

Science under uncertainty and science of uncertainty

FPA2020 Lecture 3

Toshihide Kitakado

May 20, 2020

Contents

| | | |
|----------|--|-----------|
| 1 | Types of uncertainty in fishery population analysis and management | 1 |
| 2 | Expressing uncertainty of data (1) | 2 |
| 2.1 | Binomial Distribution | 2 |
| 2.2 | Poisson Distribution | 3 |
| 2.3 | Hyper-geometric Distribution | 4 |
| 2.4 | Negative Binomial Distribution | 6 |
| 3 | Expressing uncertainty of data (2) | 7 |
| 3.1 | Normal distribution | 7 |
| 3.2 | Normal mixture distribution | 8 |
| 3.3 | Diffusion and bivariate normal | 10 |
| 4 | Uncertainty in estimation and model selection | 14 |
| 4.1 | Example:Stock-recruitment function | 14 |
| 4.2 | Estimation | 17 |
| 4.3 | Single simulation to see performance of estimation and model selection | 19 |
| 4.4 | Repeat this simulation | 22 |

1 Types of uncertainty in fishery population analysis and management

There are several kinds of uncertainty:

- Uncertainty in data (e.g. randomness, observation bias, measurement errors, mis-recording, ...)
- Uncertainty in input parameters (e.g. unknown natural mortality, ...)
- Process uncertainty (e.g. stochastic variation from deterministic model, environmental variability, process error...)
- Estimation uncertainty (e.g. estimation error, Standard error, Coefficient of variation, bias...)
- Model uncertainty (“All models are wrong, but some are useful” by George Box)
- Implementation uncertainty (especially in “conservation and management”)

Among the different types of uncertainty above, let us focus on “uncertainty in data”, “estimation uncertainty” and “model uncertainty”. The rest will be covered subsequent lectures.

2 Expressing uncertainty of data (1)

I shall list a few discrete distributions here only illustrative purposes. (I removed some distributions from my initial handout to reduce the volume of handout...)

2.1 Binomial Distribution

2.1.1 Definition

- Notation: $Y \sim \text{Bin}(n, p)$
- Probability function (PF): $P(Y = y) = \binom{n}{y} p^y (1-p)^{n-y}$, $y = 0, 1, \dots, n$ ($0 \leq p \leq 1$)
- Expectation:

$$\begin{aligned} E[Y] &= \sum_{y=0}^n y \binom{n}{y} p^y (1-p)^{n-y} = np \sum_{y=1}^n \frac{(n-1)!}{(y-1)!(n-y)!} p^{y-1} (1-p)^{n-y} \\ &= np \sum_{y=0}^{n-1} \frac{(n-1)!}{y!(n-1-y)!} p^y (1-p)^{n-1-y} = np. \end{aligned}$$

- Variance: Similarly, $E[Y(Y-1)] = n(n-1)p^2$, and therefore

$$V[Y] = E[Y^2] - E[Y]^2 = E[Y(Y-1)] + E[Y] - E[Y]^2 = n(n-1)p^2 + np - (np)^2 = np(1-p).$$

2.1.2 Properties

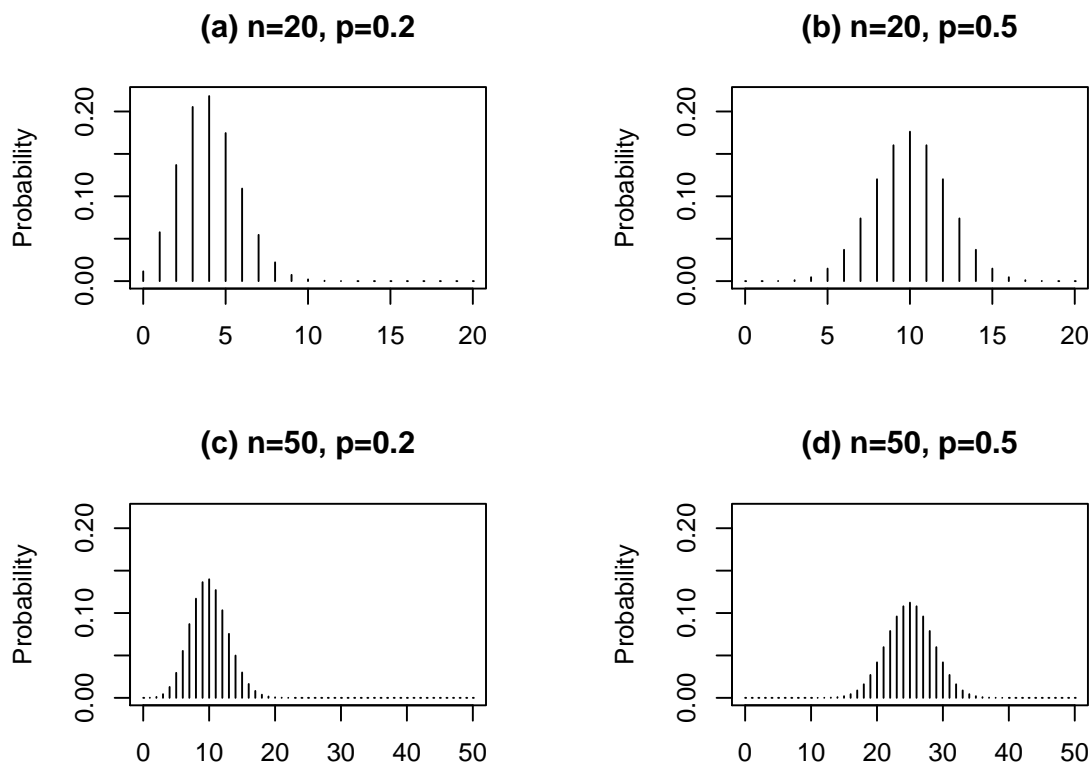
1. Reproductivity ()

Ass. Y_i ($i = 1, 2, \dots, m$) are independently distributed to $\text{Bin}(n_i, p)$. Then

$$T = \sum_{i=1}^m Y_i \sim \text{Bin}(N, p), \text{ where } N = \sum_{i=1}^m n_i.$$

2.1.3 Figures of PF

```
par(mfrow = c(2, 2), mar = c(3, 4, 4, 4))
yvec <- seq(0, 20, 1)
ylim=c(0, 0.22); PP <- "Probability"
plot(yvec, dbinom(yvec, 20, 0.2), "h", ylim=ylim, ylab=PP, main = "(a) n=20, p=0.2")
plot(yvec, dbinom(yvec, 20, 0.5), "h", ylim=ylim, ylab=PP, main = "(b) n=20, p=0.5")
yvec <- seq(0, 50, 1);
plot(yvec, dbinom(yvec, 50, 0.2), "h", ylim=ylim, ylab=PP, main = "(c) n=50, p=0.2")
plot(yvec, dbinom(yvec, 50, 0.5), "h", ylim=ylim, ylab=PP, main = "(d) n=50, p=0.5")
```



2.2 Poisson Distribution

2.2.1 Definition

- Notation: $Y \sim Pois(\lambda)$
- Probability function (PF): $P(Y = y) = e^{-\lambda} \frac{\lambda^y}{y!}$, $y = 0, 1, 2, \dots$ ($\lambda > 0$)
- Expectation and Variance: $E[Y] = V[Y] = \lambda$, which is easily derived using Maclaurin's formula

$$e^\lambda = 1 + \frac{\lambda}{1!} + \frac{\lambda^2}{2!} + \dots + \frac{\lambda^n}{n!} + \dots = \sum_{y=0}^{\infty} \frac{\lambda^y}{y!}$$

2.2.2 Properties

1. Reproductivity ()

Ass. Y_i ($i = 1, 2, \dots, m$) are independently distributed to $Pois(\lambda_i)$. Then

$$T = \sum_{i=1}^m Y_i \sim Pois\left(\sum_{i=1}^m \lambda_i\right).$$

2. Conditional probability

Ass. Y_1 and Y_2 are independently distributed to $Pois(\lambda_1)$ and $Pois(\lambda_2)$, respectively. Then, the conditional distribution of Y_1 given $T = Y_1 + Y_2 = t$ is $Bin(t, \theta)$, where $\theta = \frac{\lambda_1}{\lambda_1 + \lambda_2}$???

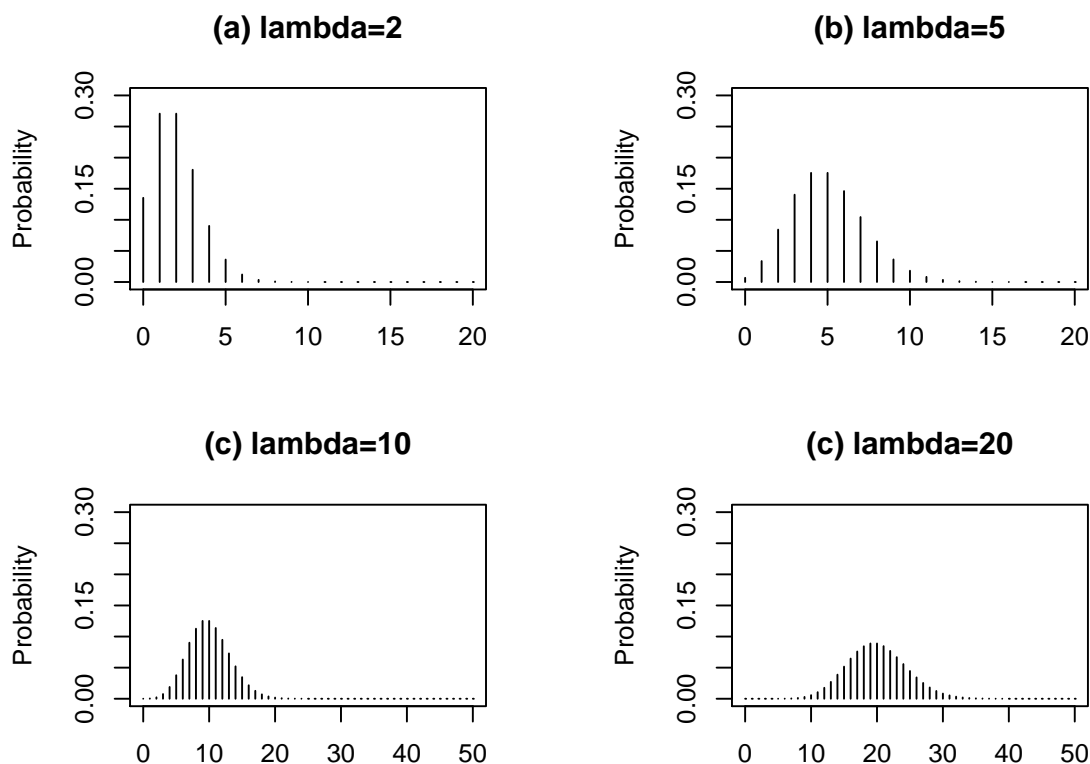
3. Ass. $Y \sim B(n, p)$. If $n \rightarrow \infty$ and $p \rightarrow 0$ but $\lambda = np$, then $Y \approx Pois(\lambda)$???

2.2.3 Figures of PF

```

par(mfrow = c(2, 2), mar = c(3, 4, 4, 4))
yvec <- seq(0, 20, 1); ylim=c(0, 0.3); PP <- "Probability"
plot(yvec, dpois(yvec, 2), "h", ylim = c(0, 0.3), ylab=PP, main = "(a) lambda=2")
plot(yvec, dpois(yvec, 5), "h", ylim = c(0, 0.3), ylab=PP, main = "(b) lambda=5")
yvec <- seq(0, 50, 1); ylim=c(0, 0.3);
plot(yvec, dpois(yvec, 10), "h", ylim = c(0, 0.3), ylab=PP, main = "(c) lambda=10")
plot(yvec, dpois(yvec, 20), "h", ylim = c(0, 0.3), ylab=PP, main = "(c) lambda=20")

```



2.3 Hyper-geometric Distribution

2.3.1 Definition

- Description: (e.g.) N =#white and black balls, M =#white balls, n =sample size, Y =rv of white balls.
- Notation: $Y \sim HG(N, M, n)$
- Probability function (PF): $P(Y = y) = \frac{\binom{M}{y} \binom{N-M}{n-y}}{\binom{N}{n}}$, $\max(0, M + n - N) \leq y \leq \min(M, n)$
- Expectation and Variance: $E[Y] = n \frac{M}{N}$, $V[Y] = n \frac{N-n}{N-1} \frac{M}{N} \left(1 - \frac{M}{N}\right)$

2.3.2 Properties

1. Ass. $p = M/N$ and N, M are large enough relative to n , then $Y \approx \text{Bin}(n, p)$
2. Ass, $X \text{ Bin}(M, p)$ and $Y \text{ Bin}(N-M, p)$, then the conditional distribution of X given $T = X+Y = t$ is $HG(N, M, t)$.

2.3.3 Figures of PF

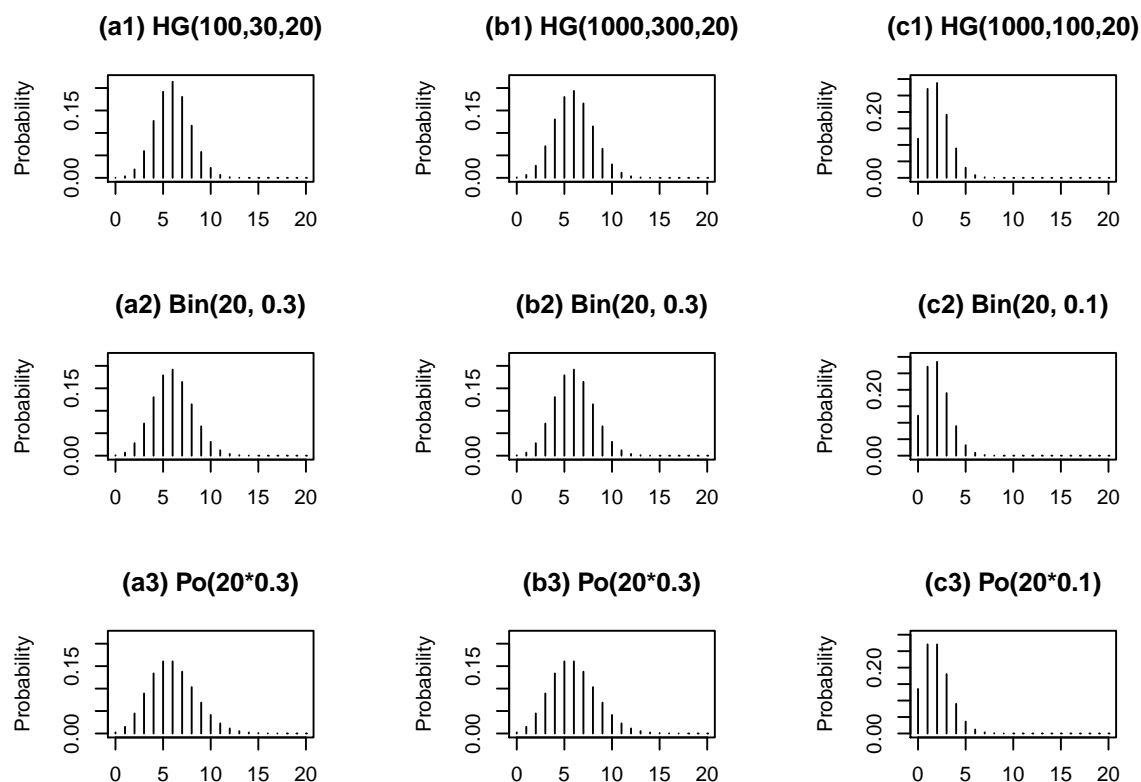
```

par(mfcol = c(3, 3), mar = c(3, 4, 4, 4))
yvec <- seq(0, 20, 1)
ylim=c(0, 0.22); PP <- "Probability"
plot(yvec, dhyper(yvec, 30, 70, 20), "h", ylim=ylim, ylab=PP,
     main = "(a1) HG(100,30,20)")
plot(yvec, dbinom(yvec, 20, 0.3), "h", ylim=ylim, ylab=PP,
     main = "(a2) Bin(20, 0.3)")
plot(yvec, dpois(yvec, 20*0.3), "h", ylim=ylim, ylab=PP,
     main = "(a3) Po(20*0.3)")

ylim=c(0, 0.22); PP <- "Probability"
plot(yvec, dhyper(yvec, 300, 700, 20), "h", ylim=ylim, ylab=PP,
     main = "(b1) HG(1000,300,20)")
plot(yvec, dbinom(yvec, 20, 0.3), "h", ylim=ylim, ylab=PP,
     main = "(b2) Bin(20, 0.3)")
plot(yvec, dpois(yvec, 20*0.3), "h", ylim=ylim, ylab=PP,
     main = "(b3) Po(20*0.3)")

ylim=c(0, 0.3); PP <- "Probability"
plot(yvec, dhyper(yvec, 100, 900, 20), "h", ylim=ylim, ylab=PP,
     main = "(c1) HG(1000,100,20)")
plot(yvec, dbinom(yvec, 20, 0.1), "h", ylim=ylim, ylab=PP,
     main = "(c2) Bin(20, 0.1)")
plot(yvec, dpois(yvec, 20*0.1), "h", ylim=ylim, ylab=PP,
     main = "(c3) Po(20*0.1)")

```



2.4 Negative Binomial Distribution

2.4.1 Definition

- Description: The “random” number of failure trials until a “predefined” number k of success has occurred. p is a success prob.
- Notation: $Y \sim NB(k, p)$
- Probability function (PF): $P(Y = y) = \binom{k+y-1}{y} p^k (1-p)^y$, $y = 0, 1, 2, \dots$ ($0 \leq p \leq 1$)
- Expectation and Variance: $E[Y] = k \frac{1-p}{p}$, $V[Y] = k \frac{1-p}{p^2}$

2.4.2 Properties

1. Reproductivity ()

Ass. Y_i ($i = 1, 2, \dots, m$) are independently distributed to $NB(p, k_i)$, then $T = \sum_{i=1}^m Y_i \sim$

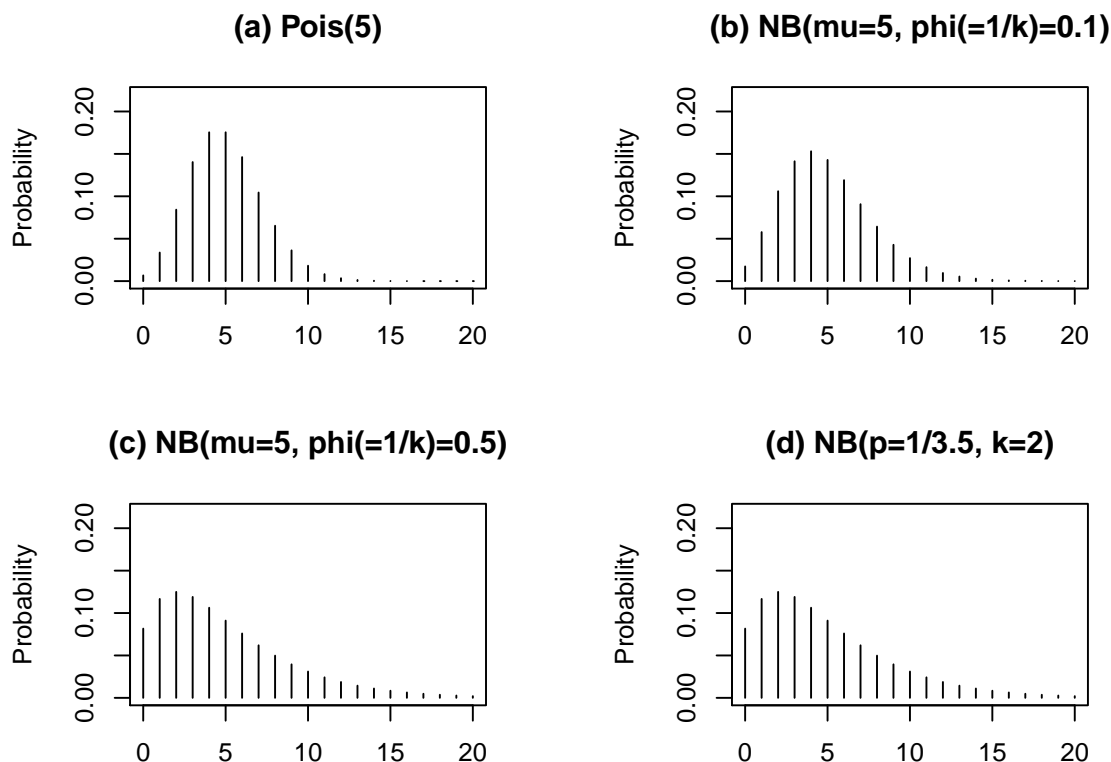
$$NB(p, \sum_{i=1}^m k_i)???$$

2. Geometric distribution $Ge(p) = NB(1, p)$: $P(Y = y) = p(1-p)^y$, $y = 0, 1, 2, \dots$ ($0 \leq p \leq 1$)

2.4.3 Figures of PF

```
#Usage
# dnbinom(x, size, prob, mu, log = FALSE)
#x: vector of (non-negative integer) quantiles
#size: target for number of "successful trials", or "dispersion parameter"
#prob: probability of success in each trial. 0 < prob <= 1
#mu: alternative parametrization via mean

par(mfrow = c(2, 2), mar = c(3, 4, 4, 4))
yvec <- seq(0, 20, 1)
ylim=c(0, 0.22); PP <- "Probability"
plot(yvec, dpois(yvec, 5), "h", ylim=ylim, ylab=PP,
     main = "(a) Pois(5)")
plot(yvec, dnbinom(yvec, mu=5, size=1/0.1), "h", ylim=ylim, ylab=PP,
     main = "(b) NB(mu=5, phi(=1/k)=0.1)")
plot(yvec, dnbinom(yvec, mu=5, size=1/0.5), "h", ylim=ylim, ylab=PP,
     main = "(c) NB(mu=5, phi(=1/k)=0.5)")
plot(yvec, dnbinom(yvec, prob=1/3.5, size=2), "h", ylim=ylim, ylab=PP,
     main = "(d) NB(p=1/3.5, k=2)")
```



3 Expressing uncertainty of data (2)

Another type of distribution family is the continuous distribution such as the normal, gamma, chi-square, t-, F- distribution. I here briefly mention the normal distribution and its mixture, and then I also mention the bivariate normal distribution.

3.1 Normal distribution

3.1.1 Definition

$$Y \sim N(\mu, \sigma^2)$$

* μ : mean

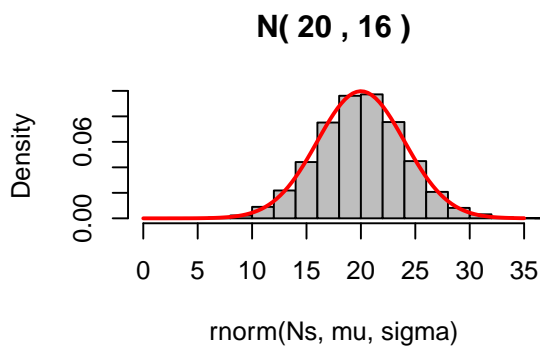
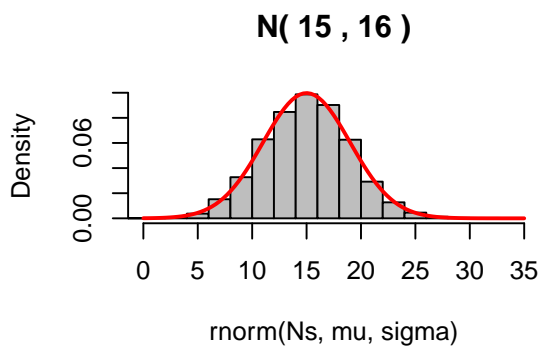
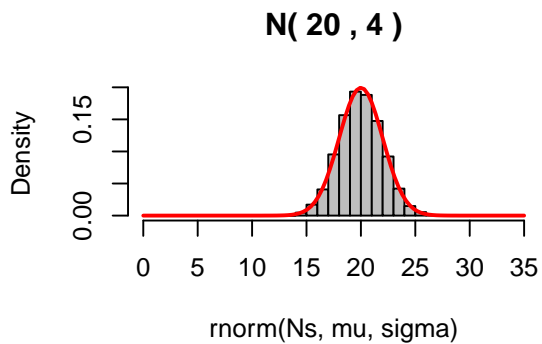
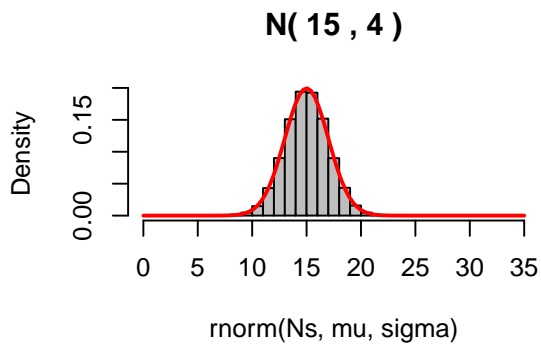
* σ^2 : variance (σ : standard deviation)

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \quad -\infty < y < \infty$$

```
NormalPDF <- function(mu, sigma, Ns, From, To){
  hist(rnorm(Ns,mu,sigma), freq=F, col="gray", xlim=c(From,To),
       main=paste("N(",mu,",",sigma^2,")") )
  curve(dnorm(x,mu,sigma), col="red", lwd=2, add=T)
}
par(mfrow=c(2,2))
NormalPDF(15,2,10000,0,35)
NormalPDF(20,2,10000,0,35)
```

```
NormalPDF(15,4,10000,0,35)
```

```
NormalPDF(20,4,10000,0,35)
```



3.2 Normal mixture distribution

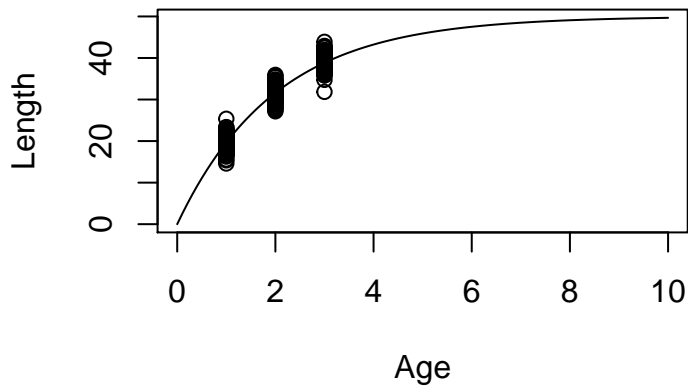
Assume that fish grows as age increases.

```
VB <- function(tt, Linf, KK, t0){ Linf*(1-exp(-KK*(tt-t0))) }
Linf <- 50
KK <- 0.5
t0 <- 0
VB(0:10, Linf, KK, t0)
```

```
[1] 0.00000 19.67347 31.60603 38.84349 43.23324 45.89575 47.51065 48.49013
[9] 49.08422 49.44455 49.66310
```

```
Agevec <- seq(0,10,0.1)
Lengthvec <- VB(Agevec, Linf, KK, t0)
plot(Agevec, Lengthvec, type="l", xlab="Age", ylab="Length")

Ns <- 100
points(rep(1,Ns),rnorm(Ns, VB(1, Linf, KK, t0), sd=2))
points(rep(2,Ns),rnorm(Ns, VB(2, Linf, KK, t0), sd=2))
points(rep(3,Ns),rnorm(Ns, VB(3, Linf, KK, t0), sd=2))
```

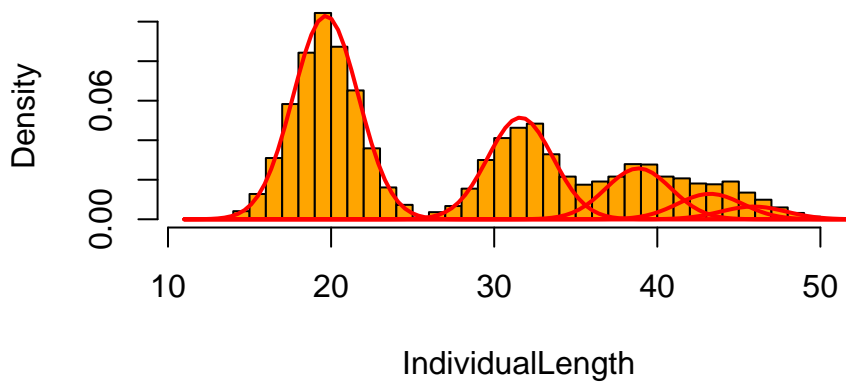
```
S <- 0.5
Age <- 1:5
Omega <- S^Age
Prop <- Omega/sum(Omega); Prop
```

```
[1] 0.51612903 0.25806452 0.12903226 0.06451613 0.03225806
```

```
Ns <- 10000
IndividualAge <- sample(Age, Ns, prob=Prop, replace=T)
MeanLength <- VB(Age, Linf, KK, t0)
IndividualLength <- rnorm(Ns, MeanLength[IndividualAge], sd=2)
hist(IndividualLength, freq=F, breaks=50, col="orange", main="Normal Mixture")

for(age in Age){
  curve(Prop[age]*dnorm(x, MeanLength[age], sd=2), col="red", lwd=2, add=T)
}
```

Normal Mixture



3.3 Diffusion and bivariate normal

3.3.1 Bivariate normal distribution

$$f(x; \mu, \Sigma) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}.$$

3.3.2 Diffusion process

Let (x, y) be a spatial position of animal released since a time t passes. Then the distribution of the position is given by a diffusion equation as follows:

$$\frac{\partial f(x, y; t)}{\partial t} = D \left[\frac{\partial^2}{\partial x^2} f(x, y; t) + \frac{\partial^2}{\partial y^2} f(x, y; t) \right].$$

A solution of the equation above is derived analytically as

$$f(x, y; t) = \frac{1}{(\sqrt{2\pi Dt})^2} e^{-\frac{x^2 + y^2}{2Dt}}.$$

3.3.3 First step: a function for drawing a diffusion process

```
library(MASS)

Diffusion<- function(DD, Nstep, Fig=F){
  #DD is a diffusion variance
  Range <- c(-3,3)*sqrt(DD*Nstep)
  Pos <- array(0, c(Nstep+1,2))
  for(i in 2:(Nstep+1)){
    Pos[i,] <- mvrnorm(1, mu=Pos[i-1,], Sigma=diag(DD,2))
  }
  if(Fig==T) plot(Pos, type="l", xlim=Range, ylim=Range, xlab="x", ylab="y")
  Out <- NULL
  Out$Pos <- Pos
  Out$r <- sqrt(sum(tail(Pos,1)^2))
  return(Out)
}
```

3.3.4 Do it

```
DD <- 1
Nstep <- 10
par(mfrow=c(2,2), mar=c(4,4,2,2))
Diffusion(DD,Nstep,Fig=T)
```

```
$Pos
      [,1]      [,2]
[1,] 0.0000000 0.0000000
[2,] -0.1699306 -0.6152037
[3,] -0.2459491 -1.9169196
[4,] -1.3694821 -0.1098484
[5,] -1.9086790 -0.9969192
[6,] -1.8976323 -0.7345934
[7,] -1.7769786  0.5123889
```

```
[8,] -1.52026649 -0.1917754
[9,] -1.96373365  0.6716279
[10,] -1.30849474 -0.4132926
[11,]  0.01973304  1.4963021
```

\$r

```
[1] 1.496432
```

Diffusion(DD,Nstep,Fig=T)

\$Pos

```
          [,1]      [,2]
[1,]  0.0000000  0.0000000
[2,] -0.02807901  0.5562071
[3,] -0.45699169  0.8564568
[4,]  0.52071038  0.3291042
[5,]  0.81937761  0.4551119
[6,]  1.56130367  0.6831570
[7,]  2.61273560  0.7894307
[8,]  4.23205134 -0.2596272
[9,]  4.01675312 -1.2576690
[10,]  3.90745156  0.2032157
[11,]  2.34797647 -1.7732424
```

\$r

```
[1] 2.942343
```

Diffusion(DD,Nstep,Fig=T)

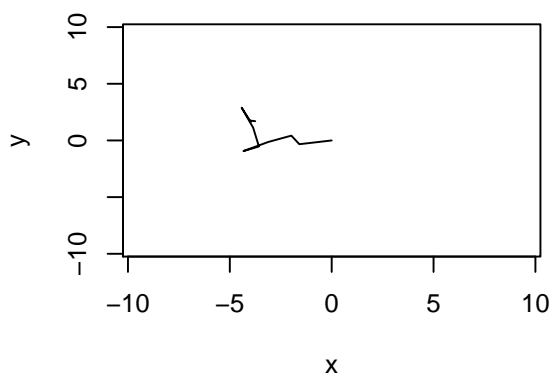
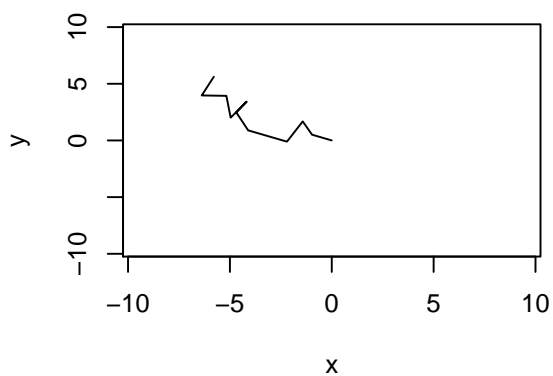
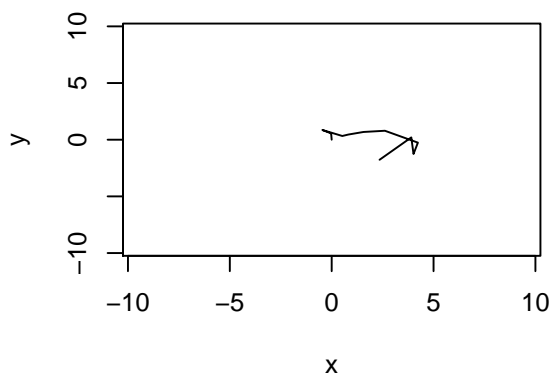
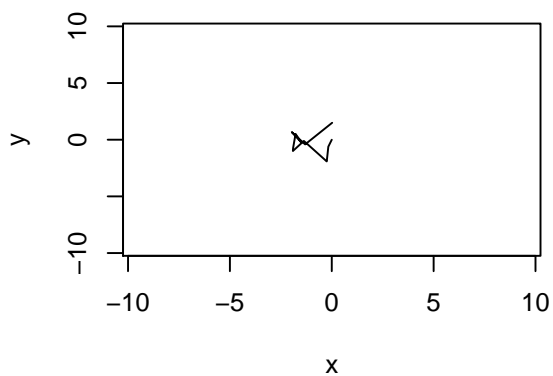
\$Pos

```
          [,1]      [,2]
[1,]  0.000000  0.000000
[2,] -0.969311  0.5078714
[3,] -1.426340  1.6798390
[4,] -2.199276 -0.1070071
[5,] -4.103724  0.8887306
[6,] -4.685524  2.4722853
[7,] -4.175380  3.4255131
[8,] -4.963794  2.0101903
[9,] -5.172858  3.9343752
[10,] -6.383020  3.9715448
[11,] -5.790257  5.6160284
```

\$r

```
[1] 8.066403
```

Diffusion(DD,Nstep,Fig=T)



```
$Pos
```

```
      [,1]      [,2]
[1,] 0.000000 0.000000
[2,] -1.588720 -0.3329174
[3,] -1.980304 0.4248365
[4,] -3.113137 -0.1346292
[5,] -4.332158 -0.9366855
[6,] -3.572728 -0.5308945
[7,] -3.856175 1.1296583
[8,] -4.420077 2.8898249
[9,] -3.972953 1.6406422
[10,] -4.037802 1.7557483
[11,] -3.752581 1.6993008
```

```
$r
```

```
[1] 4.119404
```

3.3.5 Repeat it and draw final positions

We repeat this simulation 10000 times and plot the final positions of animals after 10 steps. You can see a beautiful plot of points distributed over x-y plain in a concentric fashion.

```
Nsim <- 10000
FinalPos <- array(0, c(Nsim,2))
Distance <- numeric(Nsim)
for(it in 1:Nsim){
  Res <- Diffusion(DD,Nstep,Fig=F)
  FinalPos[it,] <- tail(Res$Pos,1)
```

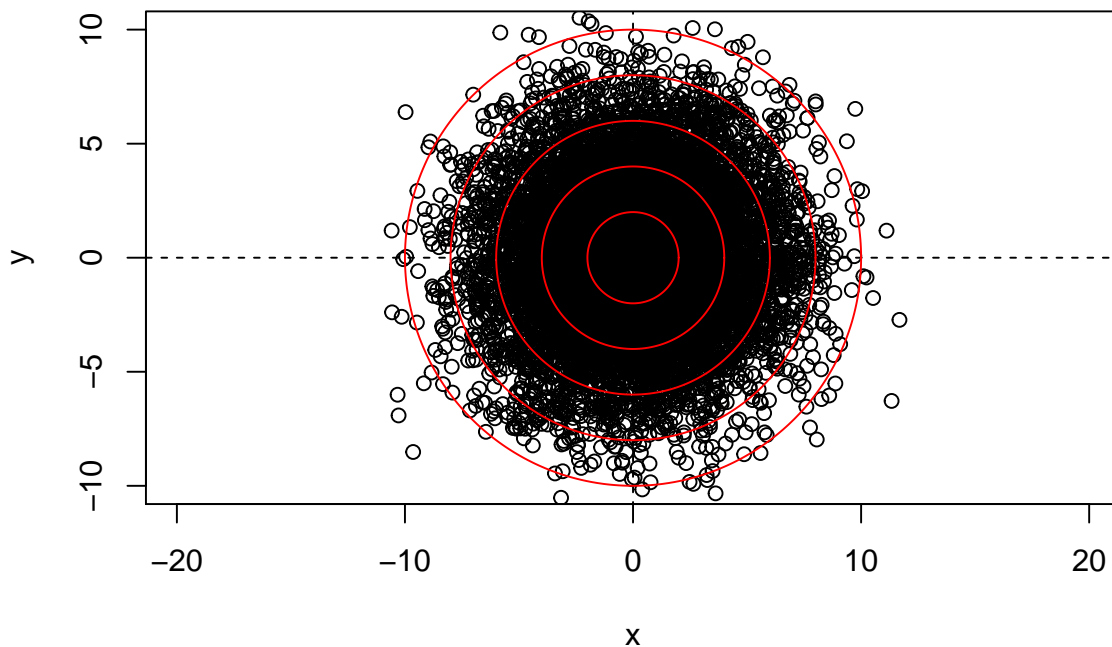
```

Distance[it] <- Res$r
}

par(mfrow=c(1,1))
plot(FinalPos[,1],FinalPos[,2], xlim=c(-1,1)*DD*Nstep, ylim=c(-1,1)*DD*Nstep,
      xlab="x", ylab="y", asp = 1)

library(plotrix)
abline(0,0, lty=2)
lines(c(0,0), c(-999,999), lty=2)
for(rr in 1:5) draw.circle(0,0,2*rr,nv=1000,border="red",col=NULL)

```



3.3.6 Chi-square and Gamma distributions

Definition

$$\begin{aligned}
 X \sim N(\mu, \sigma^2) \quad f(x; \mu, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\
 X \sim \chi_n^2 \quad f(x; n) &= \frac{1}{\Gamma(n/2)2^{n/2}} x^{n/2-1} e^{-\frac{x}{2}} \\
 X \sim Ga(\alpha, \beta) \quad f(x; \alpha, \beta) &= \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}
 \end{aligned}$$

Properties

$$\begin{aligned} X_1, X_2, \dots, X_n (iid) &\sim N(\mu, \sigma^2) \\ Y = \sum_{i=1}^n X_i^2 &\sim \chi_n^2 \\ Y &\sim Ga(n/2, 2) \end{aligned}$$

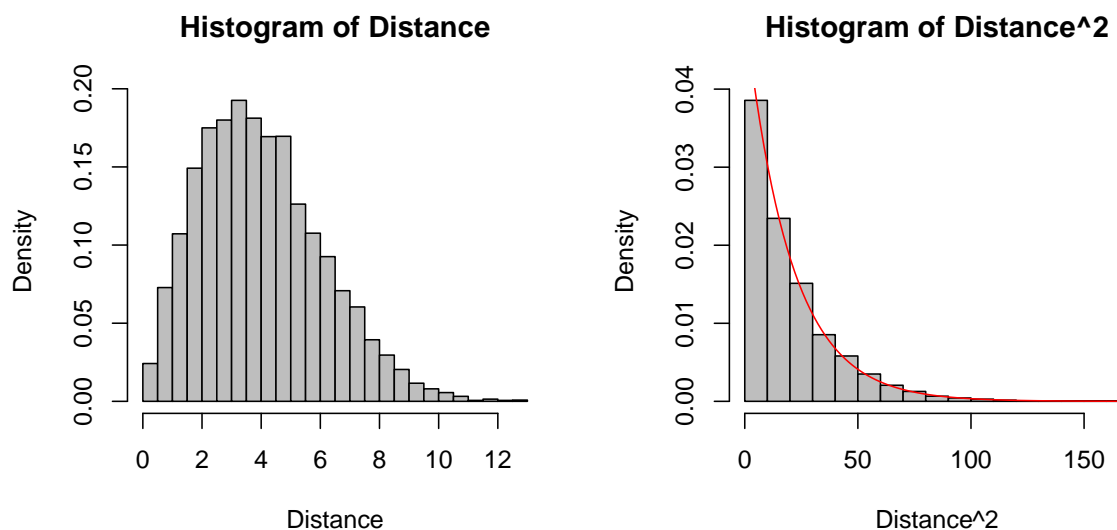
So what?

$$\begin{aligned} X(t) &\sim N(0, Dt), \quad Y(t) \sim N(0, Dt) \\ \tilde{X}(t) = \frac{X(t)}{\sqrt{Dt}} &\sim \chi_1^2, \quad \tilde{Y}(t) = \frac{Y(t)}{\sqrt{Dt}} \sim \chi_1^2 \\ \Rightarrow \tilde{X}^2(t) + \tilde{Y}^2(t) &\sim \chi_2^2 \quad (\text{because of independency}) \\ \Rightarrow X^2(t) + Y^2(t) &\sim (Dt)\chi_2^2 \end{aligned}$$

3.3.7 Draw a histogram of “Distance”

You can find that the square of distance of animal movement is distributed to a scaled chi-square distribution!

```
par(mfrow=c(1,2))
hist(Distance, freq=F, breaks=20, col="gray")
hist(Distance^2, freq=F, breaks=20, col="gray")
curve(dgamma(x, shape=2*0.5, scale=2*(DD*Nstep)), col="red", add=T)
```



4 Uncertainty in estimation and model selection

4.1 Example: Stock-recruitment function

4.1.1 Reading data

```
Data <- read.csv(file="PinkSalmonAK.csv", header=T)
head(Data,3)
```

| | Year | Harvest | Escapement | Return | SST |
|---|------|---------|------------|--------|-------|
| 1 | 1958 | NA | NA | 2678 | NA |
| 2 | 1959 | NA | NA | 10459 | 11.23 |
| 3 | 1960 | 1260 | 1418 | 2446 | 11.63 |

```
tail(Data,3)
```

```

      Year Harvest Escapement Return  SST
32 1989   13638         4577 23359 12.3
33 1990    5659         3802    NA   NA
34 1991   18112         5247    NA   NA

```

```
UseList <- 3:32
```

```
Year <- Data$Year[UseList]
```

```
SP <- Data$Escapement[UseList]
```

```
Rec <- Data$Return[UseList]
```

```
SST <- Data$SST[UseList]
```

```
RPS <- Rec/SP
```

4.1.2 Visual presentation (sorry for traditional plot here...)

```
par(mfrow=c(2,2), mar=c(4,4,3,4))
```

```
plot(Year, SP, type="l", col="red", ylab="Spawners")
```

```
par(new=T); plot(Year, Rec, type="l", col="blue", axes=FALSE, ylab="")
```

```
axis(4); mtext("Recruitments", side=4, line=2.5, cex=0.8)
```

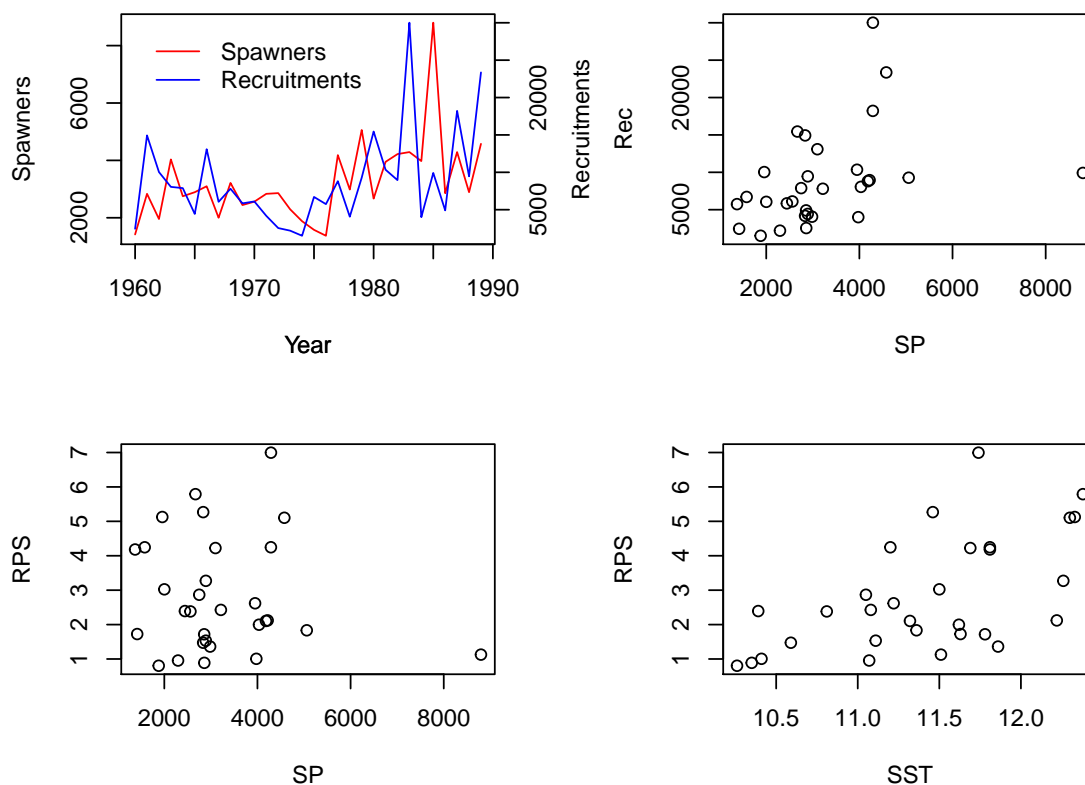
```
legend(1960, 30000, lty=1, col=c("red","blue"),
```

```
      legend=c("Spawners", "Recruitments"), bty="n")
```

```
plot(SP, Rec)
```

```
plot(SP, RPS)
```

```
plot(SST, RPS)
```



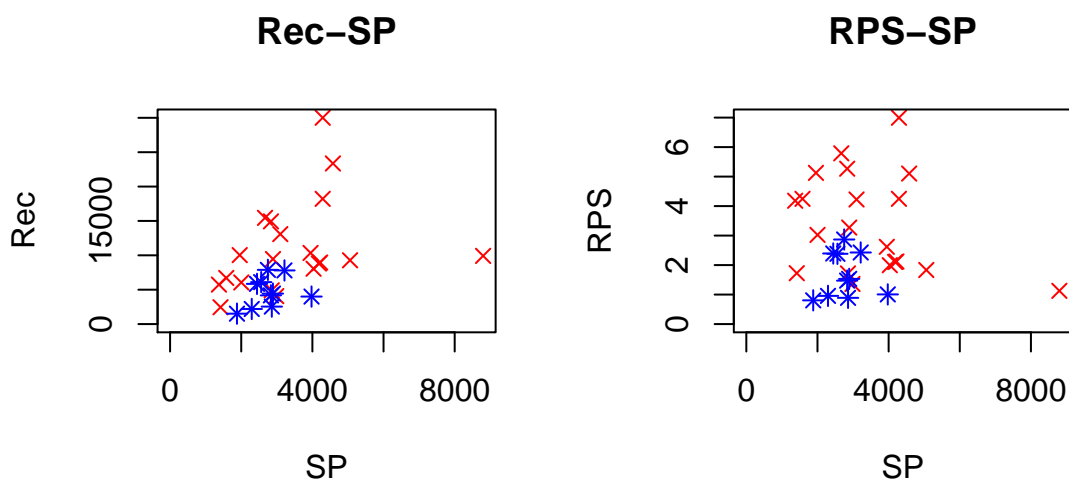
4.1.3 Another visual presentation (sorry for traditional plot here...)

```
#Data exploration 1
par(mfrow=c(1,2))
SST.high <- (SST>=11.2)
SST.high[1:10]

[1] TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE

plot(SP[SST.high], Rec[SST.high], xlim=c(0,max(SP)), ylim=c(0,max(Rec)),
     main="Rec-SP", xlab="SP", ylab="Rec", pch=4, col="red")
points(SP[SST.high==F], Rec[SST.high==F], pch=8, col="blue")

plot(SP[SST.high], RPS[SST.high], xlim=c(0,max(SP)), ylim=c(0,max(RPS)),
     main="RPS-SP", xlab="SP", ylab="RPS", pch=4, col="red")
points(SP[SST.high==F], RPS[SST.high==F], pch=8, col="blue")
```



4.1.4 Initial guess of Stock-recruitment function

```
#Def of BH function
BH <- function(SP, a, b) a*SP/(b+SP)

#Shiny presentation
library(shiny)

#User Interface (ui)
ui <- shinyUI(fluidPage(
  titlePanel("S-R with B-H"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("a", "a:", min = 0, max = 50000, value = 10000),
      sliderInput("b", "b:", min = 0, max = 50000, value = 10000)),
    mainPanel( plotOutput("Plot") )
  )
))
```



```

#Server
server <- shinyServer(function(input, output) {
  output$Plot <- renderPlot({
    plot(SP, Rec, xlim=c(0,max(SP)), ylim=c(0,max(Rec)))
    curve(BH(x,input$a,input$b), add=T, col="red")
  })
})

#Run
#runApp(list(ui=ui, server=server))

```

4.2 Estimation

4.2.1 Estimation of Stock-recruitment function (Model 1: without SST)

```

#
Res1 <- nls(log(Rec)~log(BH(SP,a,b)), start=list(a=10000,b=10000))
summary(Res1)

```

Formula: $\log(\text{Rec}) \sim \log(\text{BH}(\text{SP}, a, b))$

Parameters:

| | Estimate | Std. Error | t value | Pr(> t) |
|---|----------|------------|---------|----------|
| a | 47827 | 84438 | 0.566 | 0.576 |
| b | 16875 | 35260 | 0.479 | 0.636 |

Residual standard error: 0.6116 on 28 degrees of freedom

Number of iterations to convergence: 5

Achieved convergence tolerance: 1.833e-06

```
Est1 <- coef(Res1); Est1
```

```

      a      b
47826.98 16875.36

```

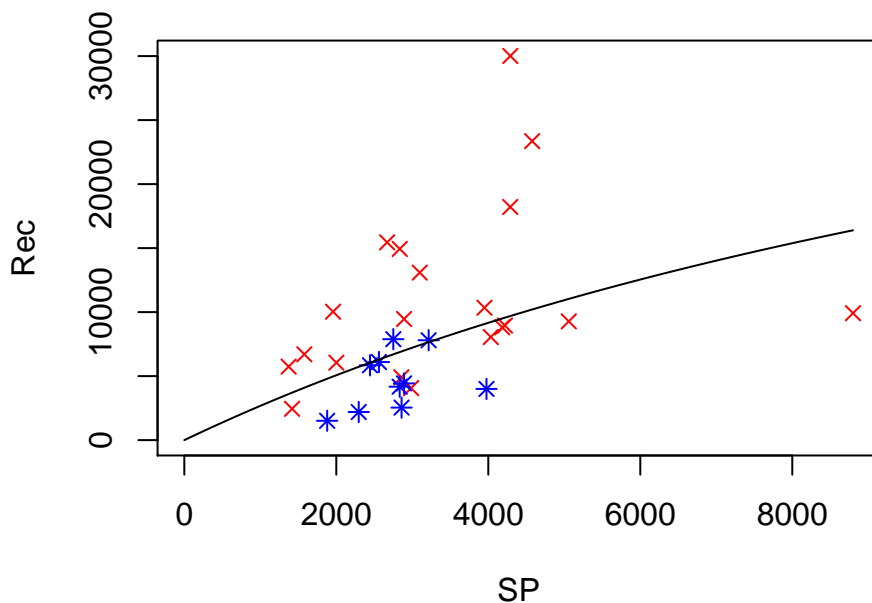
```
AIC1 <- AIC(Res1); AIC1
```

```
[1] 59.56389
```

```

par(mfrow=c(1,1))
plot(SP[SST.high], Rec[SST.high], xlim=c(0,max(SP)), ylim=c(0,max(Rec)),
     xlab="SP", ylab="Rec", pch=4, col="red")
points(SP[SST.high==F], Rec[SST.high==F], pch=8, col="blue")
curve(BH(x,Est1[1],Est1[2]), add=T)

```



4.2.2 Estimation of Stock-recruitment function (Model 2: with SST)

```
Res2 <- nls(log(Rec)~log(BH(SP, aH*(SST>=11.2)+aL*(SST<11.2), b)),
            start=list(aH=10000, aL=10000, b=10000))
summary(Res2)
```

Formula: $\log(\text{Rec}) \sim \log(\text{BH}(\text{SP}, aH * (\text{SST} \geq 11.2) + aL * (\text{SST} < 11.2), b))$

Parameters:

| | Estimate | Std. Error | t value | Pr(> t) |
|----|----------|------------|---------|----------|
| aH | 30520 | 23899 | 1.277 | 0.212 |
| aL | 14498 | 12030 | 1.205 | 0.239 |
| b | 6847 | 7807 | 0.877 | 0.388 |

Residual standard error: 0.5058 on 27 degrees of freedom

Number of iterations to convergence: 5

Achieved convergence tolerance: 6.486e-06

```
Est2 <- coef(Res2); Est2
```

| | aH | aL | b |
|--|-----------|-----------|----------|
| | 30520.435 | 14498.070 | 6847.043 |

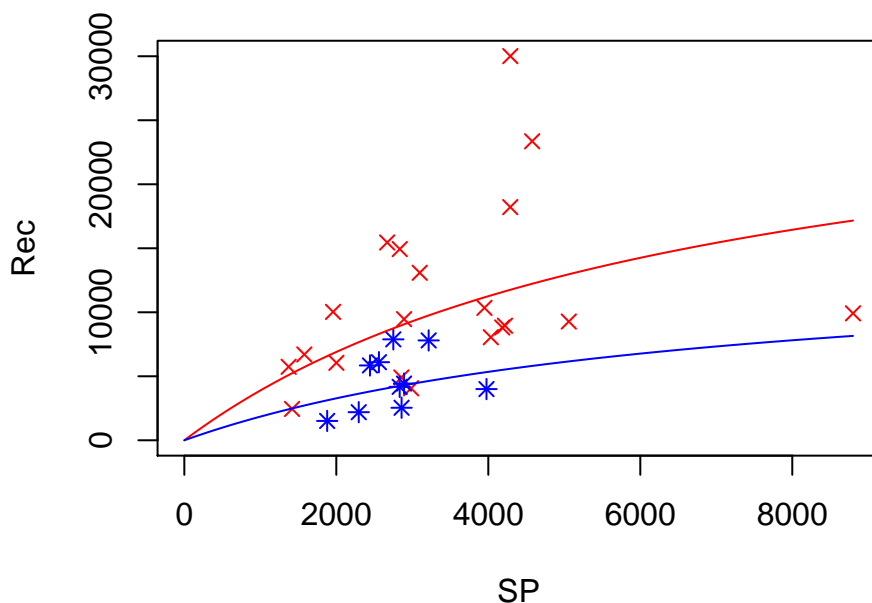
```
AIC2 <- AIC(Res2); AIC2
```

```
[1] 49.08097
```

```

plot(SP[SST.high], Rec[SST.high], xlim=c(0,max(SP)), ylim=c(0,max(Rec)),
     xlab="SP", ylab="Rec", pch=4, col="red")
points(SP[SST.high==F], Rec[SST.high==F], pch=8, col="blue")
curve(BH(x,Est2[1],Est2[3]), add=T, col="red")
curve(BH(x,Est2[2],Est2[3]), add=T, col="blue")

```



4.3 Single simulation to see performance of estimation and model selection

- Assume that Model 2 is the true model, and generate data from it
- How are the estimates?
- How is the model selection performance?

4.3.1 Setting parameters

```

aHtrue <- Est2[1]*0.7
aLtrue <- Est2[2]
btrue <- Est2[3]
set.seed(123456)

CVx <- 0.2
CVy <- 0.3
TT <- 30
TThigh <- 19

```

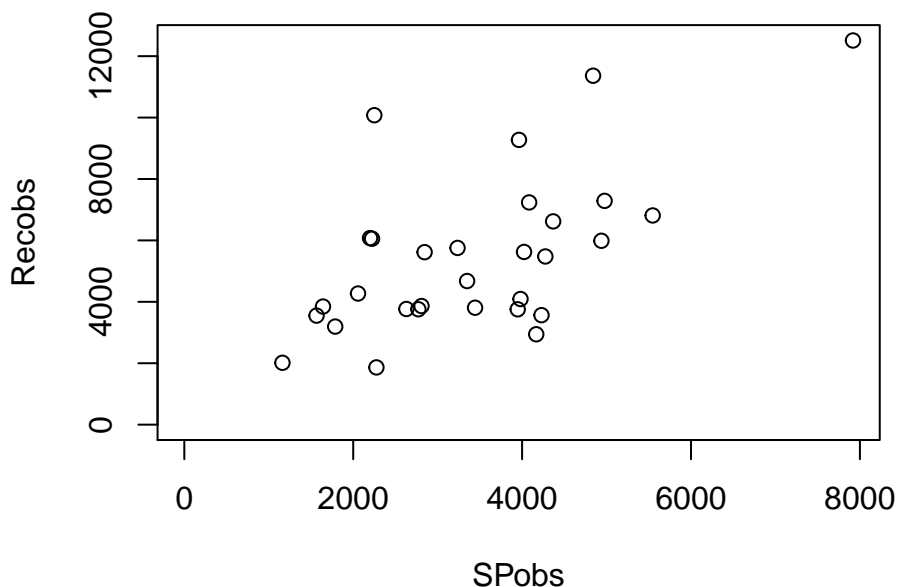
4.3.2 Generating simulation data

```

SPobs <- SP*exp(rnorm(TT,-0.5*CVx^2,CVx))
Yearhigh <- sample(1:TT, TThigh, replace=F)

```

```
Dummy <- numeric(TT)
Dummy[Yearhigh] <- 1
Recobs <- (BH(SP,aHtrue,btrue)*Dummy +
           BH(SP,aLtrue,btrue)*(1-Dummy))*exp(rnorm(TT, -0.5*CVy^2, CVy))
plot(SPobs, Recobs, xlim=c(0,max(SPobs)), ylim=c(0,max(Recobs)))
```



4.3.3 Estimation under Model 1 using simulation data

Note: I will explain ML estimation and model selection with AIC in other lectures.

```
#Estimation without SST
```

```
SimRes1 <- nls(log(Recobs)~log(BH(SPobs,a,b)), start=list(a=aHtrue,b=btrue))
SimRes1
```

```
Nonlinear regression model
```

```
model: log(Recobs) ~ log(BH(SPobs, a, b))
```

```
data: parent.frame()
```

```
  a      b
```

```
12947 4920
```

```
residual sum-of-squares: 4.074
```

```
Number of iterations to convergence: 5
```

```
Achieved convergence tolerance: 1.096e-06
```

```
SimEst1 <- coef(SimRes1)
```

```
SimAIC1 <- AIC(SimRes1)
```

4.3.4 Estimation under Model 2 using simulation data

```
#Estimation with SST
```

```
SimRes2 <- nls(log(Recobs)~log(BH(SPobs,aH*Dummy+aL*(1-Dummy),b)),
```

```
start=list(aH=aHtrue,aL=aLtrue,b=btrue))
```

```
SimRes2
```

```
Nonlinear regression model
```

```
model: log(Recobs) ~ log(BH(SPobs, aH * Dummy + aL * (1 - Dummy), b))
```

```
data: parent.frame()
```

```
 aH  aL  b
```

```
11296 7432 2877
```

```
residual sum-of-squares: 2.931
```

```
Number of iterations to convergence: 5
```

```
Achieved convergence tolerance: 5.217e-06
```

```
SimEst2 <- coef(SimRes2)
```

```
SimAIC2 <- AIC(SimRes2)
```

4.3.5 Summary (comparison of AIC)

```
plot(SPobs[Yearhigh], Recobs[Yearhigh], xlim=c(0,max(SPobs)*1.5),
      ylim=c(0,max(Recobs)*1.5), xlab="SP", ylab="Rec", pch=4, col="red")
points(SPobs[-Yearhigh], Recobs[-Yearhigh], pch=8, col="blue")
```

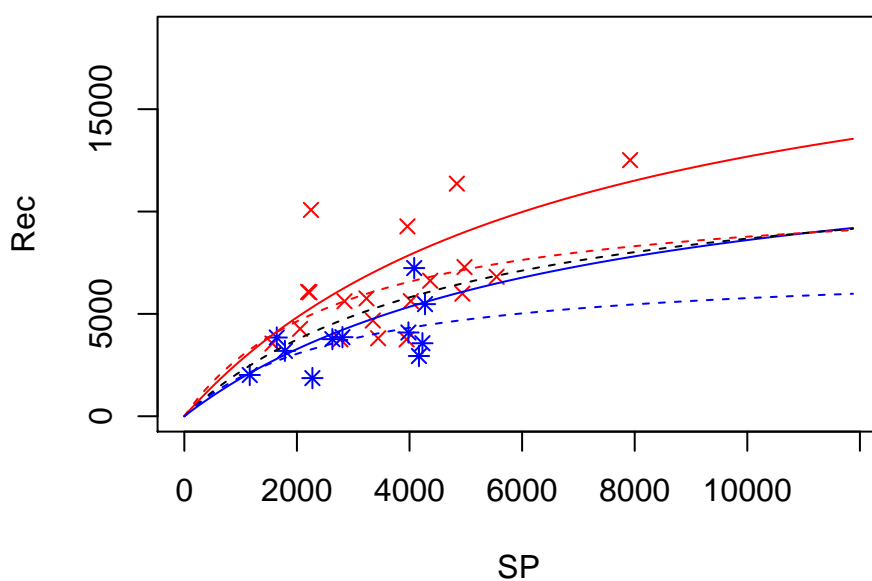
```
curve(BH(x,aHtrue,btrue), add=T, col="red")
```

```
curve(BH(x,aLtrue,btrue), add=T, col="blue")
```

```
curve(BH(x,SimEst1[1],SimEst1[2]), lty=2, add=T)
```

```
curve(BH(x,SimEst2[1],SimEst2[3]), lty=2, add=T, col="red")
```

```
curve(BH(x,SimEst2[2],SimEst2[3]), lty=2, add=T, col="blue")
```



```
data.frame(aHtrue, aLtrue, btrue)
```

```
      aHtrue  aLtrue  btrue
aH 21364.3 14498.07 6847.043
```

```
SimEst1
```

```
      a      b
12947.250 4919.677
```

```
SimEst2
```

```
      aH      aL      b
11295.685 7431.574 2876.946
```

```
data.frame(SimAIC1,SimAIC2)
```

```
      SimAIC1 SimAIC2
1 31.23628 23.36512
```

4.4 Repeat this simulation

4.4.1 A function for simulation

You can find around 5 percent chance of model misspecification in this case. So there is “model selection error” in AIC to some extent (not perfect).

```
set.seed(12345)
CVx <- 0.2
CVy <- 0.3
Nit<-100

SimAIC1 <- SimAIC2 <- numeric(Nit)

for(it in 1:Nit){

  SPobs <- SP*exp(rnorm(TT, -0.5*CVx^2, CVx))

  Yearhigh <- sample(1:TT, TThigh, replace=F)
  Dummy <- numeric(TT); Dummy[Yearhigh] <- 1
  Recobs <- (BH(SP,aHtrue,btrue)*Dummy + BH(SP,aLtrue,btrue)*(1-Dummy))*
    exp(rnorm(TT, -0.5*CVy^2, CVy))
  #plot(SPobs,Recobs)

  SimRes1 <- nls(log(Recobs)~log(BH(SPobs,exp(log_a),exp(log_b))),
                start=list(log_a=log(aLtrue),log_b=log(btrue)))
  SimAIC1[it] <- AIC(SimRes1)

  SimRes2 <- nls(log(Recobs)~
                log(BH(SPobs,exp(log_aH)*Dummy+exp(log_aL)*(1-Dummy),exp(log_b))),
                start=list(log_aH=log(aHtrue),log_aL=log(aLtrue),log_b=log(btrue)))
  SimAIC2[it] <- AIC(SimRes2)
}

Diff.AIC <- SimAIC1 - SimAIC2
```

```
mean(Diff.AIC>0)
```

```
[1] 0.94
```

```
hist(Diff.AIC, col="grey")
```

```
lines(c(0,0),c(0,100),col="red")
```

