

生物資源解析学演習 2022 Lecture 2

R の使い方基礎 (2): 確率分布 + モンテカルロ法 (参考資料)

北門 利英 (東京海洋大学海洋生物資源学科)

2022 年 10 月 11 日

個体群動態モデル (参考までに, また今後扱います)

Pella-Tomlinson 余剰生産モデル (確率変動なし)

モデルの定義

$$P_{t+1} = P_t + rP_t \left\{ 1 - \left(\frac{P_t}{K} \right)^z \right\} - C_t$$

初歩的に漸化式を利用して個体群動態を計算

```
r <- 0.2
K <- 10000
z <- 1
TT <- 50

Catch <- rep(500, TT)

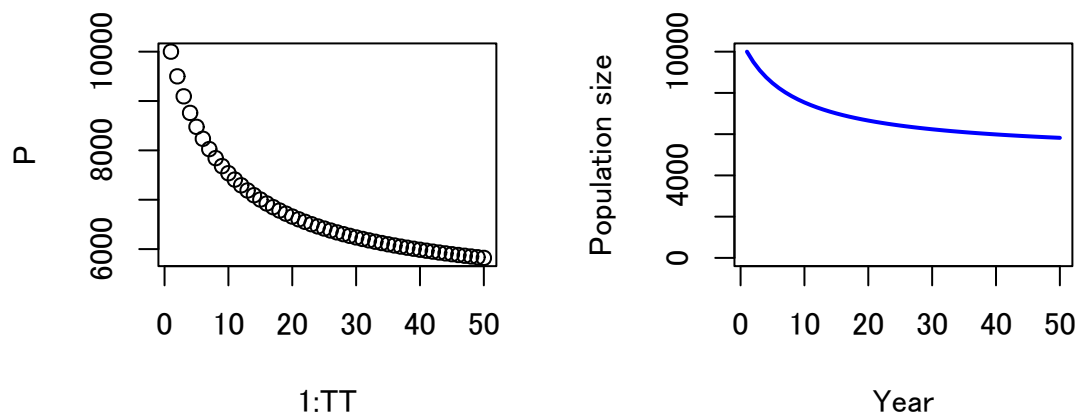
TT1 <- TT-1
P <- numeric(TT)
P[1] <- K
for(t in 1:TT1)
{
  tmp <- P[t] + r*P[t]*(1-(P[t]/K)^z) - Catch[t]
  P[t+1] <- max(tmp, 0.001)
}

print(P, digits=0)
```

```
[1] 10000 9500 9095 8760 8477 8235 8026 7843 7681 7537 7409 7293 7187 7092 7004
[16] 6924 6850 6781 6718 6659 6604 6552 6504 6459 6416 6376 6338 6303 6269 6236
[31] 6206 6177 6149 6123 6097 6073 6050 6028 6007 5987 5967 5949 5931 5913 5897
[46] 5881 5865 5850 5836 5822
```

計算結果の図示

```
par(mfrow=c(1,2))
plot(1:TT, P)
plot(1:TT, P, type="l", lwd=2, col="blue", xlab="Year", ylab="Population size", ylim=c(0,K))
```



もう少し進んだ方法：個体群動態計算用関数 `PDM.PT` を作成

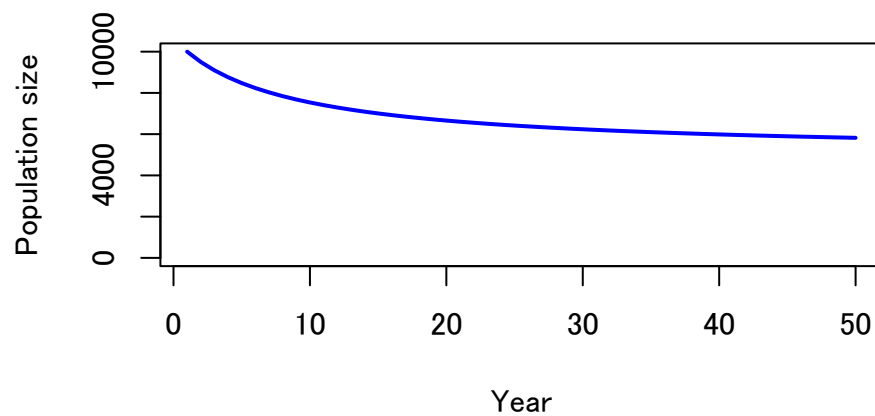
```
#Pella-Tomlinson Production model (without stochastic process error)
PDM.PT<-function(r, K, z, P1, TT, Catch)
{
  TT1 <- TT-1
  P <- numeric(TT)
  P[1] <- P1
  for(t in 1:TT1)
  {
    tmp <- P[t] + r*P[t]*(1-(P[t]/K)^z) - Catch[t]
    P[t+1] <- max(tmp, 0.001)
  }
  return(P)
}
```

新しく作った個体群動態計算用関数 `PDM.PT` を実行

```
Res <- PDM.PT(r=0.2, K=10^4, z=1, P1=10^4, TT=50, Catch=rep(500,TT))
print(Res, digits=0)

[1]10000 9500 9095 8760 8477 8235 8026 7843 7681 7537 7409 7293 7187 7092 7004
[16] 6924 6850 6781 6718 6659 6604 6552 6504 6459 6416 6376 6338 6303 6269 6236
[31] 6206 6177 6149 6123 6097 6073 6050 6028 6007 5987 5967 5949 5931 5913 5897
[46] 5881 5865 5850 5836 5822

plot(1:TT, Res, type="l", lwd=2, col="blue", xlab="Year", ylab="Population size", ylim=c(0,K))
```



Pella-Tomlinson 余剰生産モデル (確率変動あり)

モデルの定義

$$P_{t+1} = \left[P_t + rP_t \left\{ 1 - \left(\frac{P_t}{K} \right)^z \right\} - C_t \right] * \exp(\varepsilon_t), \quad \varepsilon_t \sim N(0, \sigma^2)$$

さっき作った個体群動態計算用関数 PDM.PT を書き換え (上書き更新)

```
#Pella-Tomlinson Production model (with stochastic process error)
PDM.PT<-function(r, K, z, P1, TT, Catch, sigma=0)
{
  TT1 <- TT-1
  P <- numeric(TT)
  Epsilon <- rnorm(n=TT, mean=0, sd=sigma)
  P[1] <- P1
  for(t in 1:TT1)
  {
    tmp <- P[t] + r*P[t]*(1-(P[t]/K)^z) - Catch[t]
    tmp <- tmp*exp(Epsilon[t])
    P[t+1] <- max(tmp, 0.001)
  }
  return(P)
}
```

再定義した PDM.PT を実行

```
K <- 10^4
TT <- 50

Res <- PDM.PT(r=0.2, K=K, z=1, P1=K, TT=50, Catch=rep(500,TT), sigma=0.05)
print(Res, digits=0)
```

```

[1]10000 9055 9535 9107 8093 7651 7444 7836 7771 7270 7858 8089 7550 6775 6338
[16] 6024 6335 6307 6845 6572 6259 6088 6204 5926 5414 5168 5413 5391 5197 5331
[31] 5346 5690 5775 6431 6142 6660 6404 6079 5999 6020 6141 6253 6400 6447 6351
[46] 6404 6251 6056 6130 5881

plot(1:TT, Res, type="l", col="blue", xlab="Year", ylab="Population size", ylim=c(0,K))

Res <- PDM.PT(r=0.2, K=K, z=1, P1=K, TT=50, Catch=rep(500,TT), sigma=0.05)
print(Res, digits=0)

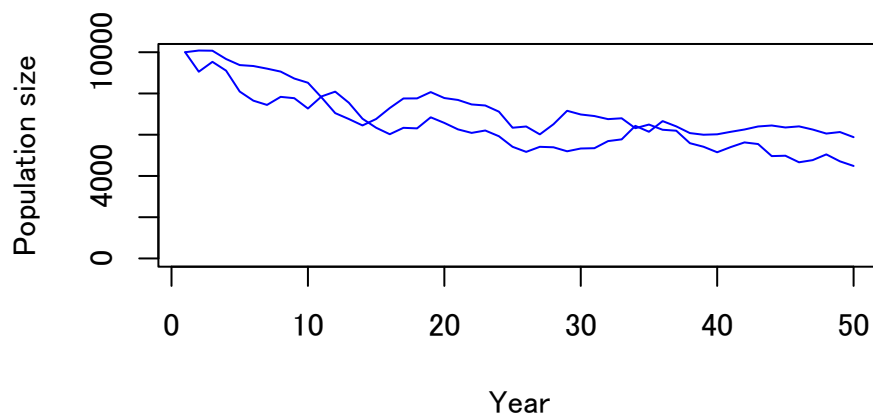
```

```

[1]1.000000e+041.008264e+041.007309e+049.670508e+039.376968e+039.331075e+039.203935e+039.061061e+038.72
[16]7.294395e+037.757264e+037.761574e+038.066901e+037.778622e+037.685507e+037.468227e+037.416880e+037.1
[31]6.905863e+036.755827e+036.796666e+036.315154e+036.493201e+036.240268e+036.194311e+035.589526e+035.4
[46]4.663132e+034.768021e+035.045899e+034.708935e+034.485068e+03

points(1:TT, Res, type="l", col="blue")

```



PDM.PT を繰り返し実行して結果を図示

```

Nsim <- 100
Pmat <- array(NA, c(Nsim, TT))
plot(0, type="n", xlab="Year", ylab="Population size", xlim=c(0,TT), ylim=c(0,K))

for(i in 1:Nsim)
{
  Pmat[i,] <- PDM.PT(0.2, 10^4, 1, 10^4, 50, rep(500,TT), 0.05)
  points(1:TT, Pmat[i,], type="l", col="lightblue")
}

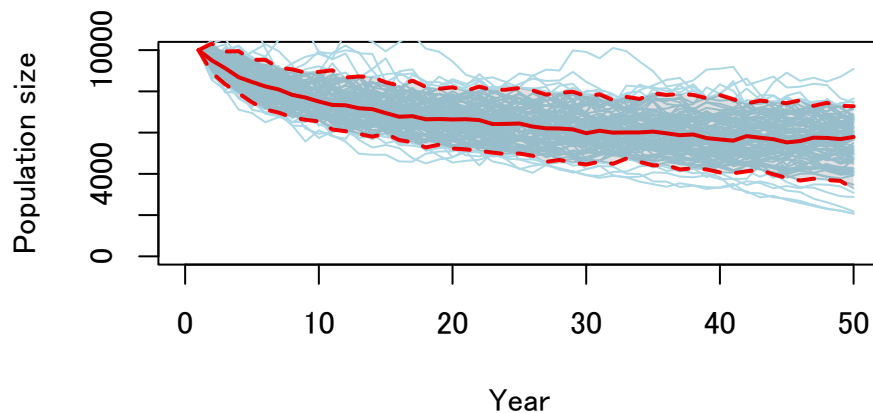
P.med <- apply(Pmat, 2, median);
P.L5per <- apply(Pmat, 2, quantile, 0.05)
P.U5per <- apply(Pmat, 2, quantile, 0.95)
points(1:TT, P.med, type="l", lwd=2, col="red")
points(1:TT, P.L5per, type="l", lty=2, lwd=2, col="red")
points(1:TT, P.U5per, type="l", lty=2, lwd=2, col="red")

```

```

polygon(
  c(1:TT, TT:1), c(P.L5per, rev(P.U5per)),
  col = "#00000020", border = NA)

```



上記を関数化

```

PDM.PT.sim <- function(r, K, z, P1, TT, Catch, sigma=0, Nsim)
{
  Pmat <- array(NA, c(Nsim, TT))
  plot(0, type="n", xlab="Year", ylab="Population size",
       xlim=c(0,TT), ylim=c(0,K), xaxs="i", yaxs="i")

  for(i in 1:Nsim)
  {
    Pmat[i,] <- PDM.PT(0.2, 10^4, 1, 10^4, 50, rep(500,TT), 0.05)
    points(1:TT, Pmat[i,], type="l", col="lightblue")
  }
  P.med <- apply(Pmat, 2, median);
  P.L5per <- apply(Pmat, 2, quantile, 0.05)
  P.U5per <- apply(Pmat, 2, quantile, 0.95)
  points(1:TT, P.med, type="l", lwd=2, col="red")
  points(1:TT, P.L5per, type="l", lty=2, lwd=2, col="red")
  points(1:TT, P.U5per, type="l", lty=2, lwd=2, col="red")

  polygon( c(1:TT, TT:1), c(P.L5per, rev(P.U5per)), col = "#00000020", border = NA)

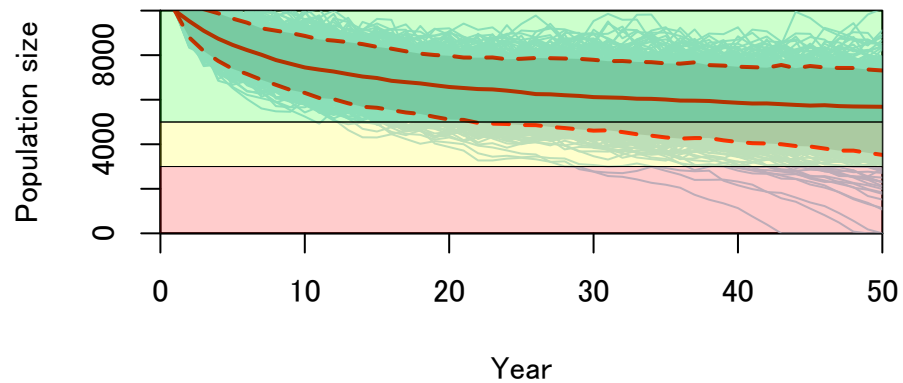
  rect(xleft=-1, ybottom=0, xright=TT, ytop=0.3*K, lwd=0, col=rgb(1,0,0,alpha=0.2))
  rect(xleft=-1, ybottom=0.3*K, xright=TT, ytop=0.5*K, lwd=0, col=rgb(1,1,0,alpha=0.2))
  rect(xleft=-1, ybottom=0.5*K, xright=TT, ytop=K, lwd=0, col=rgb(0,1,0,alpha=0.2))

  return(Pmat)
}

```

実行！

```
Res <- PDM.PT.sim(0.2, 10^4, 1, 10^4, 50, rep(500,TT), 0.05, Nsim=1000)
```



```
nf <- layout(matrix(c(1,2,0,0),2,2,byrow=TRUE), c(3,1), c(4,0), FALSE)
par(mar=c(4,4,3,1))
Res <- PDM.PT.sim(0.2, 10^4, 1, 10^4, 50, rep(500,TT), 0.05, Nsim=1000)
par(mar=c(3,0,2,1))
Hist.final <- hist(Res[,TT], breaks=seq(0,1.2*K,500), plot=FALSE)
top <- max(Hist.final$counts)
barplot(Hist.final$counts, axes=FALSE, xlim=c(0, top), space=0, horiz=TRUE)
```

