

成長曲線を R で解析する

高井 省三

筑波大学大学院人間総合科学研究科

Rによる身長成長曲線の解析を概説する。Rは統計解析とグラフィックスの環境・言語であり、Linux, Mac, Windows の計算機で動く無料のアプリケーションである。Rの詳細は<http://www.r-project.org/>から得られる。Rのダウンロードは国内のサテライト<http://cran.md.tsukuba.ac.jp/>が便利だ。Rは商用アプリケーションのSのクローンともいわれるので、それらの日本語のテキスト（渋谷・柴田：Sによるデータ解析，共立出版，1992）も参考になる。さらに，Rの情報を集めるには，以下のURLなどをネットサーフするとよい：

<http://www.okada.jp.org/RWiki/index.php?RjpWiki>

<http://hosho.ees.hokudai.ac.jp/~kato/unix/R.html>

<http://aoki2.si.gunma-u.ac.jp/R/>

ここでは，Preece-Bains model 1の成長曲線モデル（Preece MA, Bains MJ (1978) A new family of mathematical models describing the human growth curve. Ann Hum Biol, 5: 1-24.）とロジスティック曲線をRの標準パッケージ nls をつかって解析する方法を紹介する。nls（nonlinear least squares）は最小自乗法による非線形モデルの当てはめに対するもっと有効なR関数である。

Rが起動してコマンドが入力できる状態になると，Rは>記号のプロンプトを表示する。以下で，入力するコマンドを太字で，Rからの出力は普通体字で表すことにする。これらの入出力の解説は#で始まる日本語テキストをつかうことにする。

```
# 13個の年齢データから関数cを使ってベクトルをつくり，tに代入(<-)する
> t <- c(6.527, 7.527, 8.605, 9.541, 10.551, 11.532, 12.580, 13.451,
14.448, 15.417, 16.416, 17.413, 18.283)
# ベクトルtを表示させる
> t
[1] 6.527 7.527 8.605 9.541 10.551 11.532 12.580 13.451 14.448 15.417
```

```

[11] 16.416 17.413 18.283
# 13 個の身長データからベクトル h をつくる
> h <- c(117.2, 122.1, 128.3, 133.1, 137.6, 146.0, 157.8, 167.5, 172.3,
174.0, 176.5, 176.5, 175.3)
# h を表示させる
> h
[1] 117.2 122.1 128.3 133.1 137.6 146.0 157.8 167.5 172.3 174.0 176.5 176.5
[13] 175.3
# ベクトル t とベクトル h からデータフレームをつくり, dat に代入する
# データフレームはラベル・行番号付きの行列のかたちをしている
# t と h の要素数は等しくなければならない。
> dat <- data.frame(t, h)
# データフレーム dat を表示させる
> dat
      t      h
1  6.527 117.2
2  7.527 122.1
3  8.605 128.3
4  9.541 133.1
5 10.551 137.6
6 11.532 146.0
7 12.580 157.8
8 13.451 167.5
9 14.448 172.3
10 15.417 174.0
11 16.416 176.5
12 17.413 176.5
13 18.283 175.3
# Preece-Bains model 1 (式 13) を非線形最小自乗パッケージ nls で解く
# 関数の式には「=」はつかわず, 「~」をつかう
# h1, h_theta, s0, s1, theta が求める 5 つの未知パラメータ
# 解析データは dat で, 初期値を 2 に設定している

```

```

# 詳しくは、help(nls)で nls のヘルプファイルを読んでください
# 結果を result_pb1 に代入して保存する
> result_pb1 <- nls(h ~ h1 - 2*(h1 - h_theta)/(exp(s0*(t - theta)) +
exp(s1*(t - theta))), data=dat, start=list(h1=2, h_theta=2, s0=2, s1=2,
theta=2))
# 不適切な初期値だったようで、結果が求まらないというエラーメッセージが出た
# このあたり、Excel のソルバーは鷹揚（優秀）なようで…
Error in nlsModel(formula, mf, start) : singular gradient matrix at initial parameter
estimates
# Preece-Bains 論文のイギリス児童のパラメータの平均値あたりの値を設定してみる
# キーボードの「↑」キーをたたくと、一つ前に入力したコマンドが表示される
# 「→」「←」「delete」キーをつかってカーソルを移動させて修正できる
> result_pb1 <- nls(h ~ h1 - 2*(h1 - h_theta)/(exp(s0*(t - theta)) +
exp(s1*(t - theta))), data=dat, start=list(h1=175, h_theta=162, s0=0.1,
s1=1.2, theta=15))
# 今度は Error が表示されず結果が求まったので表示させる
# 5つのパラメータと残差平方和が表示される
> result_pb1
Nonlinear regression model
  model: h ~ h1 - 2 * (h1 - h_theta)/(exp(s0 * (t - theta)) + exp(s1 * (t -
theta)))
  data: dat
           h1      h_theta      s0      s1      theta
175.81527731 159.52883175  0.09493085  1.34081477  12.72909836
  residual sum-of-squares:  4.220018
# もう少し詳しい結果を、summary 関数を使って、表示させる
> summary(result_pb1)

Formula: h ~ h1 - 2 * (h1 - h_theta)/(exp(s0 * (t - theta)) + exp(s1 *
(t - theta)))

Parameters:

```

```

          Estimate Std. Error t value Pr(>|t|)
h1      1.758e+02  3.987e-01  440.93 < 2e-16 ***
h_theta 1.595e+02  7.038e-01  226.67 < 2e-16 ***
s0      9.493e-02  7.254e-03   13.09 1.10e-06 ***
s1      1.341e+00  1.127e-01   11.89 2.29e-06 ***
theta   1.273e+01  8.808e-02  144.51 5.88e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 0.7263 on 8 degrees of freedom

Correlation of Parameter Estimates:

```

      h1 h_ s0 s1
h_theta  1
s0      . + 1
s1      . . , 1
theta    + , .

```

attr("legend")

```
[1] 0 ' 0.3 .' 0.6 `,' 0.8 `+' 0.9 `*' 0.95 `B' 1
```

```
# result_pb1 から5つの係数を取り出し (coef関数), リストに並べ (as.list関数) て,
# このセッションで利用できるようにする (attach関数)
```

```
# H1 <- h1 のように代入して保存しないと, このセッションが終わると消滅する
```

```
> attach(as.list(coef(result_pb1)))
```

```
# 5つのパラメータが変数として参照できるようになった
```

```
> h1
```

```
[1] 175.8153
```

```
> theta
```

```
[1] 12.72910
```

```
# Preece-Bains の式 (19), (9), (14) から, 成長曲線のパラメータを求める
```

```
# 式の展開はこの URL に詳しい:
```

```
# http://biking.taiiku.tsukuba.ac.jp/auxology/technote/PB1/index.html
```

```
> s1 <- (s0 + s1)/4 - sqrt((s0 + s1)^2/16 - s0*s1/2)
```

```

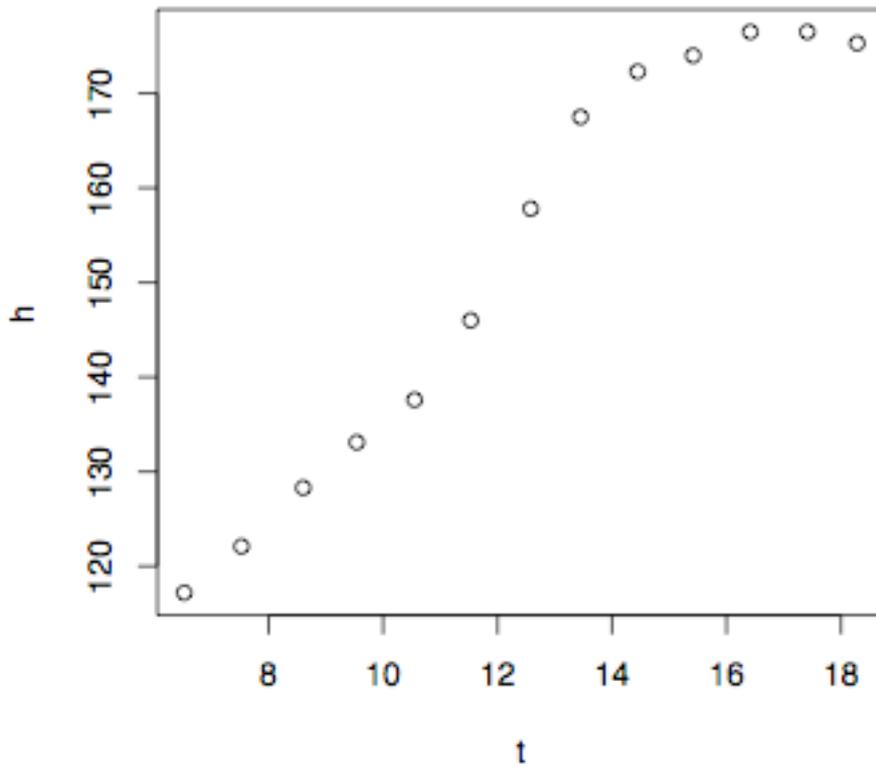
> sl
[1] 0.1036072
> sh <- (s0 + s1)/4 + sqrt((s0 + s1)^2/16 - s0*s1/2)
> sh
[1] 0.6142656
# Age at Minimal Velocity (take-off age)を求める
> amv <- theta + (log(sl - s0) - log(s1 - sl))/(s1 - s0)
> amv
[1] 8.74798
# Age at Peak Velocity (PHV age)を求める
> apv <- theta + (log(sh - s0) - log(s1 - sh))/(s1 - s0)
> apv
[1] 12.45960
# Distance at Minimal Velocity を求める
> dmvc <- h1 - ((2*(h1 - h_theta))/(exp(s0*(amv - theta)) + (exp(s1*(amv - theta))))))
> dmvc
[1] 128.6139
# Distance at Peak Velocity を求める
> dpvc <- h1 - ((2*(h1 - h_theta))/(exp(s0*(apv - theta)) + (exp(s1*(apv - theta))))))
> dpvc
[1] 156.3279
# Minimal Velocity を求める
> mvc <- sl*(h1 - dmvc)
> mvc
[1] 4.890402
# Peak Velocity を求める
> pvc <- sh*(h1 - dpvc)
> pvc
[1] 11.97045
# Preece-Bains model 1 曲線を描くために、6~19 歳を 100 等分する (seq 関数) 値

```

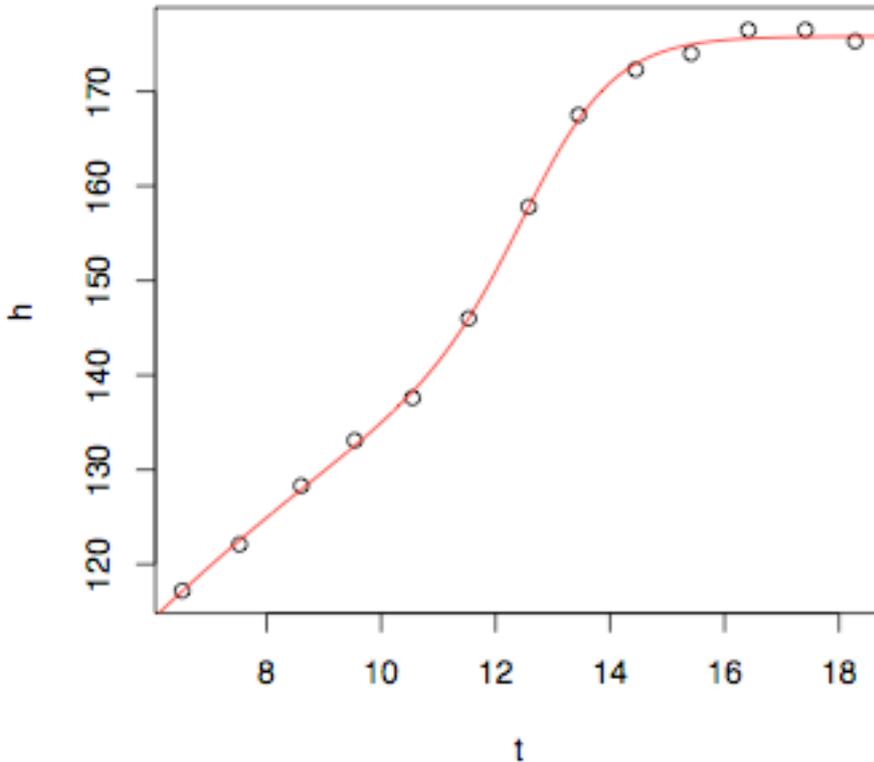
```

# を要素とする年齢のベクトル t_hat をつくる
> t_hat <- seq(6, 19, length=100)
> t_hat
 [1] 6.000000 6.131313 6.262626 6.393939 6.525253 6.656566 6.787879
 [8] 6.919192 7.050505 7.181818 7.313131 7.444444 7.575758 7.707071
# 中略
 [92] 17.949495 18.080808 18.212121 18.343434 18.474747 18.606061 18.737374
 [99] 18.868687 19.000000
# 5つの既知パラメータを代入した PBI モデル式に t_hat ベクトルを引数として予測値
# を求める関数 h_hat をつくる
# result_pb1 にある実際の値を使わなくとも, attach(as.list(coef(result_pb1)))で
# h1, theta 等を参照できるようにしてあるので, それらを利用する
> h_hat <- function(t_hat) {h1 - 2*(h1 - h_theta)/(exp(s0*(t_hat -
theta)) + exp(s1*(t_hat - theta)))}
# 実測値を散布図 (X軸: t, Y軸: h) としてプロットする
> plot(t, h)

```



```
# 100組の年齢, 予測値データを赤色で折れ線グラフとして描く  
# Y軸の値は t_hat を引数とする h_hat 関数の出力値 (ベクトル)  
# 100組くらいのデータでの折れ線グラフでも滑らかな曲線に見える  
> lines(t_hat, h_hat(t_hat), col="red")
```



```

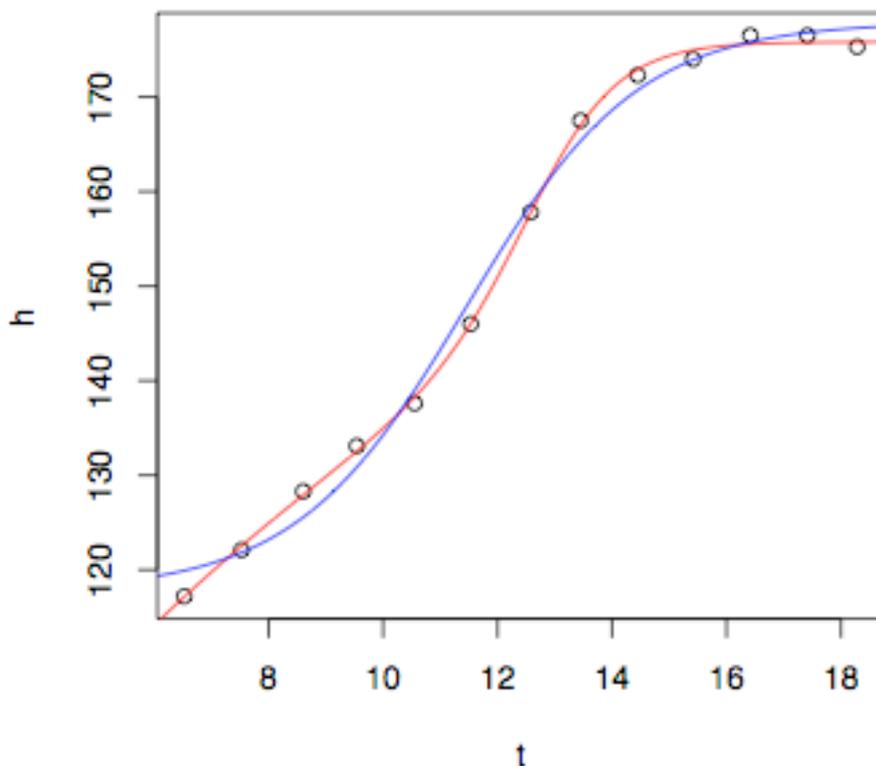
# こんどはロジスティック曲線をあてはめてみる
# 初期値はあらかじめ別の方法 (Excel ソルバー) で求めた値を使っている :-)
# Excel ソルバーでのこのような使い方は次の本が詳しい：
# 荻田・中西・上田：Excel でできる最適化の実践らくらく読本。同友館，2003
> result_logistic <- nls(h ~ a/(1 + b * exp(-c * t)) + d, data=dat,
start=list(a=60, b=2000, c=0.7, d=100))
> result_logistic
Nonlinear regression model
  model: h ~ a/(1 + b * exp(-c * t)) + d
 data:  dat
      a          b          c          d
60.1433612 2103.5814601  0.6678158 117.7635785
residual sum-of-squares: 41.04625
# 求まった係数を変数として利用できるようにする

```

```

> attach(as.list(coef(result_logistic)))
> a
[1] 60.14336
> d
[1] 117.7636
# ロジスティック曲線をあてはめた予測値を求める関数 (hh_hat) を定義する
# 引数にはさきほどの 100 個の年齢データ (h_hat) をつかう
> hh_hat <- function(t_hat) {a/(1 + b * exp(-c * t_hat)) + d}
# PBI の実測値, 予測値プロットに重ねて, ロジスティック曲線を青色で描く
> lines(t_hat, hh_hat(t_hat), col="blue")

```



```

# 当てはまりの良さを AIC で見ても, PBI モデルの方が良いことがわかる
> AIC(result_pb1)
[1] 32.26597
> AIC(result_logistic)

```

```
[1] 59.83915
```

```
# これまでにつくって保存した変数をリスト (ls) させてみると：
```

```
> ls()
```

```
[1] "amv"          "apv"          "dat"          "dmv"
[5] "dpv"          "h"            "h_amv"        "h_apv"
[9] "h_hat"        "hh_hat"       "mv"           "pv"
[13] "result_logistic" "result_pb1"   "sh"           "sl"
[17] "t"            "t_hat"
```

```
# Rを終了する
```

```
# この時に上の変数を保存するかどうか聞いてくる
```

```
# Yes とすると、次回にRを起動した時に再び使用できる
```

```
> q()
```

```
# 以下のファイルを PBI.R の名前で「/Users/hoge/Desktop/」に保存しておき
```

```
# Rの起動後、source("/Users/hoge/Desktop/PBI.R", echo=T)
```

```
# とすると、ここで述べた手順が画面に再現する（バッチ処理ができる）
```

```
# ファイルの先頭に sink("/Users/hoge/Desktop/pb.out")
```

```
# ファイルの最後に sink() を加えると
```

```
# 結果の出力先は sink 関数の引数で示すファイルに書かれる（画面には出ない）
```

```
# h1, theta などの求めたパラメータは、attach(as.list(coef(result_pb1)))で
```

```
# セッション中に参照できるようになる
```

```
# Solve Preece-Bains model 1 growth curve and logistic growth curve by R
```

```
#
```

```
t<-c(6.527,7.527,8.605,9.541,10.551,11.532,12.580,13.451,14.448,15.417,16.416,17.413,18.283)
```

```
t
```

```
h<-c(117.2,122.1,128.3,133.1,137.6,146.0,157.8,167.5,172.3,174.0,176.5,176.5,175.3)
```

```
h
```

```
dat<-data.frame(t,h)
```

```

dat
result_pb1<-nls(h ~
h1-2*(h1-h_theta)/(exp(s0*(t-theta))+exp(s1*(t-theta))),data=dat,start=list(h1=1
75,h_theta=162,s0=0.1,s1=1.2,theta=15))
result_pb1
attach(as.list(coef(result_pb1)))
h1
h_theta
s0
s1
theta
sl<-(s0+s1)/4-sqrt((s0+s1)^2/16-s0*s1/2)
sl
sh<-(s0+s1)/4+sqrt((s0+s1)^2/16-s0*s1/2)
sh
amv<-theta+(log(sl-s0)-log(s1-sl))/(s1-s0)
amv
apv<-theta+(log(sh-s0)-log(s1-sh))/(s1-s0)
apv
dmv<-h1-((2*(h1-h_theta))/((exp(s0*(amv-theta)))+(exp(s1*(amv-theta)))))
dmv
dpv<-h1-((2*(h1-h_theta))/((exp(s0*(apv-theta)))+(exp(s1*(apv-theta)))))
dpv
mv<-sl*(h1-dmv)
mv
pv<-sh*(h1-dpv)
pv
t_hat<-seq(6,19,length=100)
t_hat
h_hat<-function(t_hat){h1-2*(h1-h_theta)/(exp(s0*(t_hat-theta))+exp(s1*(t_hat-th
eta)))}
plot(t,h)

```

```
lines(t_hat,h_hat(t_hat),col="red")
result_logistic<-
nls(h~a/(1+b*exp(-c*t))+d,data=dat,start=list(a=60,b=2000,c=0.7,d=100))
result_logistic
attach(as.list(coef(result_logistic)))
a
b
c
d
hh_hat<-function(t_hat){a/(1+b*exp(-c*t_hat))+d}
lines(t_hat,hh_hat(t_hat),col="blue")
AIC(result_pb1)
AIC(result_logistic)
ls()
```