

# R を利用した資源評価

市野川 桃子

中央水産研究所 資源管理研究センター  
資源管理グループ

(2013 年 5 月 20 日)

## 後半の概要 ～Rを使った資源評価～

- ① モデルの説明  
個体群の動態はどのようなモデルで表されるか？
- ② シミュレーション  
Rを使って余剰生産モデルの挙動を調べる
- ③ 資源評価  
漁獲データから資源状態を推定する
- ④ 結果診断  
評価モデルの確からしさを検討する
- ⑤ 年齢構造モデル  
より複雑な資源評価モデルを試す

## 配布ファイルの説明

- `ichinokawa_R.pdf`: このプレゼンファイル
- `script2.r`: このプレゼンで実施する R のコード。pdf ファイルのものと全く同じです
- `script2_fun.r`: プロダクションモデルなどの関数集
- `rvpa0.7.r`: 岡村さんと共同開発している VPA 用の関数集（近日公開？）
- `Robj` 以下: 一部、時間がかかる計算の結果の R オブジェクト

## Rコードの補足 (繰り返し計算と関数の定義)

```
# 空のベクトルの定義
x <- numeric()
# 繰り返し計算 (for)
# i を 1 から 10 まで変えて x に入れる
for(i in 1:10) x[i] <- i*2

# 関数 (function)
tmpfunc <- function(x,y=1){ # x, y は引数
  return(x+y)
}

tmpfunc(x=1,y=10)
tmpfunc(1,10)
tmpfunc(10)
```

# (1) 個体群の動態は どのようなモデルで表されるか？

# (1) 個体群の動態はどのようなモデルで表されるか？

## ● 余剰生産モデル

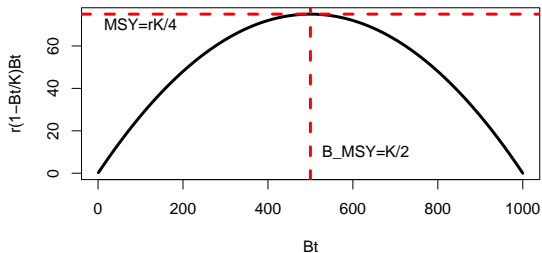
- 来年の資源量 = 今年の資源量 + 余剰生産 - 漁獲量
- 個体群内の年齢構造や加入変動などは考えない
- 対応する資源評価モデル：プロダクションモデル

## ● 年齢構造モデル

- 今年の  $a$  才の尾数 = 去年の  $a-1$  才の尾数  $\times$  生き残り率
- 個体群内の年齢構造、毎年の加入変動や親子関係を考慮
- 対応する資源評価モデル：VPA、Statistical catch at age model など

# 余剰生産モデル

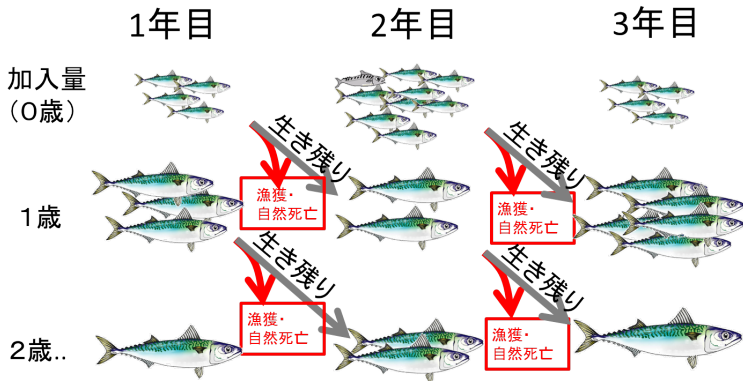
- 今年の資源量 = 去年の資源量 + 余剰生産 - 漁獲量
- $B_t = B_{t-1} + r(1 - \frac{B_{t-1}}{K})B_{t-1} - C_{t-1}$
- $MSY = rK/4, B_{MSY} = K/2$
- $C_t = E_t \times B_t$  とすると、 $E_{MSY} = r/2$



# 年齢構造モデル (個体数の動態)

今年の  $a$  才の尾数 = 去年の  $a-1$  才の尾数  $\times$  生き残り率

$$N_{at} = N_{a-1,t-1} \exp(-F_{at} - M_{at})$$





# 年齢構造モデル (漁獲量と加入量の動態)

- 漁獲量

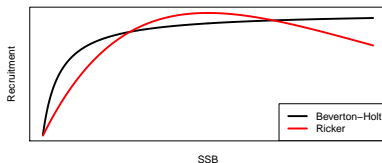
$$C_{at} = \frac{F_{at}}{(F_{at} + M_{at})} (N_{at} - N_{a-1,t-1}) \quad (1)$$

$$= \frac{F_{at}}{(F_{at} + M_{at})} (1 - \exp(-F_{at} - M_{at})) N_{at} \quad (2)$$

- 加入量 = f(親の量) + 年変動

- Beverton-Holt:  $R = aSSB / (1 + bSSB)$

- Ricker:  $R = aSSB \exp(-bSSB)$



## (2) Rを使って 余剰生産モデルの挙動を調べる

## (2) Rを使って余剰生産モデルの挙動を調べる

### 余剰生産モデル

- 来年の資源量 = 今年の資源量 + 余剰生産 - 漁獲量

#### ここでの流れ

- ① Rでプロットしてみよう
- ② 漁獲率  $E$  を変えると個体群はどのように変化するか？

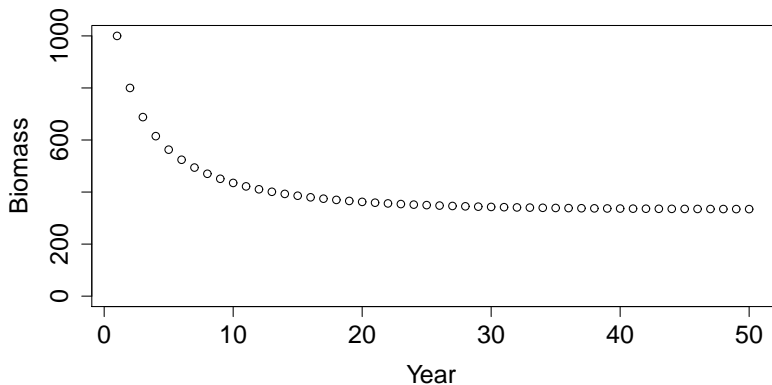
## 2. R でプロットしてみよう

$$B_{t+1} = B_t + r(1 - \frac{B_t}{K})B_t - C_t$$

```
nyear <- 50 # 計算を行う年数
r <- 0.3 # 増殖率
K <- 1000 # 環境収容力
B <- numeric()
B[1] <- K # 1年目の資源量はKと等しいとする
E <- rep(0.20, nyear) # 毎年2割ずつ漁獲
for(i in 2:nyear) # nyear回繰り返し計算する
  B[i] <- B[i-1] + r*(1-B[i-1]/K)*B[i-1] - E[i-1]*B[i-1]
```

```
# 結果のプロット
```

```
plot(B, ylab="Biomass",xlab="Year",ylim=c(0,max(B)))
```



```
# 関数化する
```

```
# SPmodel1; r, K, B1, E を与えて、余剰生産モデルにしたがった  
# 資源量 (B)、漁獲量 (C)、漁獲率 (E) を返す関数
```

```
SPmodel1<- function(r, K, B1, E, nyear=50)  
{  
  B <- numeric()  
  B[1] <- B1  
  for(i in 2:nyear){  
    B[i] <- B[i-1] + r*(1-B[i-1]/K)*B[i-1] - E[i-1]*B[i-1]  
  }  
  C <- E*B  
  return(list(B=B, C=C, E=E)) # 値を返す  
}  
# 使いかた  
res1 <- SPmodel1(0.3,1000,1000,rep(0.1,50))
```

## 2. 漁獲圧 $E$ を変えると個体群はどのように変化する か？

#  $E$  をいろいろ変えて結果に保存

```
res1 <- SPmodel1(0.3,1000,1000,rep(0.1,50))
```

```
res2 <- SPmodel1(0.3,1000,1000,rep(0.2,50))
```

```
res3 <- SPmodel1(0.3,1000,1000,rep(0.3,50))
```

# プロットする

```
par(mfrow=c(1,2),mar=c(4,4,2,1))
```

```
plot(res1$B,ylim=c(0,1000),ylab="Biomass",xlab="Year")
```

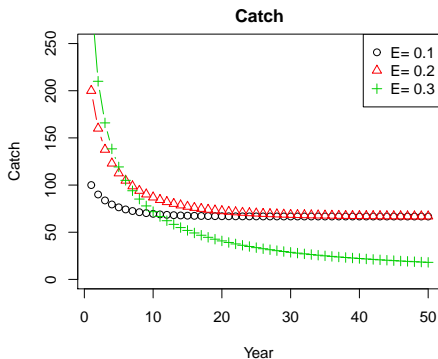
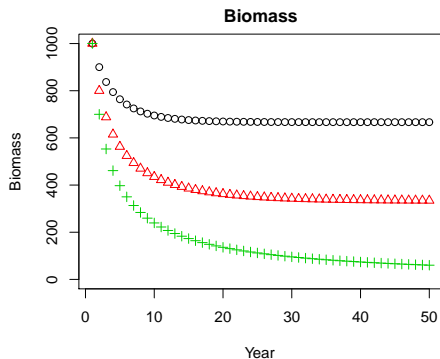
```
points(res2$B,col=2,pch=3)
```

```
points(res3$B,col=3,pch=3)
```

```
plot(res1$C,ylim=c(0,250),ylab="Catch",xlab="Year",type="b")
```

```
points(res2$C,col=2,pch=2,type="b")
```

```
points(res3$C,col=3,pch=3,type="b")
```



- 高い漁獲圧  $E = 0.3$  では最初の漁獲量が多いが、長いあいだ漁獲すると、資源も漁獲量も少なくなってしまう。



- $E_{MSY} = r/2$  より、 $r = 0.3$  のとき  $E_{MSY} = 0.3/2 = 0.15$

```
res4 <- SPmodel1(0.3,1000,1000,rep(0.15,50))
```

# さきの図に以下のプロットを追加

```
plot(res1$B,ylim=c(0,1000),ylab="Biomass",xlab="Year")
```

```
points(res2$B,col=2,pch=3)
```

```
points(res3$B,col=3,pch=3)
```

```
points(res4$B, col=4, pch=4)
```

```
plot(res1$C,ylim=c(0,250),ylab="Catch",xlab="Year",type="b")
```

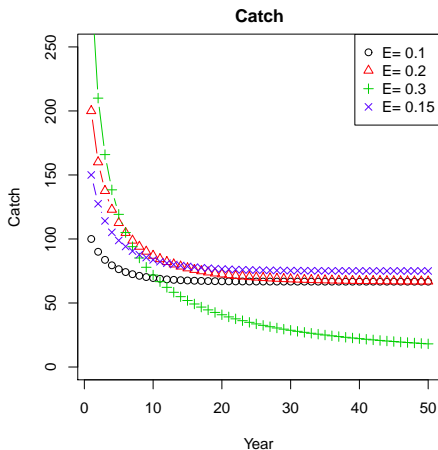
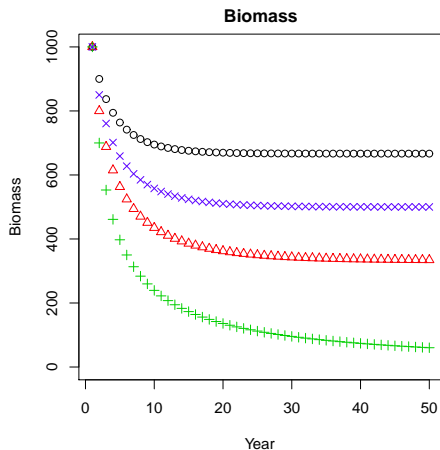
```
points(res2$C,col=2,pch=2,type="b")
```

```
points(res3$C,col=3,pch=3,type="b")
```

```
points(res4$C,col=4,pch=4)
```

```
legend('topright', col=1:4,pch=1:4, # 凡例
```

```
  legend=paste('E=',c(0.1,0.2,0.3,0.15)))
```



- MSY 付近では漁獲量レベルがほぼ同じにもかかわらず、資源量に大きな差がある。

## (3) 漁獲データから 資源状態を推定する

### (3) 漁獲データから資源状態を推定する

- 資源評価  
仮定した個体群動態モデルで 観察データを最も良く説明するパラメータを見つけること。それにより、現状の資源状態を推測する。
- 観察データ  
漁獲量と CPUE（プロダクションモデル）、年齢別漁獲尾数 (VPA)
- どうやって？  
最小二乗法・最尤法など
- ここでは、SPfunction.R に用意された関数を使って最小二乗法で推定

## ここでの流れ

- ① 擬似的なデータの生成・プロット
- ② モデルにあてはめて資源評価をする
- ③ 繰り返しやってパフォーマンスを評価

# 1. 擬似データの作成・プロット

```
source("script2_fun.r") # 関数群の読み込み
# 漁獲率は毎年少しづつ上昇するとする
Et <- seq(from=0,to=0.3,length=50)
plot(Et)

# 観測誤差 0.2 の場合
set.seed(1)
data1 <- SPobs(0.3,1000,1000, # 仮定する r,K,B1 の値
               Et, # 50 年分の漁獲率,
               1, # CPUE = q * B の q
               0.2) # CPUE の誤差の大きさ (0.2)

# 観測誤差 0.4 の場合
data2 <- SPobs(0.3,1000,1000,Et,1,0.4)

# 生成したデータと真の資源量のプロット
par(mfrow=c(1,2),mar=c(4,4,2,1))
plot(data1); title("SD=0.2")
plot(data2); title("SD=0.4")
```

## 2. モデルにあてはめて資源評価をする

# データの生成

```
set.seed(1)
```

```
data1 <- SPobs(0.3,1000,1000,Et,1,0.2)
```

# パラメータ推定

```
(est1 <- SPest(what.est=1:3, # 何のパラメータを推定するか?
```

```
                # (1: r, 2: K, 3: B1)
```

```
                data=data1, # 漁業データ
```

```
                init=c(0.3,1000,1000))) # 初期値
```

```
pred1 <- predict(est1) # 推定されたパラメータで予測
```

```
par(mfrow=c(1,3),mar=c(4,4,2,1))  
plot(data1); title('Observation') # データのプロット  
plot(pred1); title('Expected') # 予測結果のプロット  
matplot(cbind(data1$B,pred1$B),pch=1:2) # 真の資源量との比較  
legend('topright',col=1:2,legend=c('True','Estimation'),pch=1:2)
```



### 3. 繰り返しやってパフォーマンスを評価

```
# 例えば、100回、データをランダムに発生させて推定することを繰り返す
# 時間がかかります。かわりに、load("Robj/par.mat1.R") でも
N <- 100
par.mat1 <- matrix(0,N,3)
set.seed(1)
for(i in 1:N){
  # データ生成
  data1 <- SPobs(0.3,1000,1000,Et,1,0.2)
  # パラメータ推定
  par.mat1[i,] <- SPest(what.est=1:3,data=data1,
                       init=c(0.3,1000,1000))$par
}
```

```
# 推定結果の図示（縦線が真の値）
```

```
par(mfrow=c(2,2),mar=c(3,3,1,1))
```

```
# r
```

```
hist(par.mat1[,1],col="pink",nclass=20,main='r')
```

```
abline(v=0.3,col=2,lwd=3)
```

```
# K
```

```
hist(par.mat1[,2],col="pink",nclass=20,main='K')
```

```
abline(v=log(1000),col=2,lwd=3)
```

```
# B1
```

```
hist(par.mat1[,3],col="pink",nclass=20,main='B1')
```

```
abline(v=log(1000),col=2,lwd=3)
```

## CPUEの精度がより悪い場合 (CV=0.2から0.4へ)

```
# より時間がかかります。
# かわりに、load("Robj/par.mat1.R") でも
N <- 100
par.mat2 <- matrix(0,N,3)
set.seed(1)
for(i in 1:N){
  # データ生成
  data1 <- SPobs(0.3,1000,1000,seq(from=0,to=0.3,length=50),1,0.4)
  # パラメータ推定
  par.mat2[i,] <- SPest(what.est=1:3,data=data1,init=c(0.3,1000,1000))$par
}

# 推定結果の図示 (縦線が真の値)
par(mfrow=c(2,2),mar=c(3,3,1,1))
hist(par.mat2[,1],col="pink",nclass=20,main='r')
abline(v=0.3,col=2,lwd=3)
hist(par.mat2[,2],col="pink",nclass=20,main='K')
abline(v=log(1000),col=2,lwd=3)
hist(par.mat2[,3],col="pink",nclass=20,main='B1')
abline(v=log(1000),col=2,lwd=3)
```

```
# SD=0.2 のときと 0.4 のときの r, K の推定値の分布を重ねて書く
par(mfrow=c(1,2))
# r
plot(density(par.mat1[,1]),type='l',lwd=3,
      xlim=c(0.20,0.40),main='r')
points(density(par.mat2[,1]),type='l',lwd=3,col=2)
# K
plot(density(exp(par.mat1[,2])),type='l',lwd=3,
      xlim=c(800,1300),main='K')
points(density(exp(par.mat2[,2])),type='l',lwd=3,col=2)

legend('topright',legend=c('SD=0.2','SD=0.4'),col=1:2,lwd=3)
```

## それぞれの場合で推定された MSY は？

```
MSY1 <- par.mat1[,1]*exp(par.mat1[,2])/4 # MSY=r*K/4
```

```
MSY2 <- par.mat2[,1]*exp(par.mat2[,2])/4
```

```
BMSY1 <- exp(par.mat1[,2])/2 # B_MS Y=K/2
```

```
BMSY2 <- exp(par.mat2[,2])/2
```

```
plot(density(MSY1),type='l',lwd=3,main='MSY',  
      xlim=c(60,80))
```

```
points(density(MSY2),type='l',lwd=3,col=2)
```

```
plot(density(BMSY1),type='l',lwd=3,main='B_MS Y',  
      xlim=c(400,700))
```

```
points(density(BMSY2),type='l',lwd=3,col=2)
```

```
legend('topright',legend=c('SD=0.2','SD=0.4'),col=1:2,lwd=3)
```

## 一旦まとめ

- 観測されたデータ（CPUE, 漁獲量）に余剰生産モデルをフィットさせてパラメータを推定した
- シミュレーションモデルで仮定した真の値と推定値を比較した
- シミュレーションからわかったこと：CPUEの精度が大切
- **プラス $\alpha$** ：プロダクションモデルで精度の良い推定が得られるのは、どのような条件なときか？漁獲率やB1などを変えて、いろいろ調べてみよう
- **さらなる発展**：Bayesian production model（ $r$ ,  $K$ の事前情報の利用）、状態空間モデル（プロセス誤差の考慮）等も用いられる

## (4) 資源評価結果の「診断」: 不確実性の評価

## (4) 資源評価結果の「診断」：不確実性の評価

実際には「真の値」を知ることはできない → 推定値がどのくらい確実か？またはどのくらい不確実か？を評価する必要がある

**感度分析** 資源評価モデルで仮定したパラメータを変えたときに結果がどう変わるか？（例えば、 $B1 = K$  や自然死亡の仮定など）  
（モデルの安定性の検討）

**ブートストラップ** データをランダムにリサンプリングして、推定を繰り返す（信頼区間の推定）

**ジャックナイフ** データを1つずつ順番に除去して、推定を繰り返す（影響力の強いデータ・外れ値の検出）

**尤度プロファイル** 推定パラメータの1つ（または複数）を徐々に変化させたときに、目的関数がどのように変化するかを見る（信頼区間の推定）



## ここでの流れ

- ① ジャックナイフ
- ② 尤度プロファイル

とやってみる

## ジャックナイフ

# ひとつの推定例をもってくる

```
set.seed(10)
data1 <- SPobs(0.3,1000,1000,seq(from=0,to=0.3,length=50),1,0.4)
est3 <- SPest(what.est=c(1:3),data=data1,init=c(0.3,1000,1000))
plot(data1)
plot(predict(est3))
```

# 50年分 CPUE データがあるので、50回繰り返し

# これも時間がかかります。

```
# load("Robj/par.mat3.R"); load("Robj/pred.list.R")
par.mat3 <- matrix(0,50,3)
pred.list <- list()
for(i in 1:50){
  tmpdata <- data1
  tmpdata$cpue.obs[i] <- NA # i番目のCPUEをNA(欠損)とする
  tmp <- SPest(what.est=c(1:3),data=tmpdata,init=c(0.3,1000,1000))
  par.mat3[i,] <- tmp$par
  # 予測値を見たいので、予測値を毎回保存
  pred.list[[i]] <- predict(tmp)
}
```

```
# 資源量の予測値を 50 回分プロット
```

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

```
biom <- sapply(pred.list,function(x) x$B)
```

```
matplot(biom,type="l",col='gray',lwd=2)
```

```
# 最初の 1 年目と 3 年目を除去した結果
```

```
matpoints(biom[,c(1,3)],type='b',lwd=2,col=2,pch=1:2)
```

```
points(predict(est3)$B,type='b',lwd=2,pch=3)
```

```
points(data1$cpue,pch=4)
```

```
legend('topright',pch=c(1:4),lwd=2,col=c(2,2,1,1),
```

```
      legend=c('Remove 1st year','Remove 3rd year',
```

```
              'Estimated','Observed CPUE'))
```

## 尤度プロファイル

```
# これも時間が、、、  
# load("Robj/LL.matrix.R")  
# r を動かす範囲  
r.range <- seq(from=0.1,to=0.4,by=0.005)  
# K を動かす範囲  
K.range <- seq(from=700,to=1500,by=25)  
  
LL.matrix <- matrix(0,length(r.range),length(K.range))  
for(i in 1:length(r.range)){  
  for(j in 1:length(K.range)){  
    # r, K は固定して B1 のみ推定し、目的関数の値を  
    # LL.matrix に入れる  
    LL.matrix[i,j] <- SPest(what.est=c(3),data=data1,  
                           init=c(r.range[i],K.range[j],1000))$value  
  }  
}
```

## # 結果のプロット

```
par(mfrow=c(1,2))
```

```
image(r.range,K.range,log(LL.matrix),col=terrain.colors(100))  
contour(r.range,K.range,log(LL.matrix),add=TRUE)  
points(est3$par[1],exp(est3$par[2]),col="white",cex=2,pch=3)
```

## # 2色みの図

```
image(r.range,K.range,log(LL.matrix),col=c(2,"pink"),  
      breaks=c(0,2,100))  
contour(r.range,K.range,log(LL.matrix),add=TRUE)  
points(est3$par[1],exp(est3$par[2]),col="white",cex=2,pch=3)
```

## ここまでのまとめ

- 資源評価後のモデル診断は重要
- 個々のデータやモデルの仮定の影響力を知る
- 不確実性を定量評価することにより、リスクを考慮
- **プラス $\alpha$**  ブートストラップや感度分析もやってみてください

## (5) 年齢構造モデル： より複雑な、

## (5) 年齢構造モデル

- $N_{at} = N_{a-1,t-1} \exp(-F_{at} - M_{at})$

### ここでの流れ

余剰生産モデルと同様に、

- ① シミュレーションデータの作成
- ② VPA を使った資源評価
- ③ モデル診断（感度分析）

とやってみる



# 1. シミュレーションデータの作成

```
source('script2_fun.r')
# データ生成のための関数 ASobs
set.seed(10)
data2 <- ASobs() # 引数なし=デフォルトの引数が使われる
plot(data2) # 加入のランダム変動なし=プロダクションモデルと同じようなかんじ

data3 <- ASobs(Rsigma=0.3) # 毎年の加入をランダム変動
plot(data3) # 資源量の変動

#(寄り道) プロダクションモデルをあてはめてみる
(spest1 <- SPest(what.est=c(1:3),data=data2,init=c(0.3,3000,3000)))
(spest2 <- SPest(what.est=c(1:3),data=data3,init=c(0.3,3000,3000)))

# 加入のランダム変動なしの場合
matplot(cbind(predict(spest1)$B,data2$B),type='b',pch=1:2)

# 加入のランダム変動ありの場合
matplot(cbind(predict(spest2)$B,data3$B),type='b',pch=1:2)
```

## 2. VPA (virtual population analysis)

- 年齢構造すべて ( $N_{at}, F_{at}$ ) を推定したい
  - より多くのデータが必要
  - 年齢別漁獲尾数  $C_{at}$  の利用
- 漁獲方程式  $C_{at} = F_{at}/(F_{at} + M_{at})(N_{at} - N_{a+1,t+1})$  より、 $C_{at}, N_{a+1,t+1}, M_{at}$  がわかれば、 $F_{at}$  は求まるはず
- Pope の近似式  $N_{at} = N_{a+1,t+1} \exp(M_{at}) + C_{at} \exp(M_{at}/2)$
- さらに、高齢では  $F$  が等しい、近年は選択率にあまり変化がない、といった仮定をおくことで  $N_{at}, F_{at}$  を芋づる式に計算していく
- 実際の計算は、VPA 計算用に用意された関数 (rvpa0.7.r) を使う

## VPA のあてはめ

```

source("rvpa0.7.r")
data3 <- ASobs(Rsigma=0.3) # 毎年の加入をランダム変動
vdat <- list()
vdat$caa <- vdat$M <- vdat$maa <- vdat$waa <- data3$caa
# 生物パラメータ (シミュレーションで仮定したものと同一)
vdat$waa[] <- 1 # ここでは年齢別体重は考えない (本当は重要)
vdat$maa[] <- c(rep(0,4),rep(1,6)) # 成熟率
vdat$M[] <- 0.3 # 自然死亡率
# VPA の実行
vres1 <- vpa(vdat,fc.year=47:49,
             tf.year = 47:49, #最終年の選択率を参照する期間
             term.F="max",stat.tf="mean",
             Pope=TRUE,tune=FALSE,p.init=0.5)
# 推定値と真の値の比較
plot(colSums(vres1$naa),type="b",
     ylim=c(0,max(colSums(vres1$naa),data3$B)))
points(data3$B,type="b",col=2)
legend("topright",col=1:2,pch=c(1,1),legend=c("Estimation","True"))

```

### 3. モデル診断 (感度分析, M)

```
# 仮定する M を真の値から変える
vdat2 <- vdat
vdat2$M[] <- 0.2 # の真値 (0.3) から 0.2 へ
# VPA の実行
vres2 <- vpa(vdat2,fc.year=47:49,
             tf.year = 47:49,
             term.F="max",stat.tf="mean",
             Pope=TRUE,tune=FALSE,p.init=0.5)

plot(colSums(vres1$naa),type="b",
     ylim=c(0,max(colSums(vres1$naa),data3$B)))
points(data3$B,type="b",col=2)
points(colSums(vres2$naa),type="b",col=3)
legend("topright",col=1:3,pch=c(1,1),
      legend=c("Estimation","True","Higher M"))
```

# モデル診断: 感度分析 (年齢別漁獲尾数)

```

# これも時間が、、
# load("Robj/est.naa.R")
set.seed(1)
est.naa <- matrix(0,50,100)
vres3 <- list()
for(i in 1:100){
  tmpdat <- vdat
  # 年齢別漁獲尾数に多項分布の誤差があるとする
  tmp <- apply(vdat$caa,2,function(x)
              rmultinom(n=1,size=1000,prob=x))
  tmp <- sweep(sweep(tmp,2,apply(tmp,2,sum),FUN="/"),
              2,apply(vdat$caa,2,sum),FUN="*")
  tmpdat$caa[] <- tmp # caa の置き換え
  # VPA の実行
  vres3[[i]] <- vpa(tmpdat,fc.year=47:49,
                  tf.year = 47:49,
                  term.F="max",stat.tf="mean",
                  Pope=TRUE,tune=FALSE,p.init=0.5)
  est.naa[,i] <- colSums(vres3[[i]]$naa)
}

boxplot(t(est.naa),ylim=c(0,4000))
points(data3$B,type="b",col=2)

```

## ここまでのまとめ

- 年齢構造モデル (VPA) では、より詳細なデータ（体長組成や年齢組成）が必要だが、もっと多くのことを知れる（選択率・毎年の加入変動など）
- **（プラス $\alpha$ ）** どういう条件のときに、年齢構造個体群動態のもとでもプロダクションモデルのあてはめが有効か？また、有効でない場合は？（加入変動の大きさ、年齢別の選択率、漁獲圧のパターンなど）
- **（プラス $\alpha$  2）** VPAによる資源評価の誤りはどのような原因からくると考えられるか？
- **（さらなる発展）** 統合型資源評価モデル

## おわりに：Rを使った資源評価

- 資源評価の理解  
どのような個体群の動態が仮定されているか？の理解
- 資源評価する前に  
シミュレーションデータを使った資源評価モデルの挙動の理解
- 資源評価の後には  
さまざまなモデル診断による不確実性の評価
- Rを使った資源評価  
エクセルでもできる？ しかし、Rでやることにより、簡単に拡張が可能（ブートストラップなどの繰り返し計算）