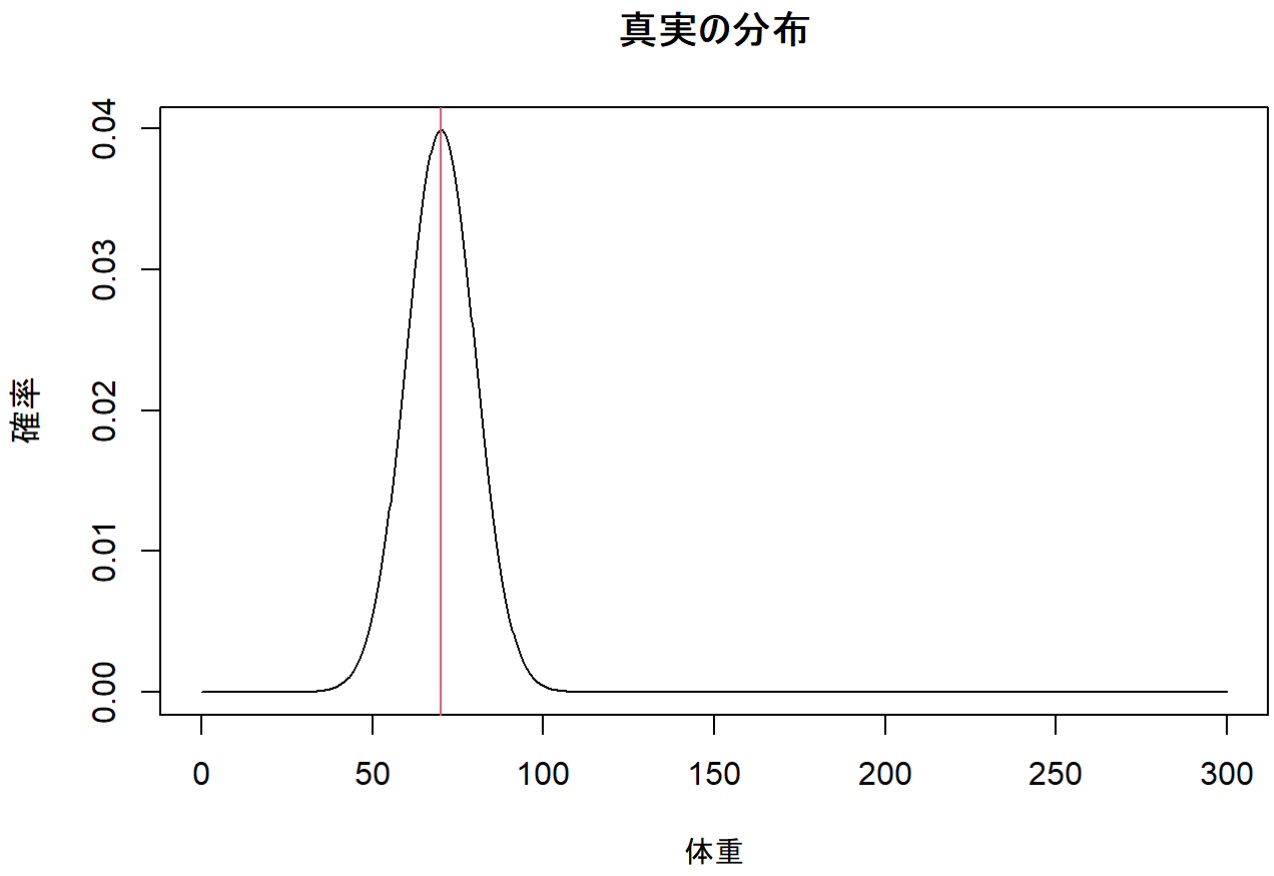
平均体重を推定する

サンプルの平均値

「真実の分布」が平均50、標準偏差10のとき、どうしたら「真実の平均」を知ることができるか？

一部のサンプルを取り出して、そのサンプルの平均を計算して、代用する。

```
m.true <- 70
sd.true <- 10

w <- seq(from=0,to=300,length=1000)
prob.w <- dnorm(w,m.true,sd.true)
plot(w,prob.w,xlab="体重",ylab="確率",type="l",main="真実の分布")
abline(v=m.true,col=2)
```



10人をサンプリングして平均値を出してみる。

```
n.sample <- 10
smpl <- rnorm(n.sample,m.true,sd.true)
mean(smpl)
```

```
## [1] 71.99814
```

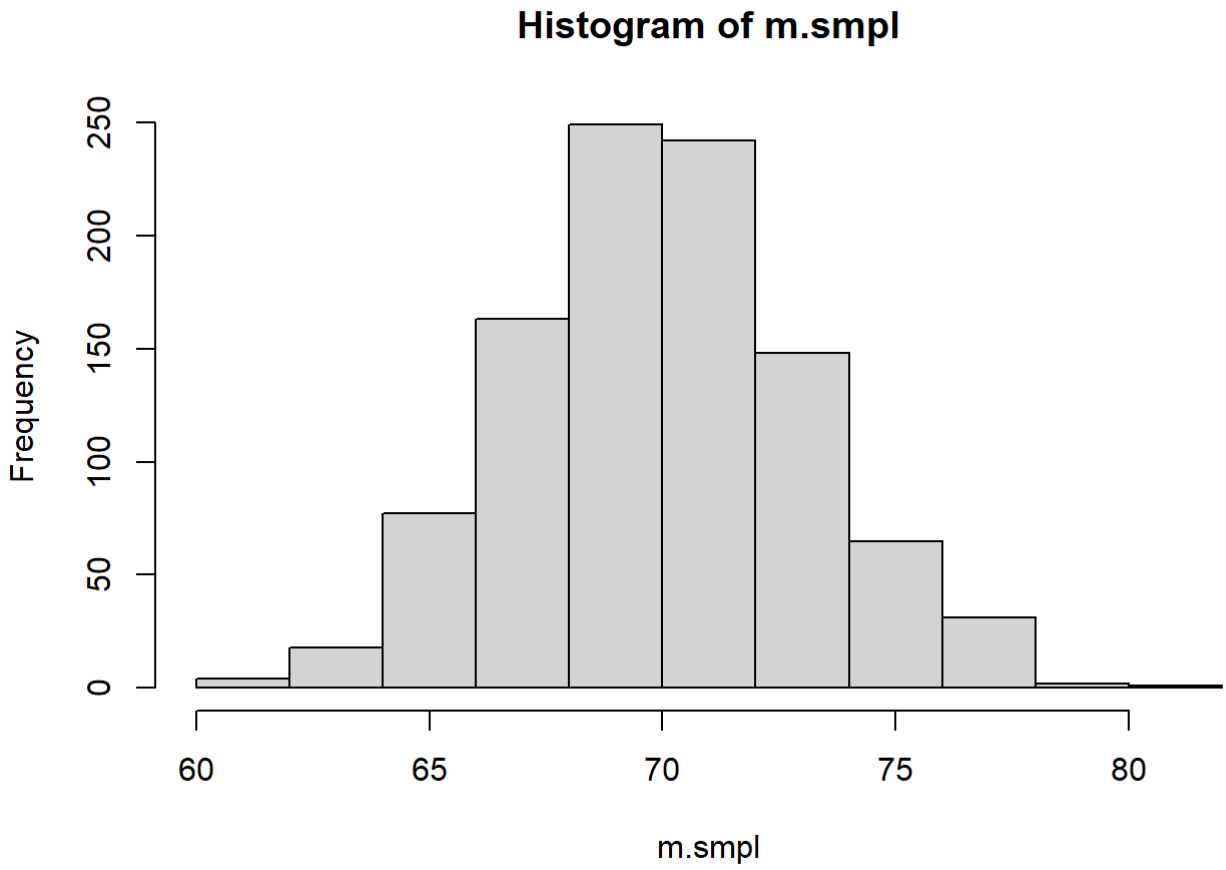
サンプリングするたびに値は変わる。

```
n.iter <- 10
for(i in 1:n.iter){
  smpl <- rnorm(n.sample,m.true,sd.true)
  print(mean(smpl))
}
```

```
## [1] 71.52448
## [1] 66.4576
## [1] 65.53299
## [1] 72.98992
## [1] 71.48538
## [1] 67.39678
## [1] 70.42768
## [1] 73.10865
## [1] 72.89232
## [1] 71.5584
```

どれくらい変わるかというと...

```
n.iter <- 1000
m.smpl <- rep(NA,n.iter)
for(i in 1:n.iter){
  smpl <- rnorm(n.sample,m.true,sd.true)
  m.smpl[i] <- mean(smpl)
}
hist(m.smpl)
```



サンプルから信頼区間

本当の値を当てることはできない 「ここから、ここの間に真の平均は入る」と言えば、当たる確率が出せる

95% 信頼区間とは、

「サンプルがあったときに、その値を使って、『ここからここまでと予想する』というルールを決める」

「そのルールに従うと、95%の場合、真の値が、その範囲に入る」

と言うようにデザインされた『ルール』のこと。 もしくは、その『ルール』に従って算出した『区間』のこと。

```
library(Rmisc)
```

```
## Warning: パッケージ 'Rmisc' はバージョン 4.1.3 の R の下で造られました
```

```
##  要求されたパッケージ lattice をロード中です
```

```
##  要求されたパッケージ plyr をロード中です
```

```
CI(smpl,ci=0.95)
```

```
##    upper     mean    lower
## 74.93747 72.13110 69.32472
```

本当に95%の確率であたっているのか？

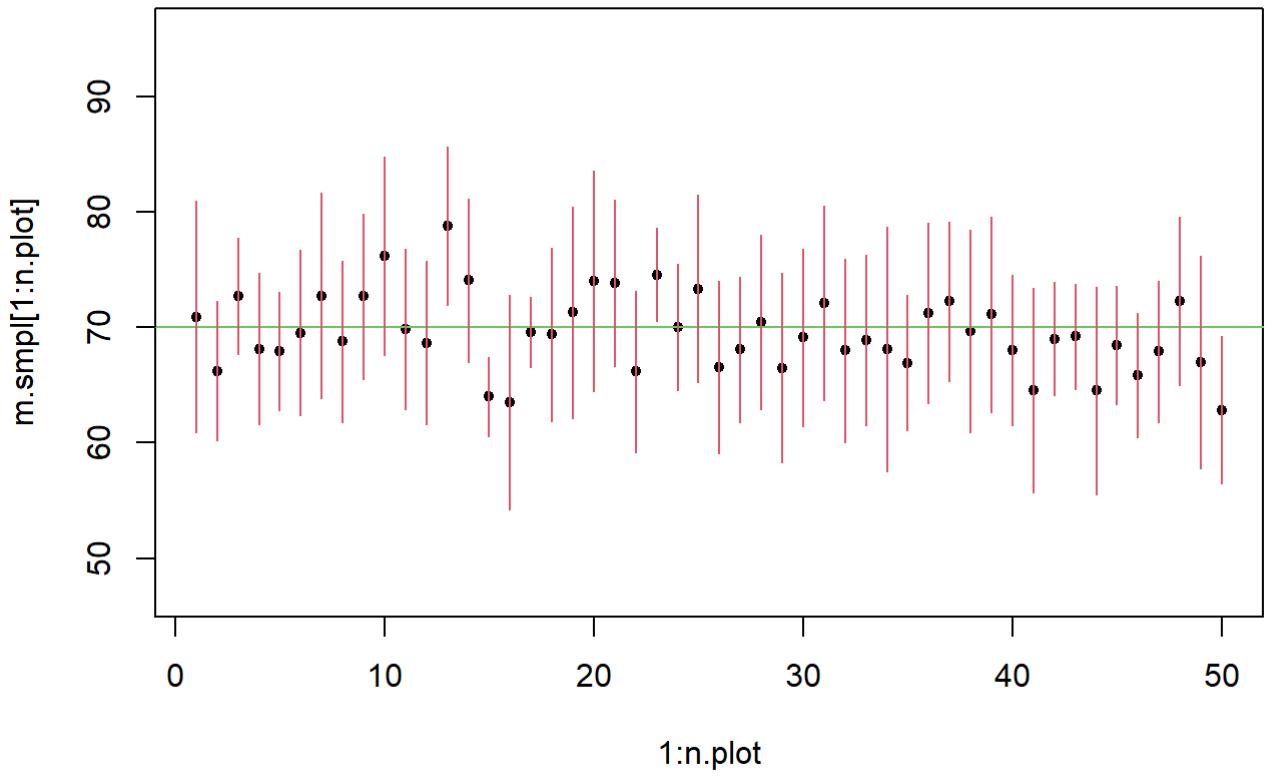
```
n.iter <- 1000
m.smpl <- rep(NA,n.iter)
up.low <- matrix(NA,n.iter,2)
atari <- rep(NA,n.iter)
for(i in 1:n.iter){
  smpl <- rnorm(n.sample,m.true,sd.true)
  tmp <- CI(smpl)
  m.smpl[i] <- tmp[2]
  up.low[i,] <- tmp[c(1,3)]
  if(up.low[i,1]>m.true & up.low[i,2]<m.true){
    atari[i] <- 1
  }else{
    atari[i] <- 0
  }
}
```

当たった確率

```
mean(atari)
```

```
## [1] 0.952
```

```
n.plot <- n.iter/20
plot(1:n.plot,m.smpl[1:n.plot],pch=20,ylim=c(min(up.low)-5,max(up.low)+5))
abline(h=m.true,col=3)
for(i in 1:n.plot){
  segments(i,up.low[i,1],i,up.low[i,2],col=2)
}
```



どうやって計算しているかは、説明していない。

正規分布を仮定して、比較的簡単に、+ - x / で計算している。

一応、式を載せますが、今日は、式は気にしないで行きます。

$$m \pm k\sqrt{\frac{a}{n}}$$

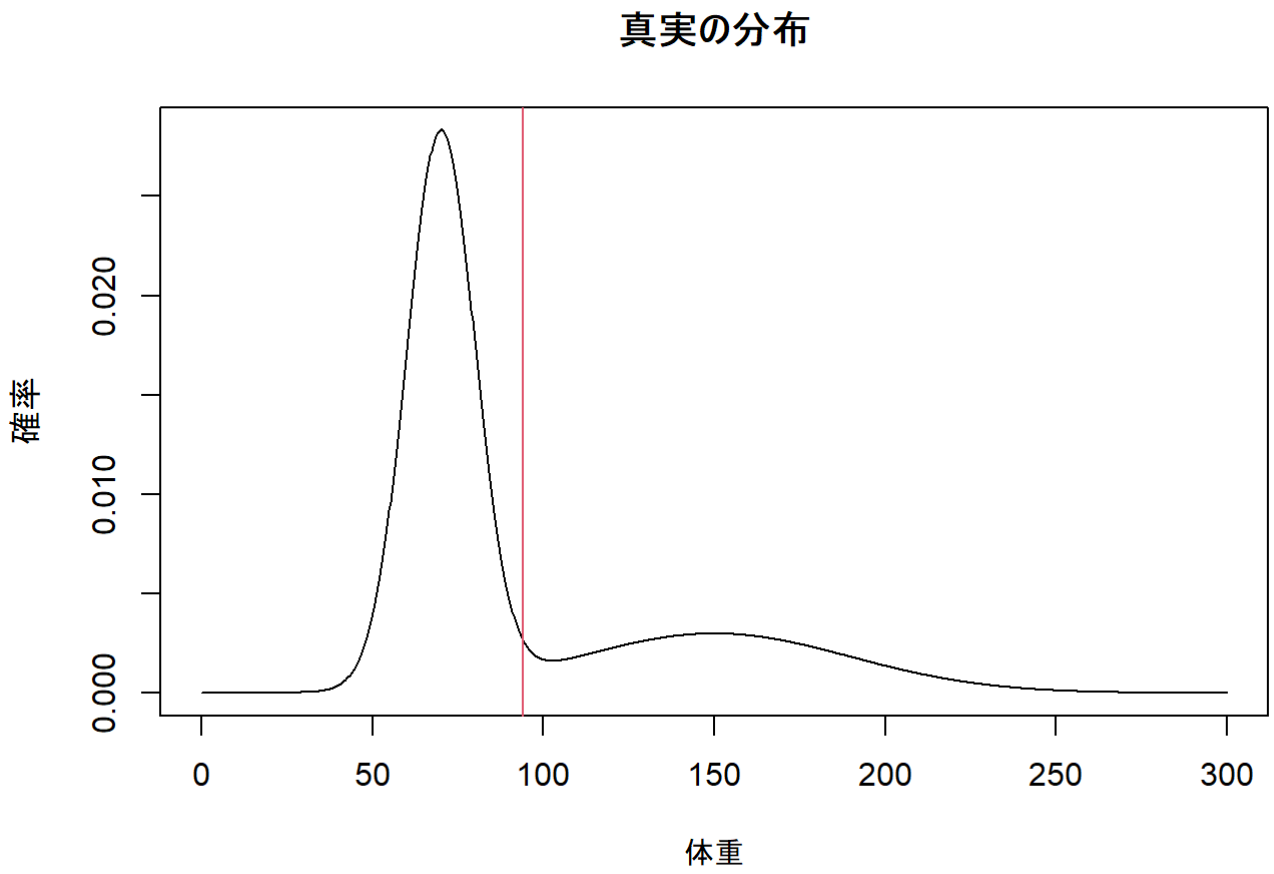
$$m = \frac{\sum x_i}{n}$$

$$a = \frac{\sum (x_i - m)^2}{n-1}$$

分布がきれいでないとき

正規分布でないとどうなるか。

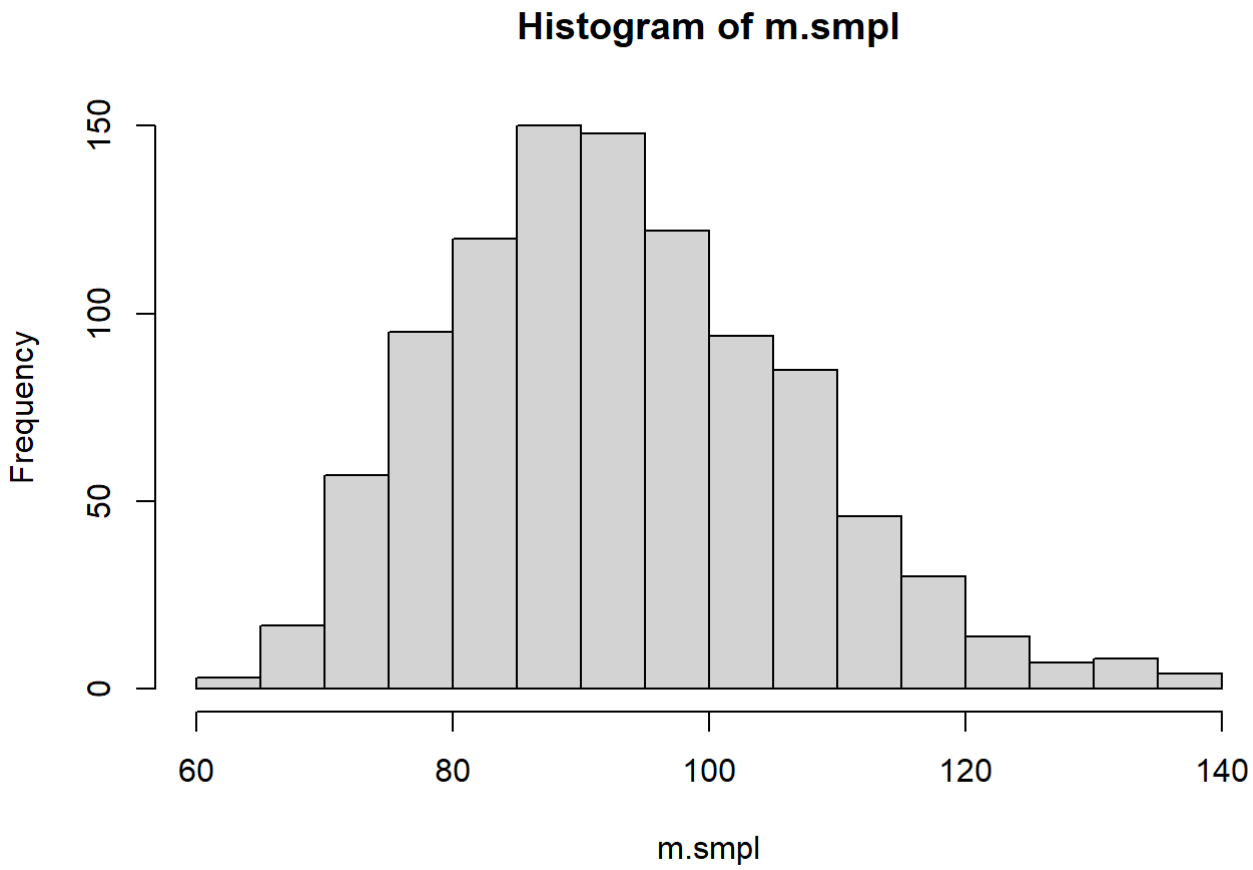
```
n <- 50
m.true1 <- 70
sd.true1 <- 10
m.true2 <- 150
sd.true2 <- 40
r <- 0.7

m.true <- r * m.true1 + (1-r)*m.true2

w <- seq(from=0,to=300,length=1000)
prob.w1 <- dnorm(w,m.true1,sd.true1)
prob.w2 <- dnorm(w,m.true2,sd.true2)
plot(w,prob.w1*r+prob.w2*(1-r),xlab="体重",ylab="確率",type="l",main="真実の分布")
abline(v=m.true,col=2)
```




```
n1 <- rbinom(1,n.sample,c(r,1-r))
smpl1 <- rnorm(n1,m.true1,sd.true1)
smpl2 <- rnorm(n.sample-n1,m.true2,sd.true2)
smpl <- c(smpl1,smpl2)
mean(smpl)
```

```
## [1] 95.17157
```

```
n.iter <- 1000
m.smpl <- rep(NA,n.iter)
for(i in 1:n.iter){
  n1 <- rbinom(1,n.sample,c(r,1-r))
  smpl1 <- rnorm(n1,m.true1,sd.true1)
  smpl2 <- rnorm(n.sample-n1,m.true2,sd.true2)
  smpl <- c(smpl1,smpl2)
  m.smpl[i] <- mean(smpl)
}
hist(m.smpl)
```



95% 信頼区間

```
library(Rmisc)
CI(smpl,ci=0.95)
```

```
##     upper      mean     lower
## 107.68422  83.22110  58.75798
```

本当に95%の確率であたっているのか？

```
n.iter <- 1000
m.smpl <- rep(NA,n.iter)
up.low <- matrix(NA,n.iter,2)
atari <- rep(NA,n.iter)
for(i in 1:n.iter){
  n1 <- rbinom(1,n.sample,c(r,1-r))
  smpl1 <- rnorm(n1,m.true1,sd.true1)
  smpl2 <- rnorm(n.sample-n1,m.true2,sd.true2)
  smpl <- c(smpl1,smpl2)
  tmp <- CI(smpl)
  m.smpl[i] <- tmp[2]
  up.low[i,] <- tmp[c(1,3)]
  if(up.low[i,1]>m.true & up.low[i,2]<m.true){
    atari[i] <- 1
  }else{
    atari[i] <- 0
  }
}
```

けっこう、外れている...。

```
mean(atari)
```

```
## [1] 0.877
```

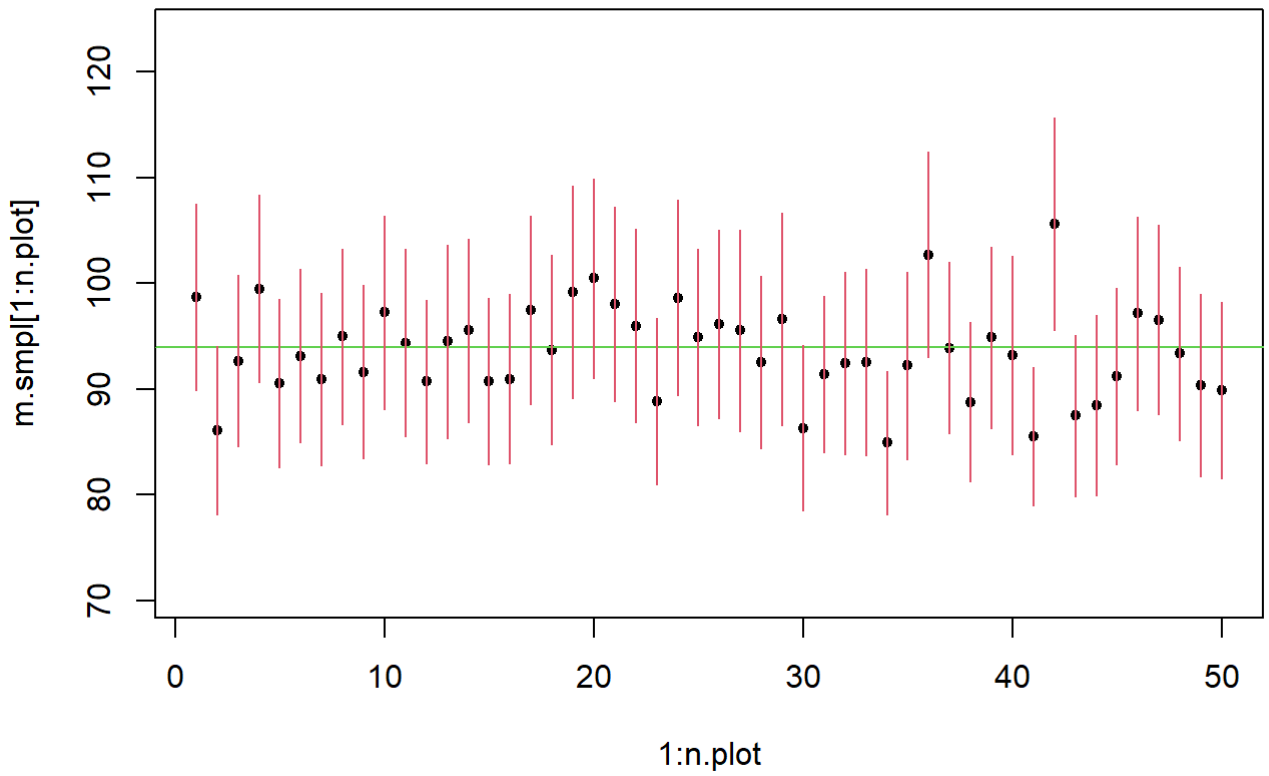
サンプル数が少ない(n.sample=10)ので、分布の全体をサンプリングできていないから。

サンプル数を増やしてやってみる。

```
n.sample <- 100
n.iter <- 1000
m.smpl <- rep(NA,n.iter)
up.low <- matrix(NA,n.iter,2)
atari <- rep(NA,n.iter)
for(i in 1:n.iter){
  n1 <- rbinom(1,n.sample,c(r,1-r))
  smpl1 <- rnorm(n1,m.true1,sd.true1)
  smpl2 <- rnorm(n.sample-n1,m.true2,sd.true2)
  smpl <- c(smpl1,smpl2)
  tmp <- CI(smpl)
  m.smpl[i] <- tmp[2]
  up.low[i,] <- tmp[c(1,3)]
  if(up.low[i,1]>m.true & up.low[i,2]<m.true){
    atari[i] <- 1
  }else{
    atari[i] <- 0
  }
}
mean(atari)
```

```
## [1] 0.949
```

```
n.plot <- n.iter/20
plot(1:n.plot,m.smpl[1:n.plot],pch=20,ylim=c(min(up.low)-5,max(up.low)+5))
abline(h=m.true,col=3)
for(i in 1:n.plot){
  segments(i,up.low[i,1],i,up.low[i,2],col=2)
}
```



良い感じ。

DNA鑑定のための区間推定

体重の区間推定がしたいわけではない。

DNA型ジェノタイプが、たまたま一致する尤度を計算するためには、ジェノタイプ頻度を推定したい。

頻度推定

簡単のために、「あたり vs. はずれ」という枠組みで、成功率を推定することにする。

確率pで当たりが出るくじ引きがある。

n回引いて、k回当たった。

さて、pはいくつか？

その信頼区間は？

```
library(binom)
```

```
## Warning: パッケージ 'binom' はバージョン 4.1.3 の R の下で造られました
```

```
set.seed(3456)
n.sample <- 30
p <- 0.05 # 真の成功率
smpl <- sample(0:1,n.sample,replace=TRUE,prob=c(1-p,p))
smpl
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
# 成功回数
k <- sum(smpl)
k
```

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

「成功率」を「成功と失敗の平均」と考えれば、体重のときと同じことができる。 平均成功率とその信頼区間とみなせば...

```
CI(smpl)
```

```
##       upper        mean       lower
##  0.10150765  0.03333333 -0.03484099
```

信頼区間が「負」を含んでいる。

区間推定をするときには、考慮するべきことがある

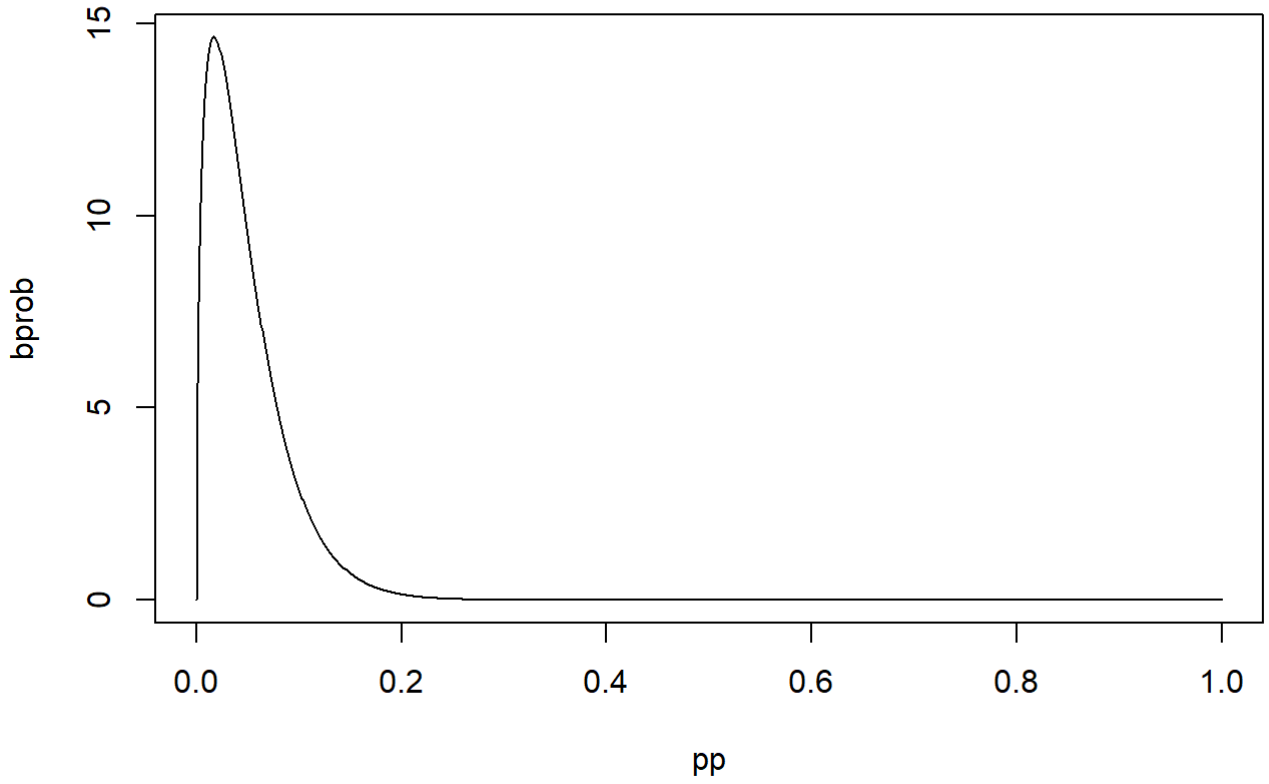
信頼区間に「負」があるのはどうして『いけない』か？

成功率は0から1だと「知っている」から。

ベイズ推定

よく考えたら、二項分布の観察はベータ分布でベイズ推定もできたはず...

```
pp <- seq(from=0,to=1,length=1000)
bprob <- dbeta(pp,k+0.5,n.sample-k+0.5) # Jeffrey's prior
plot(pp,bprob,type="l")
```

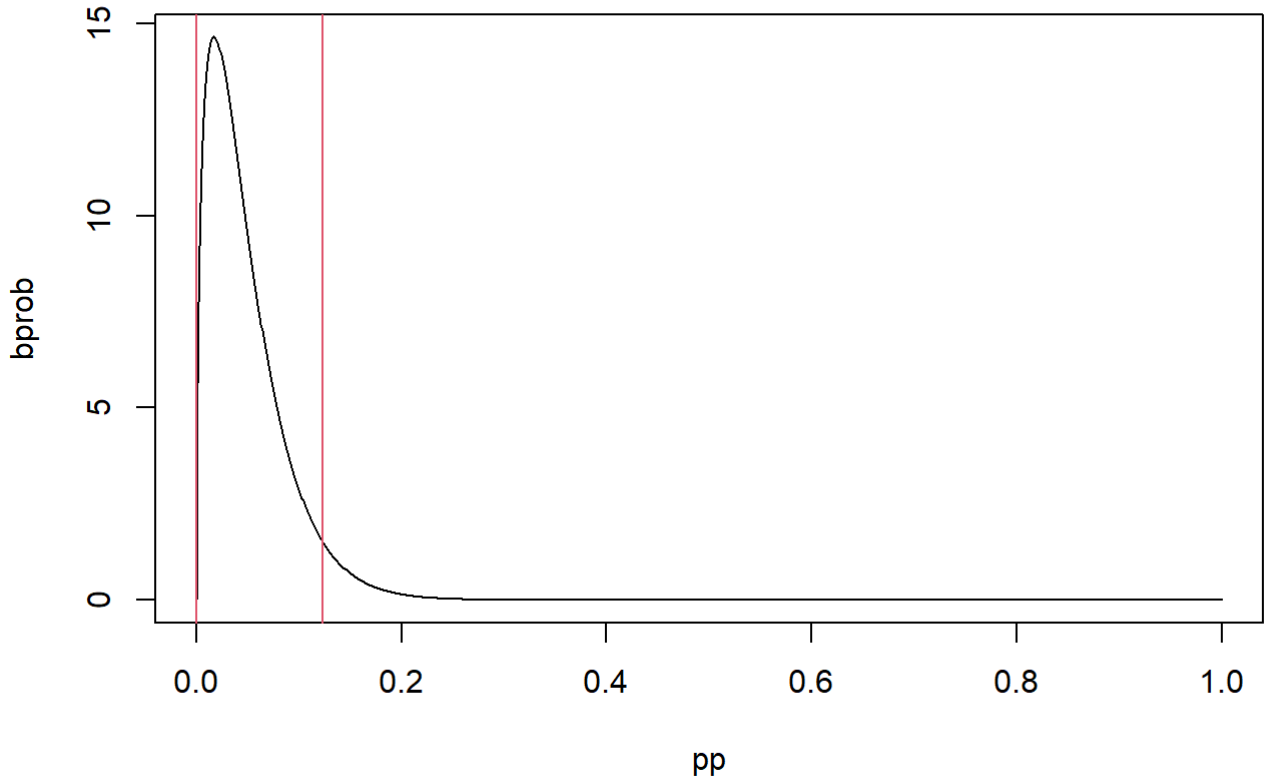


これに基づく「区間推定」もできる

```
b.ci <- binom.confint(sum(smpl),length(smpl),methods="bayes")
b.ci
```

```
##   method x  n      mean        lower     upper
## 1  bayes 1 30 0.0483871 6.903016e-05 0.1231438
```

```
plot(pp,bprob,type="l")
abline(v=b.ci[5:6],col=2)
```



#

ベイズ推定だけではない

実際、こんなに方法がある

```
binom.confint(k,n.sample,methods="all")
```

```
##           method x  n       mean         lower      upper
## 1  agresti-coull 1 30 0.03333333 -8.305484e-03 0.18091798
## 2     asymptotic 1 30 0.03333333 -3.090070e-02 0.09756737
## 3          bayes 1 30 0.04838710  6.903016e-05 0.12314380
## 4        cloglog 1 30 0.03333333  2.494567e-03 0.14513807
## 5          exact 1 30 0.03333333  8.435709e-04 0.17216946
## 6          logit 1 30 0.03333333  4.675346e-03 0.20200244
## 7         probit 1 30 0.03333333  3.475014e-03 0.16637241
## 8        profile 1 30 0.03333333  3.012987e-03 0.13868254
## 9            lrt 1 30 0.03333333  1.961442e-03 0.13868594
## 10     prop.test 1 30 0.03333333  1.742467e-03 0.19053022
## 11        wilson 1 30 0.03333333  5.908590e-03 0.16670391
```

(とはいえ)DNA鑑定に使ってみよう

アレル頻度の推定

３アレルのマーカー(アレル頻度 (A,B,C)=(0.5,0.3,0.2))

６種類のジェノタイプ

Hardy-Weinberg 平衡

$$\begin{pmatrix} X & A & B & C \\ A & 0.25 & 0.3 & 0.2 \\ B & * & 0.09 & 0.12 \\ C & * & * & 0.04 \end{pmatrix}$$

```
n.sample <- 100
pr <- c(0.25,0.09,0.04,0.3,0.2,0.12)
gt <- c("AA","BB","CC","AB","AC","BC")
smpl <- sample(gt,n.sample,replace=TRUE,prob=pr)
n.AA <- length(which(smpl==gt[1]))
n.BB <- length(which(smpl==gt[2]))
n.CC <- length(which(smpl==gt[3]))
n.AB <- length(which(smpl==gt[4]))
n.AC <- length(which(smpl==gt[5]))
n.BC <- length(which(smpl==gt[6]))
```

３アレルの観測本数

```
n.allele <- c(n.AA*2+n.AB+n.AC,n.BB*2+n.AB+n.BC,n.CC*2+n.AC+n.BC)
n.allele
```

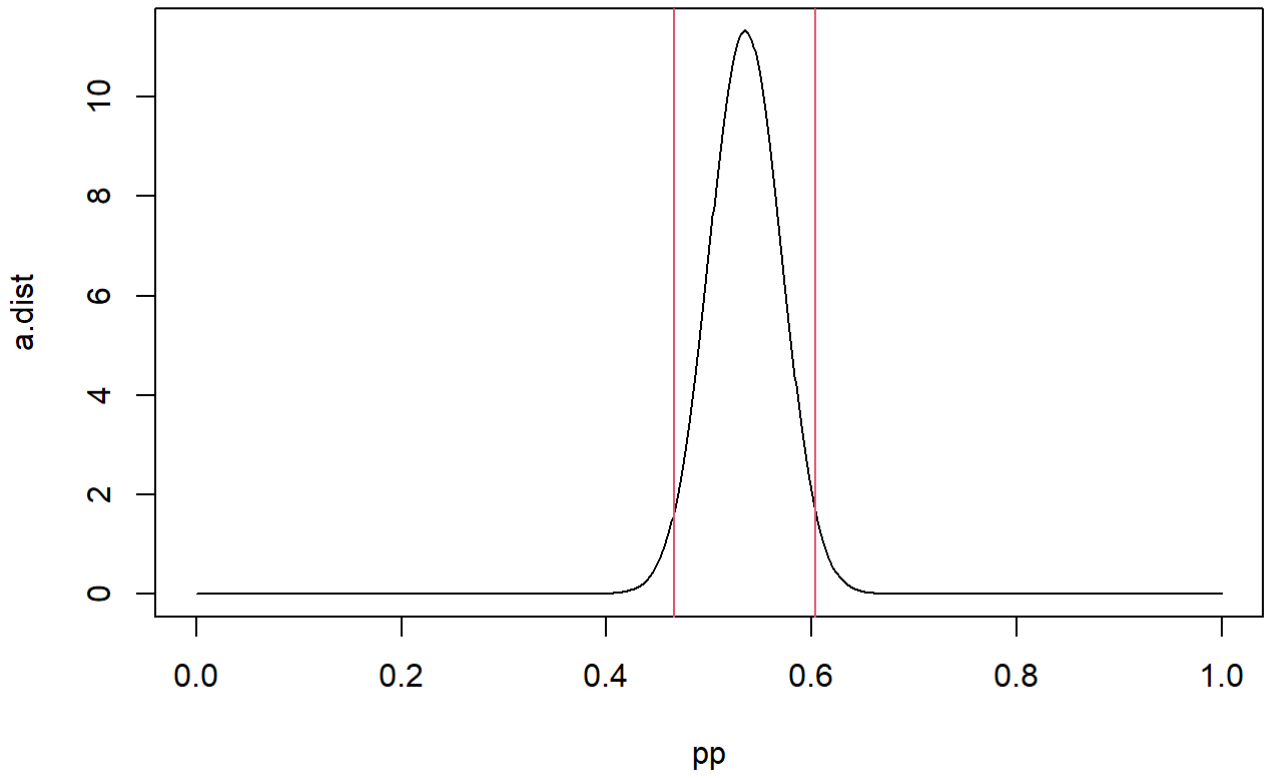
```
## [1] 107  57  36
```

Aアレルの頻度と信頼区間は、A vs non-Aなので、二項分布に基づく方法が使えそう

```
n.A <- n.allele[1]
n.non.A <- sum(n.allele)-n.A
b.ci <- binom.confint(n.A,sum(n.allele),methods="bayes")
b.ci
```

```
##   method   x   n      mean     lower     upper
## 1  bayes 107 200 0.5348259 0.4660202 0.6034381
```

```
pp <- seq(from=0,to=1,length=1000)
a.dist <- dbeta(pp,n.A+0.5,n.non.A+0.5)
plot(pp,a.dist,type="l")
abline(v=b.ci[5:6],col=2)
```



ディプロタイプ頻度の推定

AAディプロタイプの頻度はどうする？

AAの人数を元にすれば、

AA vs. non-AA として、二項分布に基づいて推定できる。

この場合は、HWEを仮定していないことになる。

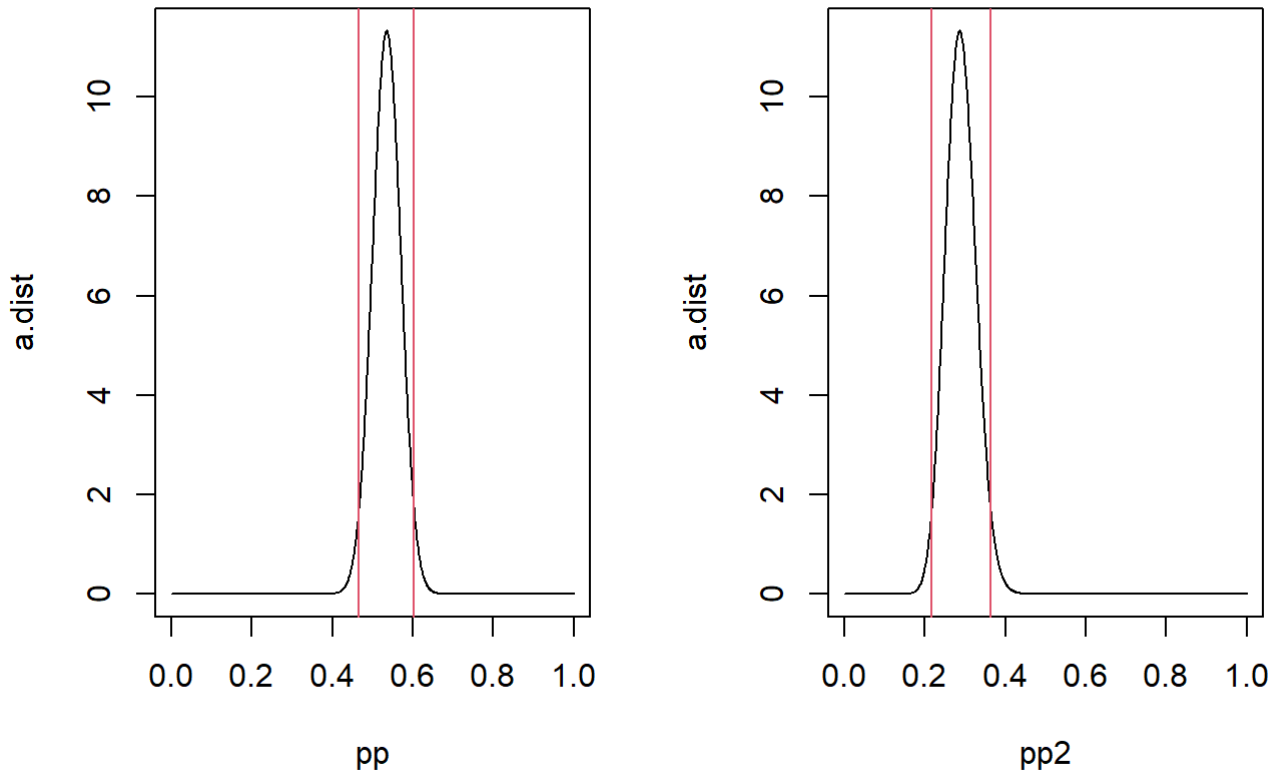
HWEを仮定するべきか、しないべきか、それ「も」問題だ。

が。

HWEを仮定したとして、アレルAの推定頻度を基に、どうやって、AAディプロタイプの信頼区間推定をするのか？

AAの頻度はアレル頻度の２乗なので…

```
par(mfcol=c(1,2))
plot(pp,a.dist,type="l")
abline(v=b.ci[5:6],col=2)
pp2 <- pp^2
plot(pp2,a.dist,type="l")
abline(v=b.ci[5:6]^2,col=2)
```




```
par(mfcol=c(1,1))
```

これでよいのか？

確認してみる。

```
af <- 0.2 # アレル頻度
gf <- af^2 # ホモジェノタイプ頻度

n.iter <- 1000
n.sample <- 100
ret <- matrix(0,n.iter,2) # アレル頻度の信頼区間

for(i in 1:n.iter){
  smpl <- sample(2:0,n.sample,replace=TRUE,prob=c(af^2,2*af*(1-af),(1-af)^2))
    s.af <- sum(smpl)
    tmp <- binom.confint(s.af,n.sample*2,methods="bayes")
    ret[i,1] <- unlist(tmp[5])
    ret[i,2] <- unlist(tmp[6])
}

length(which(ret[,1]^2 < gf & ret[,2]^2 > gf)) / n.iter
```

```
## [1] 0.947
```

ABの頻度はどうする？

アレルAの頻度とアレルBの頻度をそれぞれ求める？

アレルAの頻度が高いとき、アレルBの頻度は低いはず。

お互いに影響し合っているので、別々に推定したり、別々の信頼区間を考えるのはまずい。

多項分布のベイズ推定はディリクレ分布

A + B + C = 1 を満足する自由度２の分布

```
library(MCMCpack)
```

```
##  要求されたパッケージ coda をロード中です
```

```
##  要求されたパッケージ MASS をロード中です
```

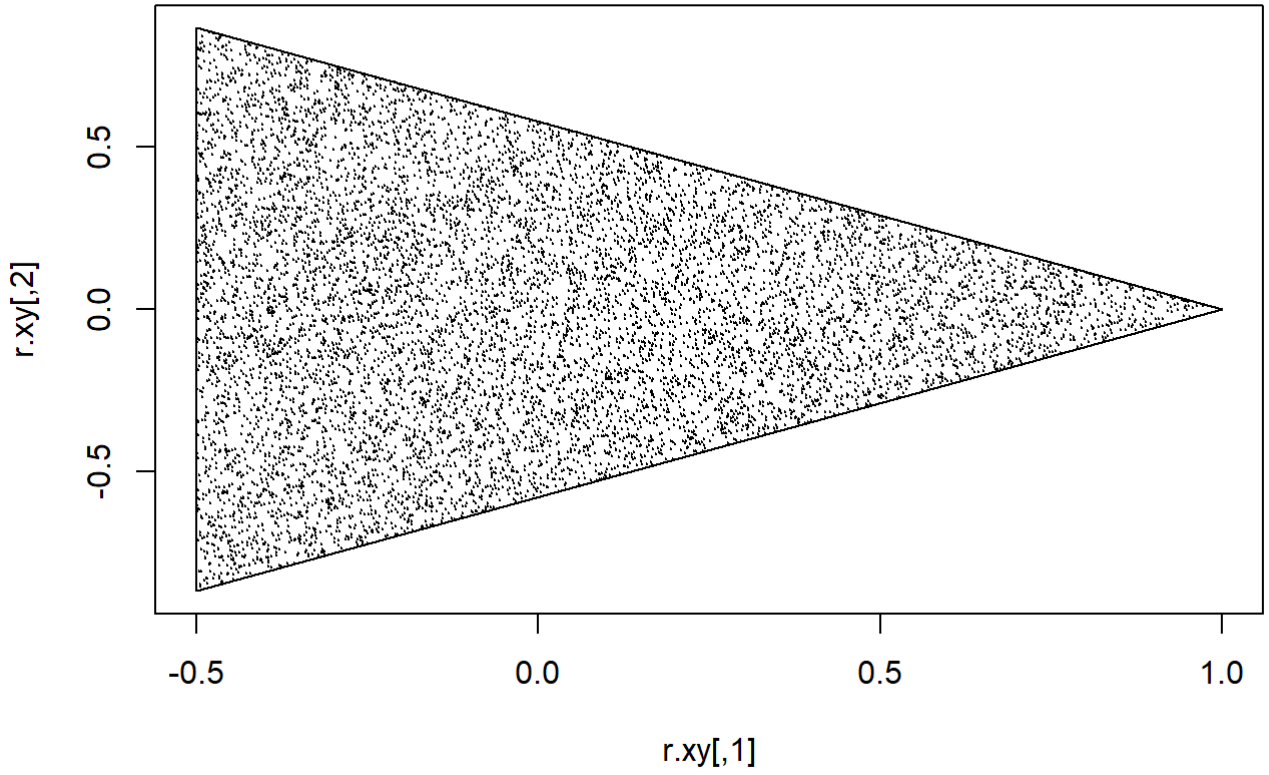
```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2022 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
n.pt <- 10^4
r.pt <- rdirichlet(n.pt,rep(1,3))
M <- matrix(c(1,0,-1/2,sqrt(3)/2,-1/2,-sqrt(3)/2),2,3)
r.xy <- t(M %*% t(r.pt))
plot(r.xy,xlim=c(-0.5,1),ylim=c(-sqrt(3)/2,sqrt(3)/2),pch=20,cex=0.1)
segments(M[1,1],M[2,1],M[1,2],M[2,2])
segments(M[1,2],M[2,2],M[1,3],M[2,3])
segments(M[1,3],M[2,3],M[1,1],M[2,1])
```

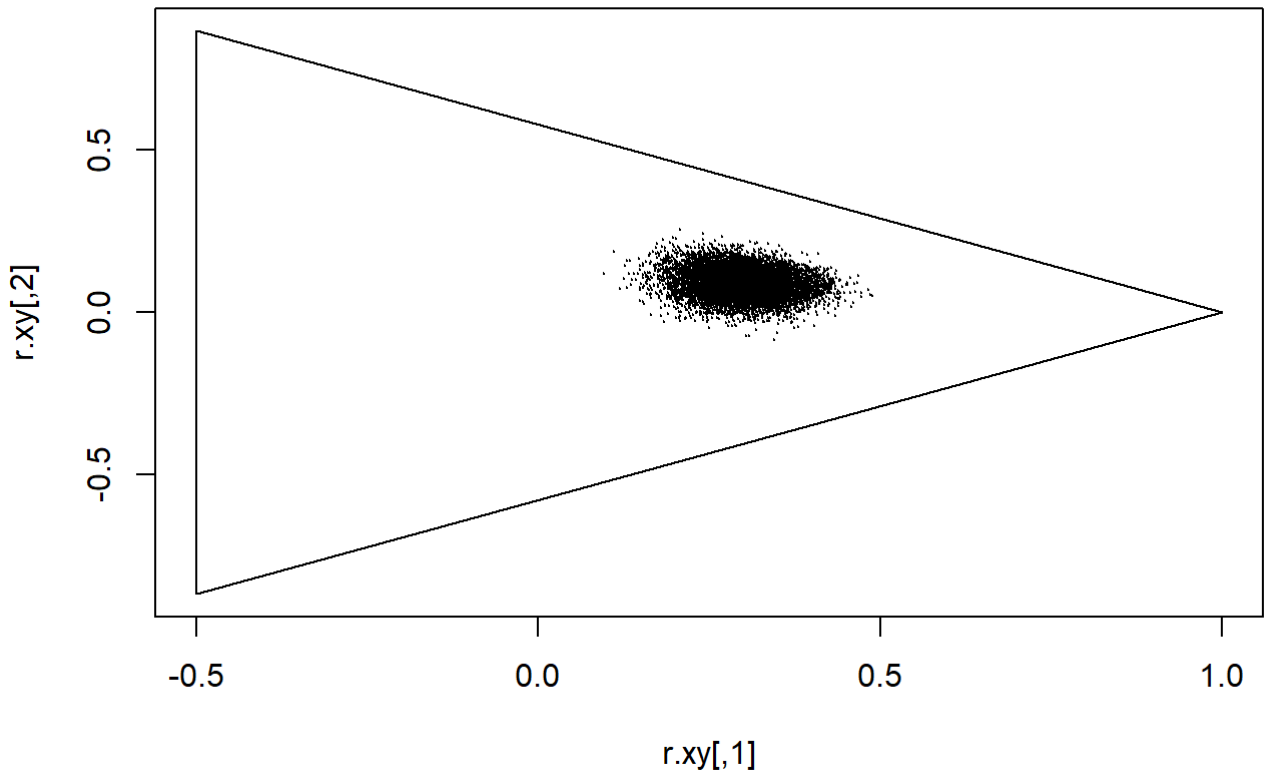


観測データに基づいて推定してみる

```
n.allele
```

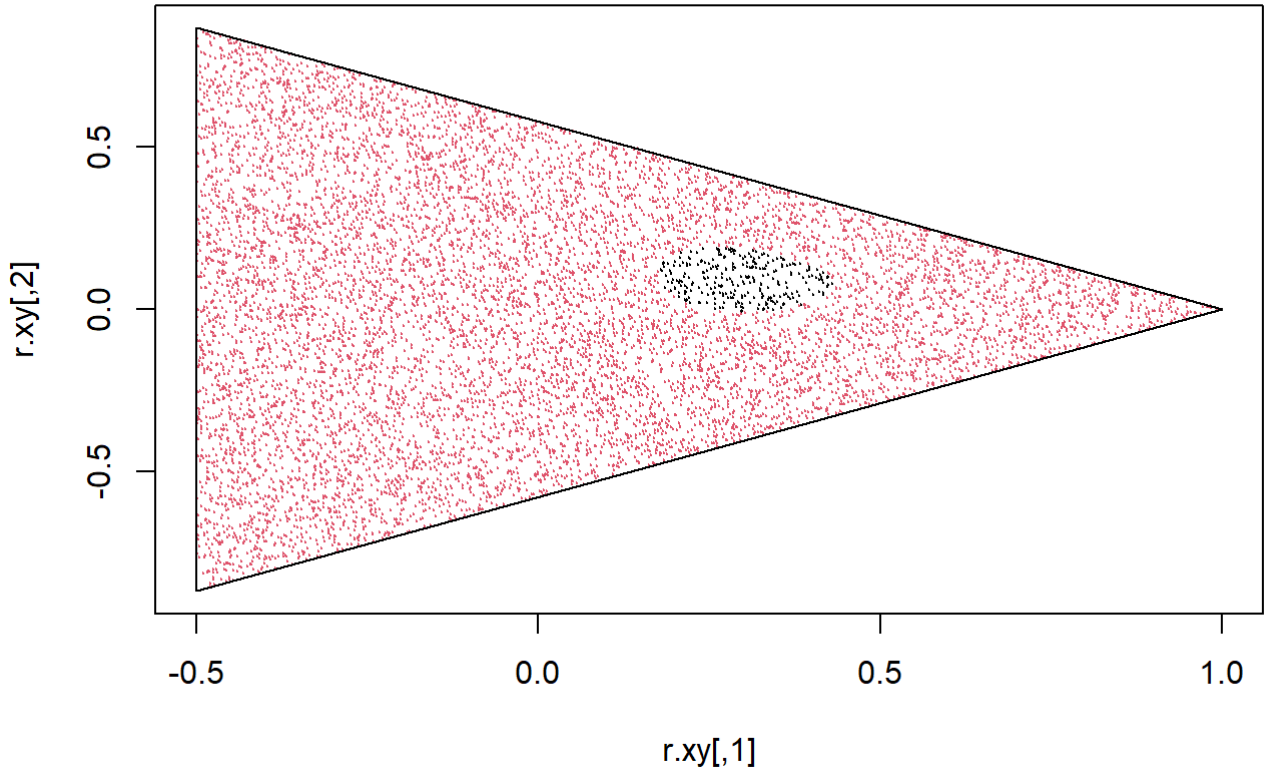
```
## [1] 107  57  36
```

```
r.pt <- rdirichlet(n.pt,n.allele+rep(0.5,3))

r.xy <- t(M %*% t(r.pt))
plot(r.xy,xlim=c(-0.5,1),ylim=c(-sqrt(3)/2,sqrt(3)/2),pch=20,cex=0.1)
segments(M[1,1],M[2,1],M[1,2],M[2,2])
segments(M[1,2],M[2,2],M[1,3],M[2,3])
segments(M[1,3],M[2,3],M[1,1],M[2,1])
```



信頼「範囲」はどうなっているか?

```
r.pt <- rdirichlet(n.pt,rep(1,3))
r.xy <- t(M %*% t(r.pt))
d.pt <- ddirichlet(r.pt,n.allele+rep(0.5,3))
d.pt.sorted <- sort(d.pt)
d.pt.sorted.cum <- cumsum(d.pt.sorted)
s <- which(d.pt.sorted.cum < sum(d.pt.sorted)*0.05)
s.max <- s[length(s)]
thres <- d.pt.sorted[s.max]
selected <- which(d.pt<thres)
plot(r.xy,xlim=c(-0.5,1),ylim=c(-sqrt(3)/2,sqrt(3)/2),pch=20,cex=0.1)
points(r.xy[selected,],xlim=c(-0.5,1),ylim=c(-sqrt(3)/2,sqrt(3)/2),pch=20,cex=0.1,col=2)

segments(M[1,1],M[2,1],M[1,2],M[2,2])
segments(M[1,2],M[2,2],M[1,3],M[2,3])
segments(M[1,3],M[2,3],M[1,1],M[2,1])
```



乱数を使ってみては？

アレルA,B,Cの頻度分布がわかったので、そこから乱数を発生させて、ディプロタイプABの頻度分布を作成してみる。

```
n.iter <- 10^4
r.allele <- rdirichlet(n.iter,n.allele+rep(0.5,3))
r.genotype <- 2 * r.allele[,1] * r.allele[,2]
quantile(r.genotype,c(0.025,0.975))
```

```
##       2.5%      97.5%
## 0.2531278 0.3514797
```

DNA鑑定での尤度比

容疑者のジェノタイプが現場の試料のそれと一致したとき。

たまたま、一致したのか、同一人物だから一致したのかは、それぞれの仮説の尤度の比で計算する。

同一人物の場合の尤度は１だから、たまたまの場合の尤度を計算すればよい。

マーカーごとの観察が独立とみなせるならば、個々のマーカーでの尤度の積。

複数のマーカー、それぞれのマーカーのアレル数を適当に与えてシミュレーションしてみる。

ディプロタイプのデータベースをシミュレーション作成する。

```
n.marker <- 5
n.allele <- sample(3:6,n.marker,replace=TRUE)
f.allele <- list()
for(i in 1:n.marker){
  f.allele[[i]] <- rdirichlet(1,rep(1,n.allele[i]))
}
gt.database <- list()
n.sample <- 1000
cnt.alleles <- list()
for(i in 1:n.marker){
  tmp <- sample(1:n.allele[i],n.sample*2,replace=TRUE,prob=f.allele[[i]])
  cnt.alleles[[i]] <- 0
  for(j in 1:n.allele[i]){
    cnt.alleles[[i]] <- c(cnt.alleles[[i]],length(which(tmp==j)))
  }
  cnt.alleles[[i]] <- cnt.alleles[[i]][-1]
}
```

すべてのマーカーで、第１、第２アレルのヘテロ型であるときの尤度の信頼区間をシミュレーションで算出する。

```
n.iter <- 10^6
r.genotypes <- matrix(0,n.iter,n.marker)
for(i in 1:n.marker){
  r.allele <- rdirichlet(n.iter,cnt.alleles[[i]]+rep(0.5,n.allele[i]))
  r.genotype <- 2 * r.allele[,1] * r.allele[,2]
  r.genotypes[,i] <- r.genotype
}
r.genotype.all <- apply(log(r.genotypes),1,sum)
exp(quantile(r.genotype.all,c(0.025,0.975)))
```

```
##          2.5%         97.5%
## 1.936253e-08 5.573304e-08
```

DNAの得られない構成員がいる家系での尤度比推定

親情報のない個人を集団標本からブートストラップしつつ、家系図に沿って1/2乱択