

# R で GLM をやってみよう

## ～glm 関数の利用と手計算での最尤推定～

玉木 一郎\*

2009/06/14

**概要:** 本テキストでは、まず一般化線形モデル (Generalized Linear Model: GLM) の概要について説明を行う。続いて、応答変数が二項分布とガンマ分布に従う場合の 2 つの例題を用いて、実際に glm 関数を用いたパラメータ推定、予測、信頼区間の推定を行う。さらに、同例題のデータを用いて直接尤度関数を記述し、optim 関数を用いた最尤法によるパラメータ推定を行うことで、GLM と最尤推定への理解を深めることを目的とした。

### 目次

1	はじめに: GLM とは?	1
2	ドブソン (2008) <sup>2)</sup> P. 144 の二項分布の例	2
2.1	データを眺める	2
2.2	統計モデリングと glm 関数によるパラメータ推定	2
2.3	予測値とその 95% 信頼区間の推定	2
2.4	パラメトリック・ブートストラップ	3
2.5	optim 関数による最尤推定	3
3	McCullagh & Nelder (1989) <sup>5)</sup> P. 300 のガンマ分布の例	4
3.1	データを眺める	4
3.2	統計モデリングと glm 関数によるパラメータ推定	4
3.3	予測値とその 95% 信頼区間の推定	4
3.4	パラメトリック・ブートストラップ	5
3.5	optim 関数による最尤推定	5
4	終わりに	5
4.1	他に勉強すると良いこと	6
4.2	参考書・サイト	6
4.3	R で統計をする際に	6

### 1 はじめに: GLM とは?

まずはじめに、このテキストでは、ふるまいに注目している変数のことを**応答変数**、これらを説明するために用いる変数を**説明変数**と呼び<sup>\*1</sup>、応答変数は何らかの確率分布に従うランダムな変数、説明変数はばらつきのない固定された値として扱います。

GLM は **Generalized Linear Model** の頭文字を並べたもので、日本語では**一般化線形モデル**と呼ばれるものです。普通の線形モデル (重回帰や分散分析) やロジスティック回帰、対数線形回帰などの**応答変数の従う確率分布が異なる線形モデル**を同一の枠組みの中で取り扱おうというものです<sup>\*2</sup>。近年 R が普及するにつれ、多くの

研究で用いられるようになりました。普通の林学の講義で取り扱われることはまず無いかと思いますが、論文や学会発表などで名前を聞いたことのあるという人も多いのではないかと思います。応答変数を  $y$ 、説明変数を  $x$  とすると、一般化線形モデルは以下の式で示すことができます<sup>\*3</sup>。

$$g[E(y_i)] = \eta_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_j x_{ij} + \dots + \beta_m x_{im}$$

$$y_i \sim f(y_i | \theta)$$

$E$  は期待値を、 $\beta$  は係数を、添字  $i$  はデータの番号を、 $j$  はパラメータの番号を示します。ここで、 $\eta$  を**線形予測子**、応答変数の期待値と線形予測子を結びつけている関数  $g$  を**連結関数**と呼びます。連結関数は恒等関数や対数関数、ロジット関数など様々な形をとることができます。どれを用いるかはモデリングの仮定やモデルのあてはまりを吟味した上で決めれば良いかと思います。特に問題がなければ、応答変数が従うと仮定した確率分布に対応する正準連結関数を使用するのが良いでしょう。～(チルダ)は左辺が右辺の確率分布に従うことを意味します。 $f$  は  $y$  が従う確率分布の**確率関数** (離散分布の場合) もしくは**確率密度関数** (連続分布の場合) を、 $\theta$  はそのパラメータを示します。一般化線形モデルでは、応答変数はその型に応じて、正規分布や二項分布、ポアソン分布などの指数型分布族に含まれる確率分布に従うと仮定します<sup>\*4</sup>。

一般化線形モデルでは最尤法でパラメータ推定を行います。先ほどの確率 (密度) 関数  $f$  をデータが与えられた時のパラメータ  $\theta$  の関数 ( $L(\theta | y_i)$ ) としてみると、 $n$

\* 名古屋大学大学院 生命農学研究科 森林生態生理学研究分野・日本学術振興会特別研究員

\*1 応答変数は他にも従属変数とか目的変数、説明変数は独立変数とか予測変数と呼ばれています。

\*2 名前の良く似た一般線形モデル (General linear model) は重回

帰と分散分析までを総合的にとらえたものです。

\*3 いわゆる誤差項は無いことに注意。

\*4 英語文献中に出てくる、いわゆる error distribution は誤差分布と訳されることもありますが、誤解をまねくので、このレジュメ中では**応答変数の従う確率分布**としました。

個のデータを観測した時の尤度関数  $L$  は以下の式で示すことができます。

$$L(\theta | y_1, \dots, y_i, \dots, y_n) = \prod_i^n L(\theta | y_i)$$

この尤度関数を最大にするようなパラメータ  $\theta$  を最尤推定量と呼びます。実際には  $L$  の対数をとったもの（対数尤度）を扱う方が便利な場合が多いです。

GLM を R で実行する上では、連結関数と応答変数の従う確率分布について、ある程度の知識があれば十分かもしれませんが、尤度関数の立て方のあたりまで知っておくと理解が深まって、応用も利くと思います。

## 2 ドブソン (2008) <sup>2)</sup>P. 144 の二項分布の例

### 2.1 データを眺める

この節では二硫化炭素ガスに 5 時間暴露されたカブトムシの死亡個体数とガス濃度との関係を GLM で解析してみます (表 1)。

表 1 カブトムシの死亡データ

ガス濃度 ( $\log_{10}$ CS <sub>2</sub> mg $l^{-1}$ )	実験に供した カブトムシの数	死亡数
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

どうやらガス濃度が高くなるとカブトムシが死にやすくなりそうです。ただし、死亡数には上限があります。ガス濃度を横軸に、死亡率（死亡個体数 / サンプル個体数）を縦軸にとり、図示してみましょう (図 1)。このような割り算データは分母にくる数のサイズが分かるように図示するのが誠実です (この例では処理間でサンプル個体数がほとんどばらついていないのであんまり意味ないのですが...)

```
> gas <- c(1.6907, 1.7242, 1.7552, 1.7842,
+         1.8113, 1.8369, 1.8610, 1.8839)
> n.beetle <- c(59, 60, 62, 56, 63, 59, 62, 60)
> dead.beetle <- c(6, 13, 18, 28, 52, 53, 61, 60)
> plot(gas, dead.beetle/n.beetle, cex = n.beetle/40)
```

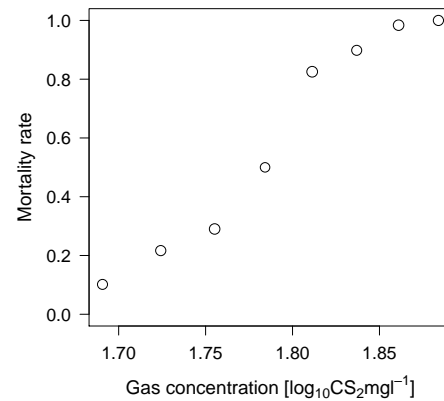


図 1 ガス濃度とカブトムシの死亡率（死亡個体数 / サンプル個体数）の関係。丸のサイズはサンプル個体数に比例する。

### 2.2 統計モデリングと glm 関数によるパラメータ推定

ここで、

- カブトムシの死亡個体数 ( $y$ ) が死亡率 ( $p = y/N$ ) とサンプル個体数 ( $N$ ) を上限値に持つ二項分布に従う。
- 死亡率 ( $p$ ) はガス濃度 ( $x$ ) が増加すると共に高まるが、0 から 1 の間の値をとる。
- 連結関数はロジット関数とする。

と仮定すると、以下の式で表現することができます。

$$\log\left(\frac{p_i}{1-p_i}\right) = \eta_i = \beta_0 + \beta_1 x_i$$

$$p_i = \frac{1}{1 + \exp(-\eta_i)}$$

$$y_i \sim \text{Binomial}(p_i, N_i)$$

これを R の glm 関数では以下のように指定します。family = binomial のデフォルトの連結関数はロジット関数になっています。詳しい指定方法については ?family を参照して下さい。

```
> beetle <- cbind(dead.beetle, n.beetle - dead.beetle)
> glm.beetle <- glm(beetle ~ gas, family = binomial)
> glm.beetle$coefficients
(Intercept)      gas
-60.71745     34.27033
```

### 2.3 予測値とその 95% 信頼区間の推定

次にモデルによる予測値を図に重ねます (図 2)。

```
> x <- seq(min(gas), max(gas), by = 0.01)
> eta.pred <- glm.beetle$coefficients["(Intercept)"] +
+           glm.beetle$coefficients["gas"]*x
> p.pred <- 1/(1 + exp(-eta.pred))
> lines(x, p.pred)
```

続けて予測値の 95% 信頼区間 (パラメータの 95% 信頼区間については後述) を計算し、図に重ねます。このと

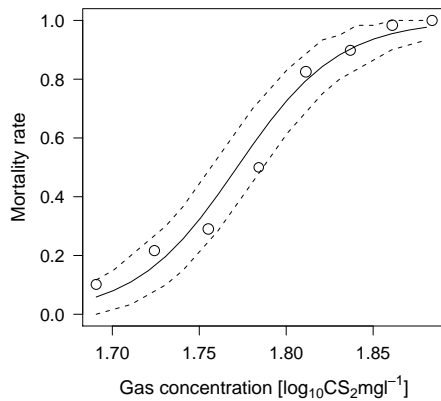


図2 ガス濃度とカブトムシの死亡率（死亡個体数/サンプル個体数）の関係。丸のサイズはサンプル個体数に比例する。実線は予測値を、破線は予測値の95%信頼区間を示す。

き、実際にはデータポイントごとに二項分布のサンプル数 ( $N$ ) が異なりますが、今回はそれらの調和平均を丸めた値を用いて、 $N$  を固定して信頼区間を推定しました。図を見ると信頼区間の幅は死亡率が0や1に近い時に狭く、0.5付近で最も広がっています（物差をあてて見ると分かりやすいです）。

```
> mean.N <- round(length(n.beetle)/sum(1/n.beetle))
> p.lower <- qbinom(0.025, mean.N, p.pred)/mean.N
> p.upper <- qbinom(0.975, mean.N, p.pred)/mean.N
> lines(x, p.lower, lty = 2)
> lines(x, p.upper, lty = 2)
```

## 2.4 パラメトリック・ブートストラップ

先ほど、予測値の95%信頼区間の計算方法を紹介したので、今度はパラメトリック・ブートストラップを用いた帰係数の信頼区間の推定方法を紹介します。この方法を簡単に説明すると、まず推定したパラメータを含むモデルを用いて乱数データを発生させ、その乱数データを応答変数にしてパラメータを再推定します。このシミュレーションを繰り返すことで、パラメータの信頼区間を推定します。

```
> n.sim <- 1000 # シミュレーション回数
> beta <- data.frame(beta0 = rep(NA, n.sim),
+                   beta1 = rep(NA, n.sim))
> eta <- glm.beetle$coefficients["(Intercept)"] +
+       glm.beetle$coefficients["gas"]*gas
> p <- 1/(1 + exp(-eta))
> N <- n.beetle
> sapply(1:n.sim,
+       function(i){
+         y.random <- rbinom(length(p), N, p)
+         beetle.random <- cbind(y.random, N - y.random)
+         beta[i, ] <- glm(beetle.random ~ gas,
+                          family = binomial)$coefficients
+       })
> beta.ci <- apply(beta, 2, quantile,
+                  prob = c(0.025, 0.975))
> beta.ci
      beta0      beta1
2.5% -72.61395 29.59513
97.5% -52.42950 40.95304
```

## 2.5 optim 関数による最尤推定

$n$  個のデータを観測した時の尤度は  $L = \prod_i^n f(y_i)$  という掛け算の形をとります。このままでは  $f(y)$  が極端に小さかったり大きかったりする場合にふれが大きくなるため、対数尤度  $LL = \log(L) = \sum_i^n \log(f(y_i))$  について考えます。 $LL$  は単純に  $L$  の対数をとっただけなので  $L$  が最大のとき、 $LL$  も最大値をとります。この  $LL$  を最大にするようなパラメータを求めたいわけです。実際に **optim** 関数は、与えられた関数を最小にするようなパラメータを推定するので、 $NLL = -LL$  について考えます。二項分布の確率関数は  $f(y) = {}_N C_y p^y (1-p)^{N-y}$  なので、**optim** に与える関数を以下のように記述します。

```
> f.beetle1 <- function(parameters){
+   beta0 <- parameters[1]
+   beta1 <- parameters[2]
+   eta <- beta0 + beta1*gas
+   p <- 1/(1 + exp(-eta))
+   y <- dead.beetle
+   N <- n.beetle
+   NLL <- -sum(log(choose(N, y)*p^y*(1-p)^(N-y)))
+ }
```

続いて適当な初期値を与えて **optim** を実行します。初期値は関数型を考えたり、グラフから適当に読み取るべきなのですが、今回はズルして **glm** の推定結果に近い値を入れておきました\*5。

```
> optim.beetle1 <- optim(c(-60, 30), fn = f.beetle1)
> optim.beetle1$par
[1] -60.73001 34.27737
```

最初に行った **glm.beetle\$coefficients** の値と見比べて見てください。ほぼ等しいことが分かります。今回は確率関数を直接記述しましたが、Rには各種確率分布の確率（密度）関数が最初から入っているので、信頼区間の計算をしたときと同様に、それらを利用することができます。

```
> f.beetle2 <- function(parameters){
+   beta0 <- parameters[1]
+   beta1 <- parameters[2]
+   eta <- beta0 + beta1*gas
+   p <- 1/(1 + exp(-eta))
+   y <- dead.beetle
+   N <- n.beetle
+   NLL <- -sum(dbinom(y, N, p, log = T))
+ }
> optim.beetle2 <- optim(c(-60, 30), fn = f.beetle2)
```

\*5 できるかぎりそれっぽい値になるよう努力して下さい。初期値によってうまくいかない場合があります。

### 3 McCullagh & Nelder (1989) <sup>5)</sup>P. 300 のガンマ分布の例

#### 3.1 データを眺める

この節では血液の凝固時間 (秒) と血漿濃度 (%) の関係を GLM で解析してみます (表 2)。このデータは `glm` 関数のヘルプに入っているものですが、今回はそのうちの `lot1` だけ使います (そのため変数名を `time` に変更しました)。

表 2 血液の凝固時間 (秒) と血漿濃度 (%) の関係

血漿濃度 (%)	凝固時間 (秒)
5	118
10	58
15	42
20	35
30	27
40	25
60	21
80	19
100	18

血漿濃度が高くなるにつれて凝固時間は短くなるようです。横軸に血漿濃度の対数値<sup>\*6</sup>、縦軸に凝固時間をとって図示してみます (図 3)。

```
> u <- c(5, 10, 15, 20, 30, 40, 60, 80, 100)
> time <- c(118, 58, 42, 35, 27, 25, 21, 19, 18)
> plot(log(u), time, ylim = c(20, 140))
```

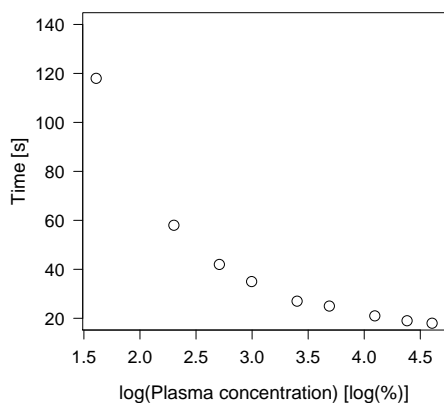


図 3 血漿濃度 (%) の対数値と血液の凝固時間の関係

<sup>\*6</sup> いきなり対数をとっていますが、とらない場合も後で試してみてください。尤度がずいぶん低くなります。

#### 3.2 統計モデリングと `glm` 関数によるパラメータ推定

ここで、

- 凝固時間 ( $y$ ) は正の値を示す連続データなので、ガンマ分布に従う。
- 血漿濃度の対数値 ( $x$ ) に対する凝固時間の減少具合が双曲線的なので、連結関数には逆数関数を用いる。
- `shape` パラメータ ( $\alpha$ ) は常に一定。
- `scale` パラメータ ( $s$ ) が線形予測子の変化に応じて変化する<sup>\*7</sup>。
- ガンマ分布の平均は  $\alpha s$ 、分散は  $\alpha s^2$ 。

と仮定すると、以下の式で表現することができます。

$$\eta_i = \beta_0 + \beta_1 x_i$$

$$E(y_i) = \alpha s_i = 1/\eta_i$$

$$y_i \sim \text{Gamma}(\alpha, s_i)$$

これを R の `glm` 関数では以下のように指定します。`family = Gamma` のデフォルトの連結関数は逆数関数になっています。

```
> glm.time <- glm(time ~ log(u), family = Gamma)
> glm.time$coefficients
(Intercept)      log(u)
-0.01655438    0.01534311
```

`log(u)` の回帰係数が正の値になっていますが、これは連結関数が逆数関数のためです。この場合のように、回帰係数だけ見て「説明変数は応答変数に対して正の影響を及ぼしている」と判断しないで下さい。

#### 3.3 予測値とその 95% 信頼区間の推定

次にモデルによる予測値とその 95% 信頼区間を計算しますが、信頼区間を計算する際に、`shape` パラメータの最尤推定値が必要になります。そこで、まず `shape` パラメータを計算しておきます。ガンマ分布の GLM では `shape` パラメータは `dispersion` パラメータの逆数になります。`1/summary(glm.time)$dispersion` で計算できるのですが、`glm` 関数では繰り返し重み付け最小自乗法でパラメータ推定を行っているため、この `shape` パラメータの推定値は最尤推定値と異なる場合があります<sup>\*8</sup>。というわけで、以下のようにして `shape` パラメータ (a) の最尤推定値を計算します。

<sup>\*7</sup> `scale` パラメータの代わりに `rate` パラメータ ( $1/s$ ) を指定することもできます。ガンマ分布を扱う際には、混乱を避けるために `scale` なのか `rate` なのか、どちらの話をしているのか明示しておく必要があります。

<sup>\*8</sup> といってもどちらの `shape` パラメータでも信頼区間の幅はそんなに変わらないのですが...。ちなみに回帰係数の推定値は最尤推定値とほとんど変わりません。

```
> 1/summary(glm.time)$dispersion
[1] 408.8208
> library("MASS")
> a <- gamma.shape(glm.time)$alpha # shape パラメータ
> a
[1] 538.1315
```

この値を用いて予測値とその 95% 信頼区間を計算し、図示します (図 4)。

```
> x <- seq(1.6, 4.6, by = 0.1)
> eta.pred <- glm.time$coefficients["(Intercept)"] +
+   glm.time$coefficients["log(u)"]*x
> time.pred <- 1/eta.pred
> s.pred <- 1/eta.pred/a # scale パラメータ
> time.lower <- qgamma(0.025, shape = a, scale = s.pred)
> time.upper <- qgamma(0.975, shape = a, scale = s.pred)
> lines(x, time.pred)
> lines(x, time.lower, lty = 2)
> lines(x, time.upper, lty = 2)
```

これまた少し分かりにくいので、物差しをあてて見てほし

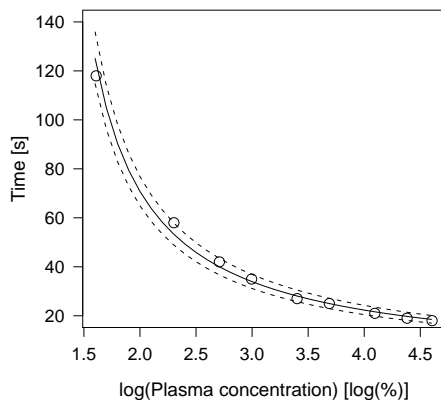


図 4 血漿濃度 (%) の対数値と血液の凝固時間の関係。実線はモデルの予測値を、破線は予測値の 95% 信頼区間を示す。

いのですが、x 軸の値が小さくなると (scale パラメータが大きくなる) と信頼区間の幅が大きくなるのが分かります。

### 3.4 パラメトリック・ブートストラップ

先ほどの二項分布の例と同様にパラメータの信頼区間を推定します。

```
> n.sim <- 1000
> beta <- data.frame(beta0 = rep(NA, n.sim),
+   beta1 = rep(NA, n.sim))
> eta <- glm.time$coefficients["(Intercept)"] +
+   glm.time$coefficients["log(u)"]*log(u)
> s <- 1/eta/a # scale
> sapply(1:n.sim,
+   function(i){
+     random.y <- rgamma(length(s),
+       shape = a, scale = s)
+     beta[i, ] <- glm(random.y ~ log(u),
+       family = Gamma)$coefficients
+   })
> beta.ci <- apply(beta, 2, quantile,
+   prob = c(0.025, 0.975))
> beta.ci
      beta0      beta1
2.5% -0.01809224 0.01460593
97.5% -0.01483667 0.01601967
```

### 3.5 optim 関数による最尤推定

同様に optim 関数で最尤推定します\*9。

```
> f.time <- function(parameters){
+   beta0 <- parameters[1]
+   beta1 <- parameters[2]
+   a <- parameters[3] # shape パラメータ
+   eta <- beta0 + beta1*log(u)
+   s <- 1/eta/a # scale パラメータ
+   NLL <- -sum(dgamma(time, shape = a,
+     scale = s, log = T))
+ }
> optim.time <- optim(c(-0.016, 0.015, 530), fn = f.time)
> optim.time$par # beta0, beta1, shape パラメータ
[1] -0.01655432 0.01534309 538.14832592
```

## 4 終わりに

今回は R の統計機能の紹介を担当してくれとのこと、R と言えば GLM かなと思ひ、自分が GLM の存在を知った頃に知りたかったこと、調べるのが困難だったことを中心にまとめました\*10。それでもけっこうな量になってしまいました…。今回述べてはいないけど知っておいて欲しいことは他にもあります\*11。キーワードを後で述べておくので、自分で調べてみて下さい。

最近では GLM に関する書籍や情報も増えてきましたが、これ一つだけを見ておけば大丈夫というものには特に無いように思われます。自分がこれまで触れてきた中で良いと思うものをあげておくので参照してみてください。いろいろな本を読んで、**R で実験して**自身の知識を高めて欲しいと思います。

また、情報源によって表現が異なったり\*12、勘違いしているものも時にはあるので注意が必要です。本や論文中の数式も間違っていることも時にはあります。自分も極力それを無くすように記述してきましたが、文章やコードの誤り、不適切な表現等ありましたらご連絡下さい\*13。

\*9 警告が出ますが、ガンマ分布の shape パラメータや scale パラメータは正の値しかとらないので、パラメータ値の探索の過程でそれらが負になると警告が出ます。しかし、optim は気を取り直して計算を続けてくれるので気にしなくて良いです。method = "L-BFGS-B"にすると、パラメータの探索範囲を狭めることができますが、今回の例題ではデフォルトの"Nelder-Mead"の方がうまくパラメータを推定できるようです。気になる人は試してみてください。

\*10 GLM を使ってみたくて使い始めた R ですが、実は自分の研究では glm 関数を使う機会がありませんでした…。

\*11 多重検定とか多重共線性、検定 vs モデル選択、頻度主義者 vs ベイズアンミたいに、面倒いし宗教的だったりして、自分ではあんまり触れたくないこともあります…。

\*12 単に deviance と書いてあっても、-2LL のことを指さない場合もあります。

\*13 連絡先 (garageit(at)gmail.com)。あ、2 章と 3 章で変数名ががぶっているのが許して下さい…。

#### 4.1 他に勉強すると良いこと

適当につらつらキーワードを並べておきます。

- いろいろな確率分布の特徴（連続 or 離散，上限・下限，平均と分散の関係，関数型等々）。
- 回帰係数の検定。
- 情報量基準とモデル選択。
- 多重共線性。
- 引数 `offset` の使い道。
- 過分散（overdispersion）への対処。
- 一般化線形混合モデル（Generalized Linear Mixed Model: GLMM）やランダム効果。
- ベイズ統計学。
- マルコフ連鎖モンテカルロ（Markov Chain Monte Carlo: MCMC）法。
- 統計学の歴史的経緯。
- ノンパラメトリック・ブートストラップ。

#### 4.2 参考書・サイト

個人的に役に立つと思う本とウェブサイトをあげておきます。

- 本
  - Ecological models and data (Bolker 2008)<sup>1)</sup>: とにかく最尤推定一筋な本です。すぐに役立つ知識は載っていませんが，自身の技術を底上げすることができます。
  - 一般化線形モデル入門 (ドブソン 2008)<sup>2)</sup>: 長らく絶版で入手困難でしたが，最近第 2 版が翻訳されました。あまり系統立てて書かれていない気がしますが，分かりやすいです。
  - 計算統計 II (伊庭ら 2005)<sup>3)</sup>: GLM とは直接関係ありませんが，良い統計の本だと思います。
  - 統計学を拓いた異才たち (サルツブルグ 2006)<sup>7)</sup>: 統計の歴史を楽しく読めます。
  - 統計モデル入門 (丹後 2000)<sup>8)</sup>: 知りたいことはだいたい書いてある良い本だと思いますが，難解です。
  - S-PLUS による統計解析 (ヴェナブル・リプリー 2001)<sup>9)</sup>: S-PLUS のコマンドが書いてありますが，R でも使えるので役に立ちます。難解です。
  - 生物統計学入門 (山田・北田 2004)<sup>10)</sup>: 確率分布の話がきちんと書いてある良い入門書だと思います。
- ウェブサイト
  - KuboWeb<sup>4)</sup>: 生態学者なら（でなくても）はずせ

ません。統計学授業は一読しておくの良いと思います。ここで `offset` 技も習得して下さい。

- RjpWiki<sup>6)</sup>: 一般的な関数について分からないことは，まずはここで検索しましょう。

#### 4.3 R で統計をする際に

- なるべく新しいバージョンの R を使しましょう（バグが修正されているので）。昔インストールした古いのを使い続けられないこと。
- 検定とかする前に，必ず図を描いて，視覚的に確かめること（河崎君担当箇所参照）。
- 関数の使い方が分からない時，初めて使う時は必ず関数の `help` を参照しましょう。
- まともなテキストエディタを使いましょう（安部君担当箇所参照）。

#### 引用文献・サイト

- 1) Bolker BM, 2008. Ecological models and data in R. Princeton University Press.
- 2) ドブソン AJ, 2008. 一般化線形モデル入門, 原著 第 2 版 (Dobson AJ, 2002. An introduction to generalized linear models-2nd edition. Chapman & Hall. 田中豊・森川敏彦・山中竹春・富田誠 訳). 共立出版.
- 3) 伊庭幸人・種村正美・大森裕浩・和合肇・佐藤整尚・高橋明彦, 2005. 計算統計 II マルコフ連鎖モンテカルロ法とその周辺. 岩波書店.
- 4) KuboWeb. <http://hosho.ees.hokudai.ac.jp/~kubo/index-j.html>
- 5) McCullagh P & Nelder JA, 1989. Generalized linear models-2nd edition. Chapman & Hall.
- 6) RjpWiki. <http://www.okada.jp.org/RWiki/>
- 7) サルツブルグ D, 2006. 統計学を拓いた異才たち (Salsburg D, 2001. The lady tasting tea. How statistics revolutionized science in the twentieth century. Henry Holt & Company. 竹内恵行・熊谷悦生 訳) 日本経済新聞社.
- 8) 丹後俊郎, 2000. 統計モデル入門. 朝倉書店.
- 9) ヴェナブルズ WN・リプリー BD, 2001. S-PLUS による統計解析, 原著 第 3 版 (Venables WN & Ripley BD, 1999 Modern applied statistics with S-PLUS-3rd edition. Springer-Verlag. 伊藤幹夫・大津泰介・戸瀬信之・中東雅樹 訳). シュプリンガー・フェアクラーク 東京.
- 10) 山田作太郎・北田修一, 2004. 生物統計学入門. 成山堂書店.