生物資源解析学演習 Lecture 2

確率分布と最尤推定の初歩

北門利英

2019 年 *10* 月 *15* 日

目次

1	確率分布の復習	1
1.1	R における確率分布に関連した表記	1
1.2	2 項分布	1
1.3	ポアソン分布	7
1.4	一様分布	11
2	最尤推定法	17
2.1	準備：最適化	17
2.2	2 項分布のパラメータの最尤推定	20

1 確率分布の復習

1.1 R における確率分布に関連した表記

R には統計解析やモデリングで利用する種々の確率分布に対応した関数が用意されている．これらは，各種の確率分布に従う乱数の発生の他，確率関数 (確率密度関数)，累積確率分布，そしてパーセント点の評価に分類される．例えば，2 項分布 $Bin(N, p)$ を例にとると以下の通り．これと同様にして，頭文字を r，d，p，q とすることで他の確率分布にも適用できる．

```
rbinom(n=乱数の数, size=N, prob=p) #独立な乱数の生成
dbinom(x=実現値の値, size, prob, log = FALSE) #確率関数の値
pbinom(q, size, prob) #累積確率の値
qbinom(p, size, prob) #パーセント点の値
```

表1　それぞれの確率分布に対応した関数

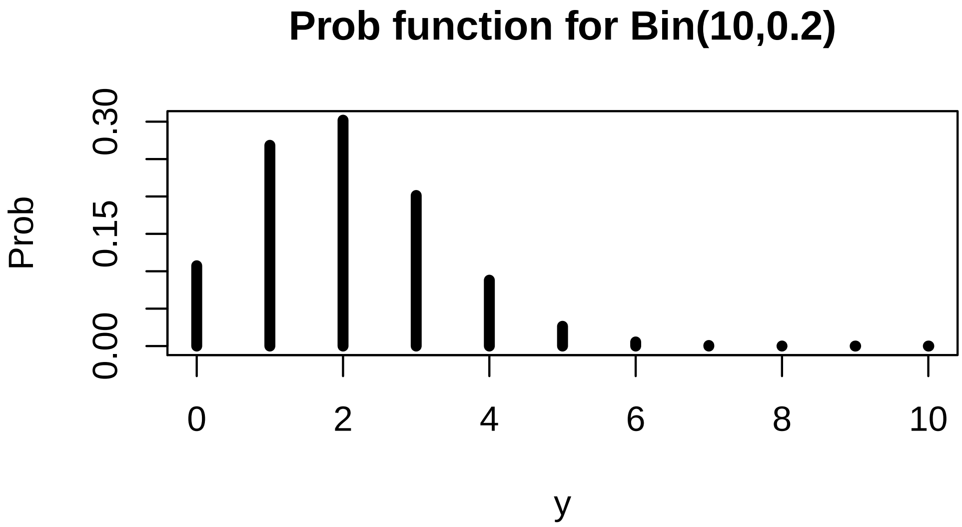
離散型確率分布	
`_binom( , N, p)`	2 項分布 $Bin(N,p)$
`_pois( , lambda)`	ポアソン分布 $Pois(\lambda)$
`_nbinom( , N, p)`	2 項分布 $NB(N,p)$
`_geom( , p)`	幾何分布 $Geom(p)$
`_hyper( , m=M, n=N-M, k=n)`	超幾何分布 $HG(N,M,n)$
`_multinom( , N, p)`	多項分布 $Multinom(N,p)$
連続型確率分布	
`_unif( , min=a, max=b)`	一様分布 $U(a,b)$
`_norm( , mu, sigma)`	正規分布 $N(\mu,\sigma^2)$
`_lnorm( , mu, sigma)`	対数正規分布 $LN(\mu,\sigma^2)$
`_gamma( , shape=a, scale=b)`	ガンマ分布 $Ga(a,b)$ with $E[Y]=ab$
`_exp( , scale=sigma)`	ガンマ分布 $Exp(\sigma)$ with $E[Y]=\sigma$
`_chisq( , df=n)`	カイ 2 乗分布 $\chi^2(n)$
`_beta( , shape1=a, shape2=b)`	ベータ分布 $\beta(n_1,n_2)$
`_t( , df=n)`	t 分布 $t(n)$
`_f( , df1=n1, df2=n2)`	t 分布 $F(n_1,n_2)$

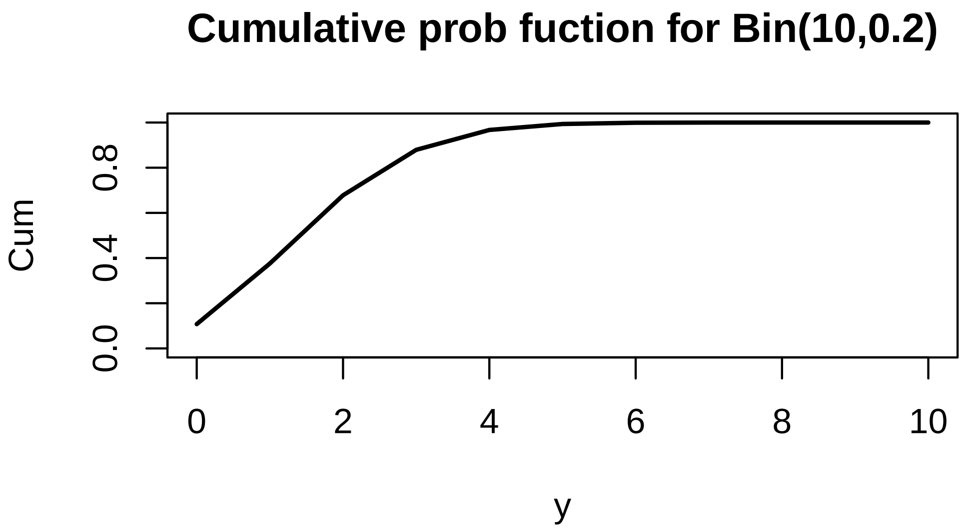
1.2　2 項分布

1.2.1　確率分布 (確率関数) と累積確率分布関数のグラフ化

- $Y \sim Bin(10,0.2)$ に対して $Pr(Y=y)$ および $P(Y<=y)(y=0,1,...,10)$ の計算

```
N <- 10
p <- 0.2
Yrange <- 0:10
Prob <- dbinom(Yrange, N, p)
Cum <- pbinom(Yrange, N, p)
par(mfrow=c(2,1))
plot(Yrange, Prob, type="h", lwd=5, xlab="y",
     main="Prob function for Bin(10,0.2)")
plot(Yrange, Cum, type="l", lwd=2, xlab="y",
     main="Cumulative prob fuction for Bin(10,0.2)" ,ylim=c(0,1))
```





- $Y \ Bin(20, 0.1)$ に対して $Pr(Y = y)$ および $P(Y<=y) (y=0,1,…,20)# の計算

Do by yourself!

```
plot(Yrange, Prob, type="h", lwd=5, xlab="y",
     main=paste("2 項分布 Bin(",N,",",p,") の確率関数"))
```

1.2.2　2 項分布に従う例題：哺乳類の雌の個体数変動

Notation

- N_t: t 年に初めの雌の数
- S_t: t 年における生き残り雌の数
- P_t: t 年の生き残り雌のうち，妊娠雌の数 (= 出産数)
- B_t: 出産数 P_t 個のうち雌の数
- 1 年あたりの死亡率 d (=0.2)

- 1 年あたりの妊娠率 b (=0.4) [雌 1 個体から 1 個体出産]

Stochastic nature

$$\begin{array}{lcl} S_t|N_t & \sim & Bin(N_t, 1-d) \\ P_t|S_t & \sim & Bin(S_t, b) \\ B_t|P_t & \sim & Bin(P_t, 0.5) \\ N_{t+1} = S_t + B_t & & \end{array}$$

- 1 年目の雌個体数 Nf (=5)

- 20 年予測 (TT=10)

- シミュレーション繰り返し数 Nsim (=100)

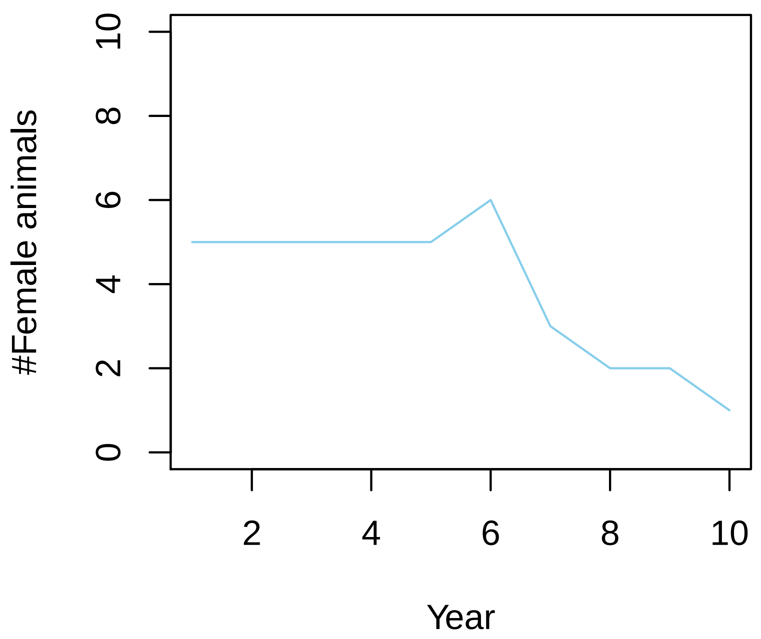
```
#パラメータ値の設定
d <- 0.2
b <- 0.4
TT <- 10
TT1 <- TT-1
Nsim <- 100
Nf <- array(0, c(Nsim, TT))
Nf[,1] <- 5
```

```
#1 回だけの計算
for(t in 1:TT1) {
  Surv <- rbinom(1, Nf[1, t], 1-d)
  Preg <- rbinom(1, Surv, b)
  Birthf <- rbinom(1, Preg, 0.5)
  Nf[1, t+1] <- Surv + Birthf
```

```
}
Nf[1,]
```

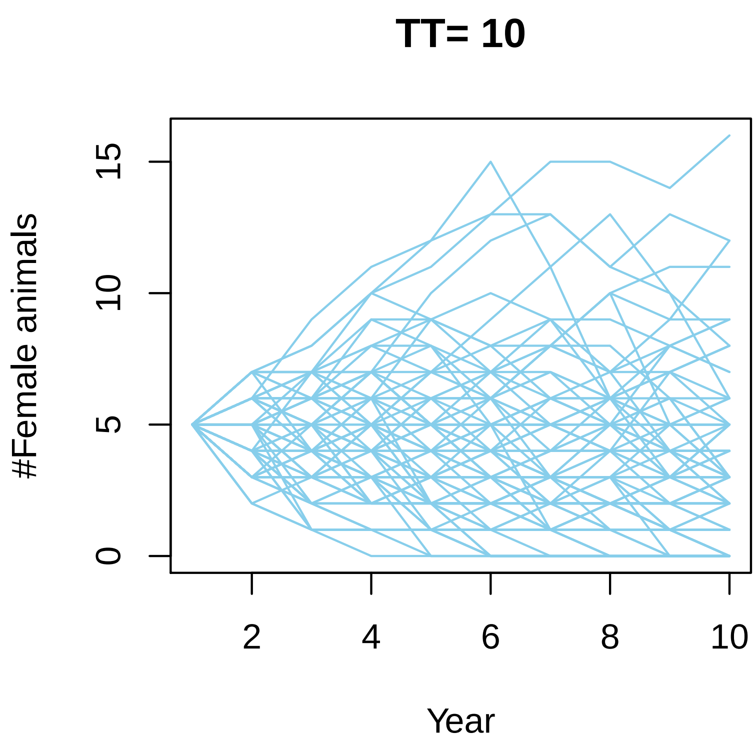
```
 [1] 5 5 5 5 5 6 3 2 2 1
```

```
plot(Nf[1,], type="l", col="skyblue", xlab="Year", ylab="#Female animals", ylim=c(0,10))
```




```
#計算の繰り返し
for(i in 1:Nsim){
 for(t in 1:TT1){
  Surv <- rbinom(1, Nf[i, t], 1-d)
  Preg <- rbinom(1, Surv, b)
  Birthf <- rbinom(1, Preg, 0.5)
  Nf[i, t+1] <- Surv + Birthf
 }
}
plot(Nf[1,], type="l", col="skyblue", ylim=c(0,max(Nf)),
     xlab="Year", ylab="#Female animals", main=paste("TT=",TT))
for(i in 1:Nsim){
```

```
 points(Nf[i,], type="l", col="skyblue")
}
```



10 年後に絶滅する確率は？

```
paste("Extinction prob = ", mean(Nf[,TT] == 0))
```

```
[1] "Extinction prob =  0.21"
```

1.2.3　HW1 上記の設定で以下の考察のポイントについて検討しなさい

- 年間死亡率が変わると絶滅確率も変化するでしょう．その関係をグラフで表現してみてください

- 年間妊娠率が変わると絶滅確率も変化するでしょう．その関係をグラフで表現してみてください

- 個体群は，いずれは絶滅するでしょうか？例えば TT=20,50,100,1000 などと設定し，考察

してみてください

1.3　ポアソン分布

rpois(n=乱数の数, lambda) #独立な乱数の生成

dpois(x=実現値の値, lambda, log = FALSE) #確率関数の値

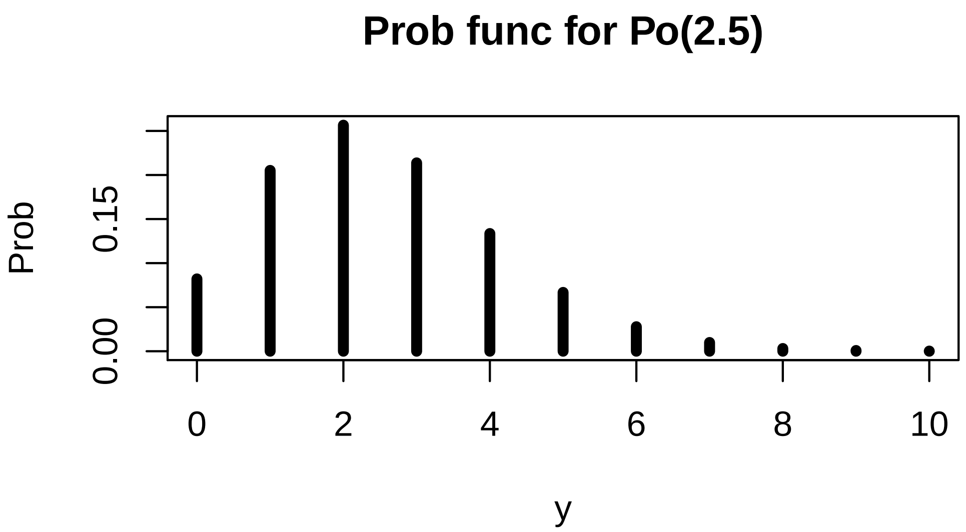
ppois(q, lambda=lambda) #累積確率の値

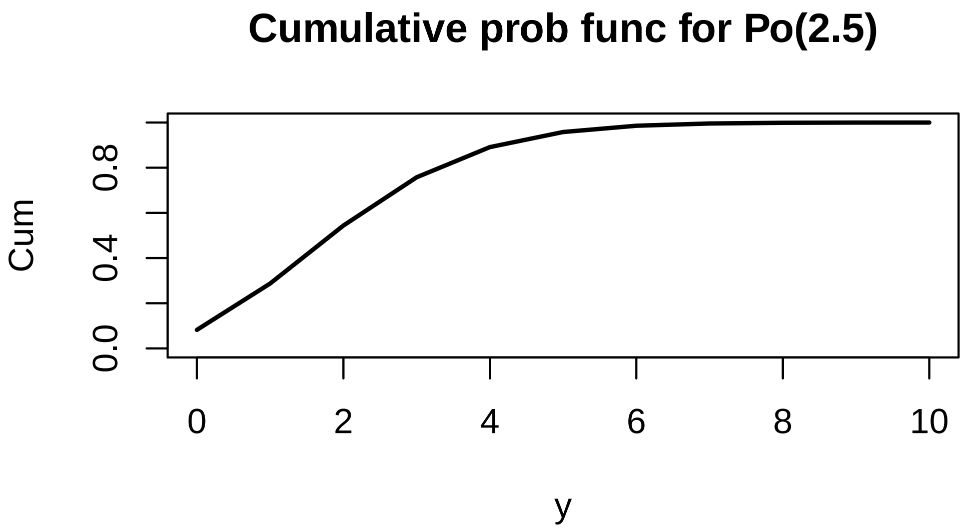
qpois(p, lambda=lambda) #パーセント点の値

1.3.1　確率分布 (確率関数) と累積確率分布関数のグラフ化

$Y \sim Po(2.5)$ に対して $Pr(Y = y)$ および $P(Y <= y)(y = 0, 1, ..., 10)$ の計算

```
lambda <- 2.5
Yrange <- 0:10
Prob <- dpois(Yrange, lambda)
Cum <- ppois(Yrange, lambda)
par(mfrow=c(2,1))
plot(Yrange, Prob, type="h", lwd=5, xlab="y",
     main="Prob func for Po(2.5)")
plot(Yrange, Cum, type="l", lwd=2, xlab="y",
     main="Cumulative prob func for Po(2.5)", ylim=c(0,1))
```





$Y \sim Po(5)$ に対して $Pr(Y = y)$ および $P(Y <= y)(y = 0, 1, ..., 20)$ の計算

Do by yourself!

1.3.2　HW2 雌個体数動態の発展

$Y \sim Po(2.5)$ と $Y \sim Bin(50, 0.05)$ の比較

Do by yourself!

1.3.3 HW3 雌個体数動態の発展

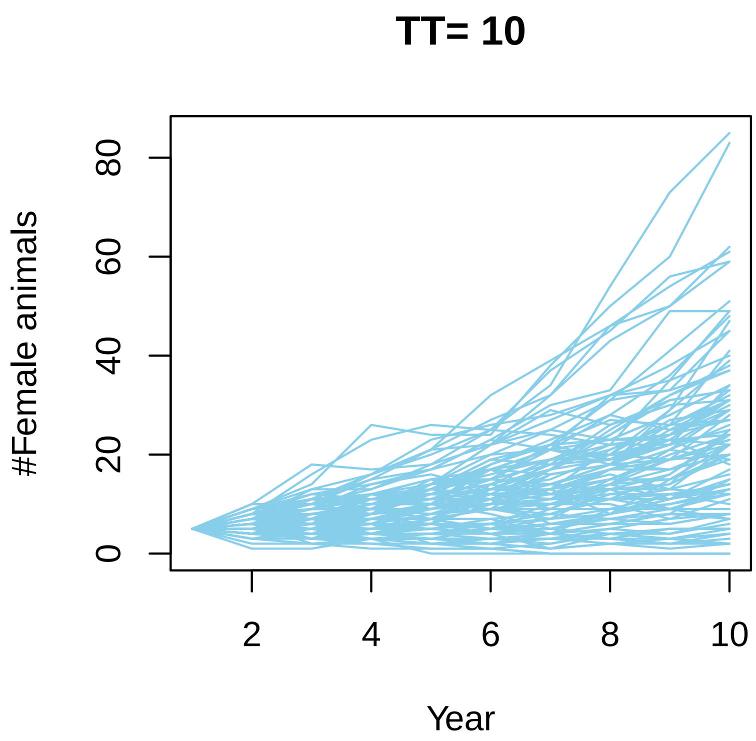
- 上記の雌個体数の動態では雌 1 個体から 1 個体出産と仮定した

- 次に雌 1 個体から複数個体が生まれる可能性を考慮したい

- 出生数が 1+Po(0.2) に従うとき，に雌個体数の計算はどのように行えばよいか

- またこの場合，絶滅確率に変化はあるか？(T=20, 50 くらいで計算)

```
d <- 0.2
b <- 0.4
lambda <- 0.2
TT <- 10; TT1<-TT-1
Nsim <- 100
Nf <- array(0, c(Nsim, TT))
Nf[,1] <- 5
for(i in 1:Nsim){
 for(t in 1:TT1){
  Surv <- rbinom(1, Nf[i, t], 1-d)
  Preg <- rbinom(1, Surv, b)
  if(Preg==0){Birthf <- 0} else {Birthf <- Preg + sum(rpois(Preg, lambda))}
  Nf[i, t+1] <- Surv + Birthf
 }
}
plot(Nf[1,], type="l", col="skyblue", xlab="Year", ylab="#Female animals",
     ylim=c(0,max(Nf)), main=paste("TT=",TT))
for(i in 1:Nsim){
 points(Nf[i,], type="l", col="skyblue")
}
```

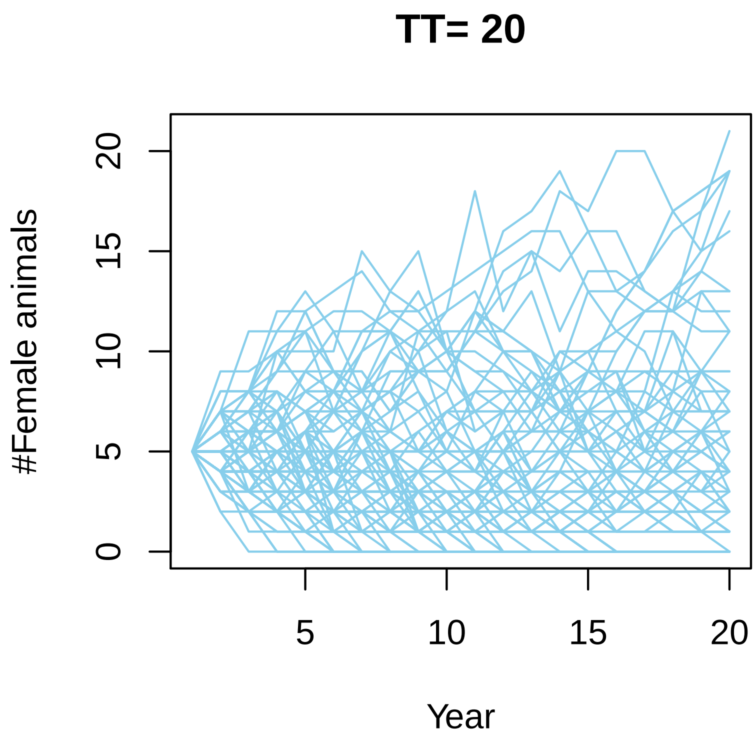



```r
paste("Extinction prob = ", mean(Nf[,TT] == 0))
```

```
[1] "Extinction prob =  0.03"
```

```r
TT <- 20; TT1<-TT-1
Nf <- array(0, c(Nsim, TT))
Nf[,1] <- 5
for(i in 1:Nsim){
 for(t in 1:TT1){
  Surv <- rbinom(1, Nf[i, t], 1-d)
  Preg <- rbinom(1, Surv, b)
  if(Preg==0){Birthf <- 0} else {
    Birth <- Preg + sum(rpois(Preg, lambda))
    Birthf <- rbinom(1,Birth,0.5)
  }
  Nf[i, t+1] <- Surv + Birthf
 }
}
plot(Nf[1,], type="l", col="skyblue", xlab="Year", ylab="#Female animals",
```

```r
     ylim=c(0,max(Nf)), main=paste("TT=",TT))
for(i in 1:Nsim){
 points(Nf[i,], type="l", col="skyblue")
}
```




```r
paste("Extinction prob = ", mean(Nf[,TT] == 0))
```

```
[1] "Extinction prob =  0.4"
```

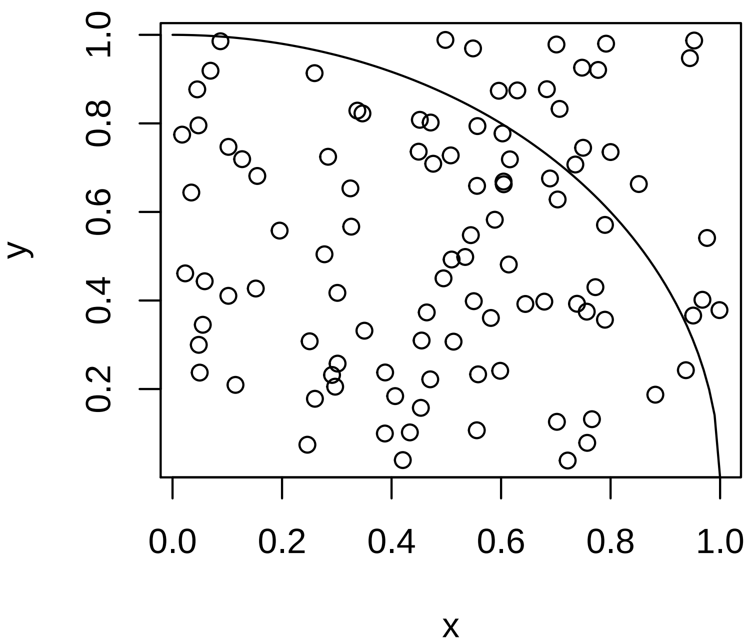
1.4 一様分布

$$Y \sim U(a,b) f(y) = \frac{1}{b-a} \quad \text{for } a < y < b$$

1.4.1　一様乱数を用いたモンテカルロ法による円周率の計算

$$X \sim U(0,1) Y \sim U(0,1) Pr(X^2 + Y^2 \leq 1) = \pi/4$$

```
Nit <- 100
x <- runif(Nit, 0, 1)
y <- runif(Nit, 0, 1)
plot(x, y)
curve( sqrt(1-x^2), xlim=c(0,1), add=T)
```




```
4*mean(x^2+y^2 <= 1)
```

```
[1] 3.2
```

関数化すると

```
Calc_pi <- function(Nit, Fig=F)
{
 x <- runif(Nit, 0, 1)
```

```r
 y <- runif(Nit, 0, 1)
 if(Fig==T){
  plot(x, y, main=paste("#interations = ", Nit), pch=19, cex=0.6, col="gray")
  curve( sqrt(1-x^2), xlim=c(0,1), add=T, col="red")
 }
 4*mean(x^2+y^2 <= 1)
}
```

```r
par(mfrow=c(2,2))
Calc_pi(100, Fig=T)
```

```
[1] 3.16
```

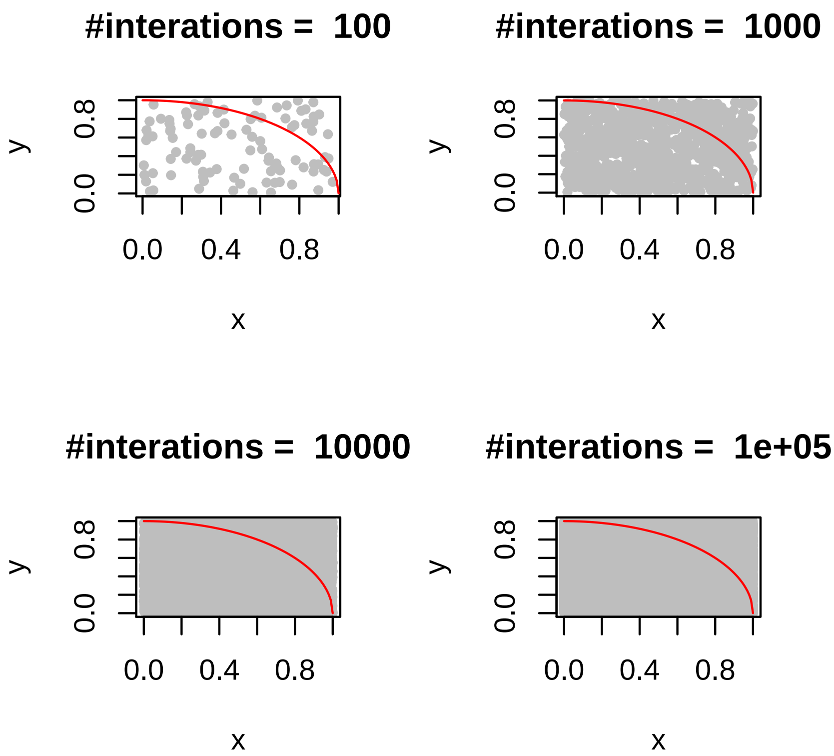
```r
Calc_pi(1000, Fig=T)
```

```
[1] 3.18
```

```r
Calc_pi(10000, Fig=T)
```

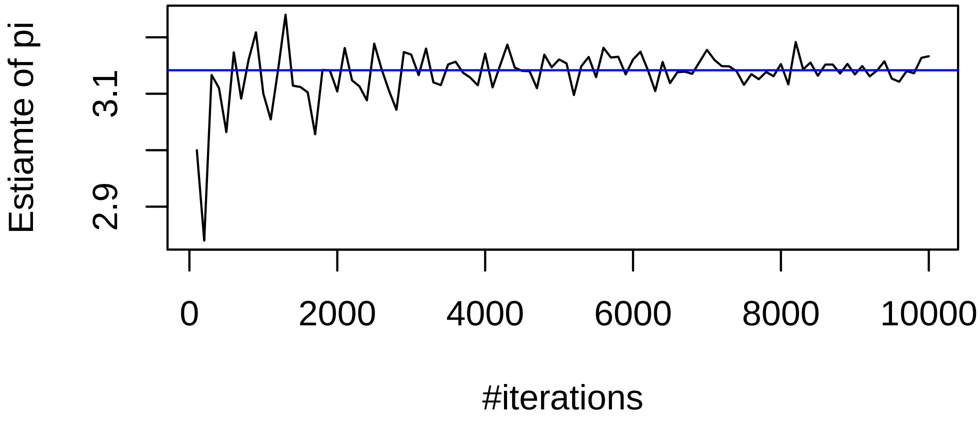
```
[1] 3.1464
```

```r
Calc_pi(100000, Fig=T)
```



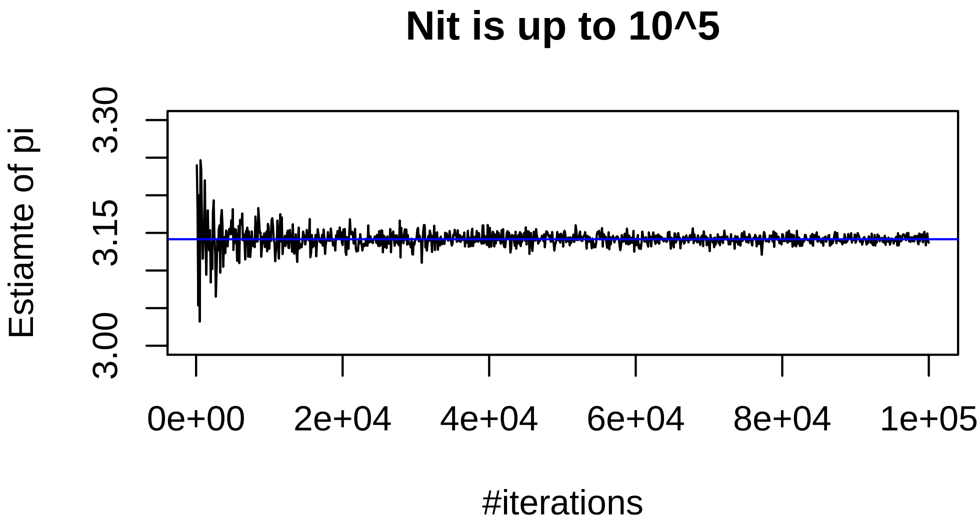

```
[1] 3.14368
```

```r
Nit_vec <- seq(100,10000,100)
Pi_vec <- sapply(Nit_vec, Calc_pi)

plot(Nit_vec, Pi_vec, type="l", xlab="#iterations", ylab="Estiamte of pi")
abline(pi, 0, col="blue")
```

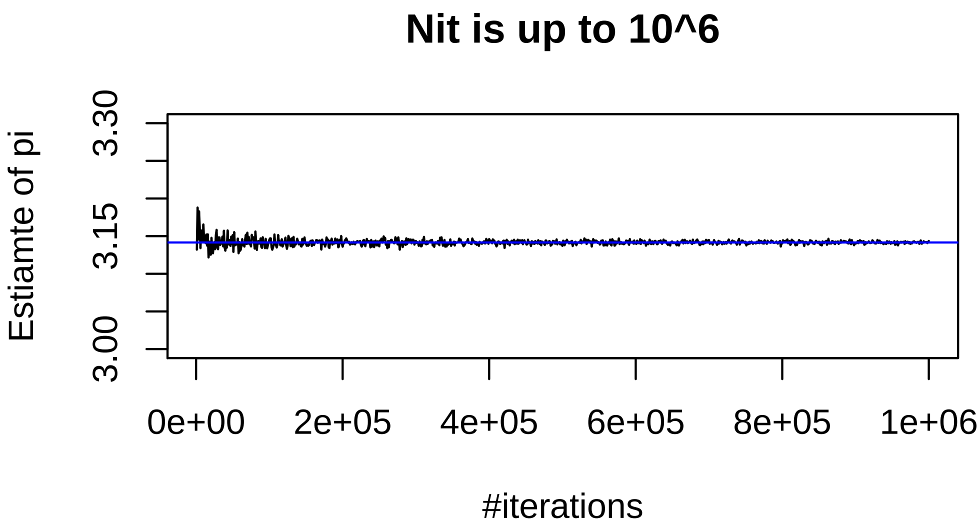


1.4.2 HW4 繰り返し数を 10^5 までとするとどうでしょうか？

```
Nit_vec <- seq(100,100000,100)
Pi_vec <- sapply(Nit_vec, Calc_pi)

plot(Nit_vec, Pi_vec, type="l", xlab="#iterations", ylab="Estiamte of pi",
     main="Nit is up to 10^5", ylim=c(3,3.3))
abline(pi, 0, col="blue")
```




```r
Nit_vec <- seq(1000,1000000,1000)
Pi_vec <- sapply(Nit_vec, Calc_pi)

plot(Nit_vec, Pi_vec, type="l", xlab="#iterations", ylab="Estiamte of pi",
     main="Nit is up to 10^6", ylim=c(3,3.3))
abline(pi, 0, col="blue")
```

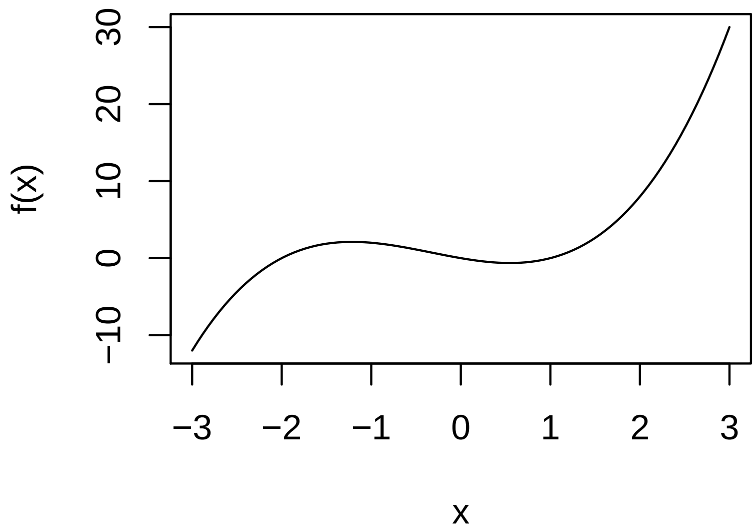


2　最尤推定法

2.1　準備：最適化

3 次関数の最小化を通して R による最適化法に慣れましょう!

```
#関数の定義の仕方（利用する関数 function）
f <- function(x){x^3 + x^2 - 2*x }

#関数の描き方
curve(f, xlim=c(-3,3))
```




```
#関数の描き方 (another one)
#x <- seq(-3,3,0.01)
#plot(x, f(x), type="l")

#関数の最小化（利用する関数 optim）
#optim(初期値, 関数名, 最適化の方法 (通常は BFGS ニュートン法))
optim(-2, f, method="BFGS")
```

$par

```
[1] -2.612088e+21

$value
[1] -1.782228e+64

$counts
function gradient
       6        6

$convergence
[1] 0

$message
NULL
```

```r
optim(0, f, method="BFGS")
```

```
$par
[1] 0.5485839

$value
[1] -0.6311303

$counts
function gradient
      12        6

$convergence
[1] 0

$message
NULL
```

```r
optim(1, f, method="BFGS")
```

```
$par
[1] 0.5485839
```

```
$value
[1] -0.6311303

$counts
function gradient
      11        6

$convergence
[1] 0

$message
NULL
```

```r
#定義域を [0,1] に制限しての関数の最小化
optim(-2, f, method="L-BFGS-B", lower=0, upper=1)
```

```
$par
[1] 0.5485836

$value
[1] -0.6311303

$counts
function gradient
       7        7

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```r
#定義域を [0,1] に制限しての関数の最小化 (長くなるけど，こちらの方がお勧め)
f <- function(logitx){
x <- exp(logitx)/(1+exp(logitx))
# x <- 1/(1+exp(-logitx))
x^3 + x^2 - 2*x
}
```

```r
res <- optim(-5, f, method="BFGS")
res$par
```

```
[1] 0.1949502
```

```r
exp(res$par) /(1+exp(res$par))
```

```
[1] 0.5485838
```

2.2 2 項分布のパラメータの最尤推定

2 項分布に関する R の関数を覚え，計算に使えるようになりましょう！

2 項分布に関する R の関数 dbinom(観測値の値 (y)，試行回数 (N)，確率 (p)，log=T or F) rbinom(乱数の数，試行回数 (N)，確率 (p))

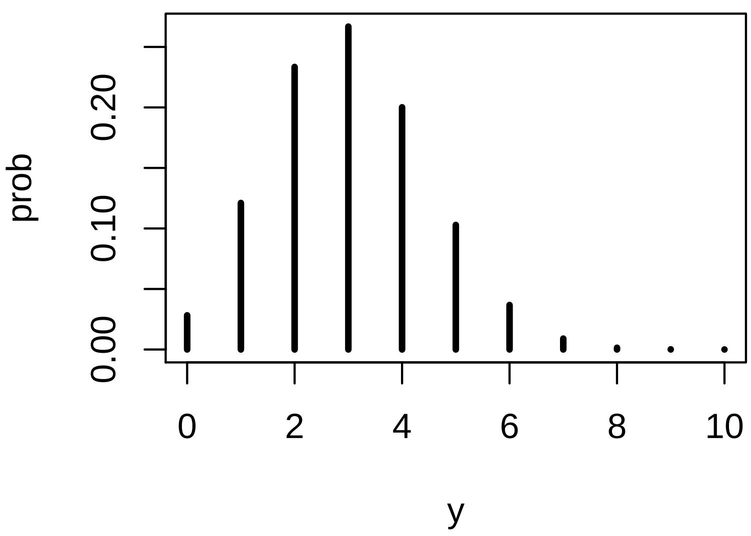
```r
#R による確率の計算
N <- 10
p <- 0.3
dbinom(3, N, p)
```

```
[1] 0.2668279
```

```r
y <- seq(0, N, 1)
prob <- dbinom(y, N, p)
prob
```

```
 [1] 0.0282475249 0.1210608210 0.2334744405 0.2668279320 0.2001209490
 [6] 0.1029193452 0.0367569090 0.0090016920 0.0014467005 0.0001377810
[11] 0.0000059049
```

```r
plot(y, prob, type="h", lwd=3)
```




```
#R による乱数の生成
ns <- 20
yobs <- rbinom(ns, N, p)
yobs
```

```
 [1] 2 1 3 3 2 1 3 5 3 4 3 3 4 4 4 4 5 3 3 2
```

```
mean(yobs)
```

```
[1] 3.1
```

```
var(yobs)
```

```
[1] 1.252632
```

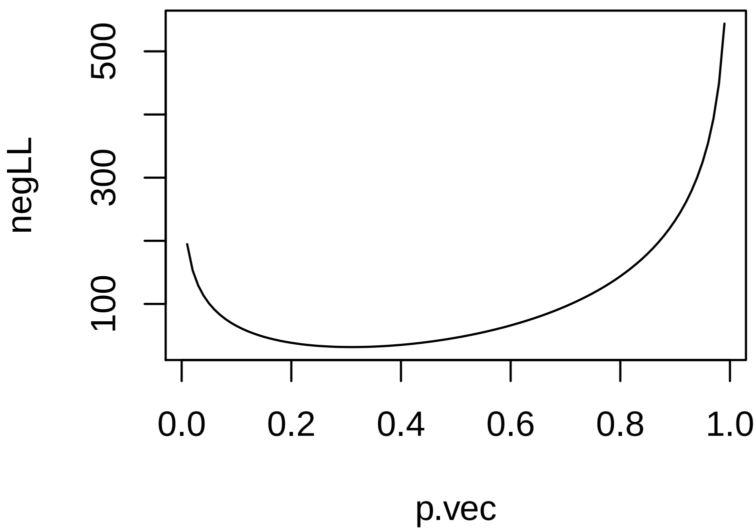
```
sd(yobs)
```

```
[1] 1.11921
```

```
median(yobs)
```

```
[1] 3
```

```r
#p に対する (負の) 対数尤度関数の定義とグラフ
negloglike.bin <- function(p){
 prob <- dbinom(yobs, N, p, log=T)
 loglike <- sum(prob)
 negloglike <- -loglike
 negloglike
}
p.vec <- seq(0.01, 0.99, 0.01)
negLL <- sapply(p.vec, negloglike.bin)
plot(p.vec, negLL, type="l")
```




```r
# p に対する (負の) 対数尤度関数の最小化
res.bin <- optim(0.5, negloglike.bin, method="L-BFGS-B", lower=0.001, upper=0.999)
res.bin
```

```
$par
[1] 0.3100006

$value
[1] 31.70065
```

```
$counts
function gradient
       6        6

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```r
p.est <- res.bin$par
p.est
```

```
[1] 0.3100006
```

```r
mean(yobs)/N
```

```
[1] 0.31
```

```r
# （補足） p に対する（負の）対数尤度関数の最小化
negloglike.bin <- function(par){
 p <- 1/(1+exp(-par))
 prob <- dbinom(yobs, N, p, log=T)
 loglike <- sum(prob)
 negloglike <- -loglike
 negloglike
}
res.bin <- optim(0, negloglike.bin, method="BFGS")
p.est <- 1/(1+exp(-res.bin$par))
p.est
```

```
[1] 0.31
```