FAS6337C - Lab 3

Marcel Gietzmann-Sanders

Load and Split the Data

```
setwd("/workspaces/schooling/population_dynamics/lab_3/")
trout_data <- read.table("data/trout.txt", header=T, sep="")
head(trout_data)
```

```
##               bay  tl   sex annuli age yearsold
## 1 CharlotteHarbor 387 FALSE      1   1    1.652
## 2 CharlotteHarbor 355 FALSE      1   1    1.652
## 3 CharlotteHarbor 355 FALSE      1   1    1.652
## 4 CharlotteHarbor 320 FALSE      1   1    1.652
## 5 CharlotteHarbor 335 FALSE      1   1    1.652
## 6 CharlotteHarbor 410 FALSE      1   1    1.652
```

```
ch_data <- na.omit(
    trout_data[trout_data$bay == 'CharlotteHarbor',]
)
ch_data <- ch_data[order(ch_data$yearsold),]
ir_data <- na.omit(
    trout_data[trout_data$bay == 'IndianRiver',]
)
ir_data <- ir_data[order(ir_data$yearsold),]
```

Function to Map a Vector to Parameters

```
map_columns <- function(v, cols) {
    c <- rep(0, 8)
    i <- 0
    for (col in cols) {
        i <- i + 1
        if (endsWith(col, 'Linf')) {
            j <- 1
        } else if (endsWith(col, 'vbk')) {
            j <- 3
        } else if (endsWith(col, 'tknot')) {
            j <- 5
        } else {
            j <- 7
        }

        if (startsWith(col, 'ch_')) {
            c[j] <- v[i]
        } else if (startsWith(col, 'ir_')) {
            c[j+1] <- v[i]
        } else {
            c[j] <- v[i]
            c[j+1] <- v[i]
        }
    }
    return(c)
}

map_columns(c(1, 3, 2, 3, 4), c('ir_Linf', 'ch_Linf', 'vbk', 'tknot', 'sig'))
```

```
## [1] 3 1 2 2 3 3 4 4
```

Basic Functions

```
predict_length <- function(yearsold, Linf, vbk, tknot) {
  pred_tl <- Linf * (1 - exp(-vbk * (yearsold - tknot)))
  return(pred_tl)
}

get_likelihood <- function(yearsold, tl, Linf, vbk, tknot, sig) {
  pred_tl <- predict_length(yearsold, Linf, vbk, tknot)
  NLL <- -1 * sum(dnorm(tl, pred_tl, sig, log=T), na.rm=T)
  return(NLL)
}
```

Function to Fit on Arbitrary Columns

```
do_likelihood_fit <- function(v, cols, runs) {
  objective <- function(v) {
    c <- map_columns(v, cols)

    ch_Linf <- exp(c[1])
    ch_vbk <- c[3]
    ch_tknot <- c[5]
    ch_sig <- exp(c[7])

    ir_Linf <- exp(c[2])
    ir_vbk <- c[4]
    ir_tknot <- c[6]
    ir_sig <- exp(c[8])

    ch_NLL <- get_likelihood(ch_data$yearsold, ch_data$tl, ch_Linf, ch_vbk, ch_tknot, ch_sig)
    ir_NLL <- get_likelihood(ir_data$yearsold, ir_data$tl, ir_Linf, ir_vbk, ir_tknot, ir_sig)

    NLL <- ch_NLL + ir_NLL
    return(NLL)
  }

  for (i in 1:runs) {
    fit <- optim(v, objective, hessian=T)
    v <- fit$par
  }
  return(fit)
}
```

Let’s Try It!

To share a parameter just make sure it doesn’t have a prefix (‘ch_’ or ‘ir_’).

We can start by sharing L_∞:

```
v <- c(
  6.7, 0.2, -1.5,  3.7,
  0.2, -1.2,  4.1
)
cols <- c(
  'Linf', 'ch_vbk', 'ch_tknot', 'ch_sig',
  'ir_vbk', 'ir_tknot', 'ir_sig'
)
(fit <- do_likelihood_fit(v, cols, 25))
```

```
## $par
## [1]  6.8068761  0.1532091 -1.8059854  3.7401227  0.2239987 -1.0478161  4.0655605
##
## $value
## [1] 12239.52
##
## $counts
## function gradient
##      192       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##              [,1]          [,2]          [,3]          [,4]          [,5]
## [1,] 188324.30850  5.088257e+05 -1.887554e+04 -5.441198e+01  2.218592e+05
## [2,] 508825.67475  2.337803e+06 -8.853964e+04  8.041794e+00 -9.094947e-07
## [3,] -18875.53594 -8.853964e+04  3.518782e+03  3.565174e-01  0.000000e+00
## [4,]    -54.41198  8.041794e+00  3.565174e-01  2.204096e+03 -9.094947e-07
## [5,] 221859.22437 -9.094947e-07  0.000000e+00 -9.094947e-07  6.574851e+05
## [6,] -15462.87936  0.000000e+00  0.000000e+00  0.000000e+00 -4.738922e+04
## [7,]     54.78169  0.000000e+00  0.000000e+00  0.000000e+00  2.943767e+00
##               [,6]          [,7]
## [1,] -1.546288e+04  5.478169e+01
## [2,]  0.000000e+00  0.000000e+00
## [3,]  0.000000e+00  0.000000e+00
## [4,]  0.000000e+00  0.000000e+00
## [5,] -4.738922e+04  2.943767e+00
## [6,]  3.650500e+03 -2.950856e-03
## [7,] -2.950856e-03  2.390953e+03
```

Or we can share σ:

```
v <- c(
  6.7, 0.2, -1.5,  3.7,
  6.9,  0.2, -1.2
)
cols <- c(
  'ch_Linf', 'ch_vbk', 'ch_tknot', 'sig',
  'ir_Linf', 'ir_vbk', 'ir_tknot'
)
(fit <- do_likelihood_fit(v, cols, 25))
```

```
## $par
## [1]  6.6676369  0.2069044 -1.4436927  3.9341872  6.8839452  0.1897212 -1.2165016
##
## $value
## [1] 12295.98
##
## $counts
## function gradient
##      182       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  7.578281e+04  2.335499e+05 -1.300820e+04   -0.55997748  0.000000e+00
## [2,]  2.335499e+05  7.323867e+05 -4.197229e+04   -2.61466539 -4.547474e-07
## [3,] -1.300820e+04 -4.197229e+04  2.540665e+03    0.22086579  4.547474e-07
## [4,] -5.599775e-01 -2.614665e+00  2.208658e-01 4593.40523503  2.110082e+00
## [5,]  0.000000e+00 -4.547474e-07  4.547474e-07    2.11008182  9.963053e+04
## [6,] -2.273737e-07  4.547474e-07 -2.273737e-07    9.33348656  3.583302e+05
## [7,] -4.547474e-07  0.000000e+00  4.547474e-07   -0.07147719 -1.990348e+04
##               [,6]          [,7]
## [1,] -2.273737e-07 -4.547474e-07
## [2,]  4.547474e-07  0.000000e+00
## [3,] -2.273737e-07  4.547474e-07
## [4,]  9.333487e+00 -7.147719e-02
## [5,]  3.583302e+05 -1.990348e+04
## [6,]  1.309389e+06 -7.485600e+04
## [7,] -7.485600e+04  4.555389e+03
```