HOKKAIDO UNIVERSITY

Title	講義のーと ： データ解析のための統計モデリング
Author(s)	久保, 拓弥
Description	この講義資料は，著者のホームページhttp://hosho.ees.hokudai.ac.jp/~kubo/ce/EesLecture2008.htmlからダウンロードできます。 http://hosho.ees.hokudai.ac.jp/~kubo/ce/EesLecture2008.html
Issue Date	2008
Doc URL	https://hdl.handle.net/2115/49477
Type	learning object
File Information	kubostat2008c.pdf, 第３回

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

データ解析のための統計モデリング (2008 年 10-11 月)

全 5 (+2) 回中の第 3 回 (2008–11–06)

一般化線形モデル (GLM) 1

久保拓弥 kubo@ees.hokudai.ac.jp

http://hosho.ees.hokudai.ac.jp/~kubo/ce/EesLecture2008.html

この講義のーとが「データ解析のための統計モデリング入門」として出版されました!

http://hosho.ees.hokudai.ac.jp/~kubo/ce/IwanamiBook.html

まちがいを修正し，詳しい解説・新しい内容をたくさん追加したものです

今日のもくじ

1. 一般化線形モデル (GLM) って何なの? 1
2. GLM の部品: 確率分布，link 関数，線形予測子 4
3. GLM の例題: 架空植物のデータ解析 5
4. 統計モデリング: glm(..., family = poisson) 9
5. 因子型の説明変数の場合 14
6. 説明変数が数量型 + 因子型のモデル 16
7. Deviance って何なの? 17
8. 「よい」モデルを選ぶモデル選択 21
9. 今日のまとめ 22

「生態学の統計モデリング」第 3 回目です．前回は統計モデルの基本部品である確率分布，とくに**ポアソン分布**について説明しました．そしてポアソン分布のパラメーターは**最尤推定法**によって推定できる，といったことを説明しました．前回の例は簡単だったので，ポアソン分布の平均をあらわす λ は簡単に推定でき，さらに「最尤推定値は標本平均値に等しい」といったことが簡単に示しました．

今日はもう少し複雑な統計モデル，たとえば確率分布の平均が λ だとすると，

$$\lambda = \big(\beta_1 + \beta_2 \times (\text{要因 } 1) + \beta_3 \times (\text{要因 } 2) + \cdots\big) \text{の関数}$$

といったモデルを考えて，そのパラメーター $\{\beta_j\} = \{\beta_1, \beta_2, \cdots\}$ を推定す

る，といった問題をあつかいます．これは**一般化線形モデル**の推定問題として解決できる場合があります．

1. 一般化線形モデル (GLM) って何なの?

具体的な統計モデリングに入る前に，一般化線形モデルをめぐる歴史的な経緯みたいなものを少し見てみましょう．以前にも説明したように，データ解析に使われる統計モデルの範囲は図 1 の内側 → 外側にむけて拡大してきました． [1]

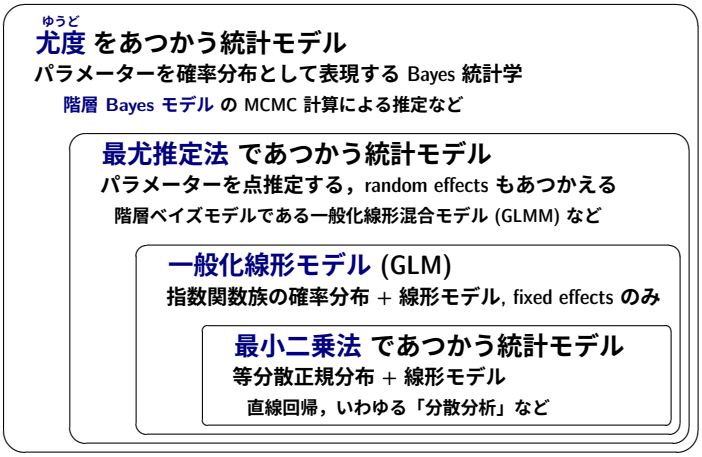


図 1: 統計モデルの「地図」

まず一番内側,「**等分散正規分布の統計モデル**しか考えないからパラメーター推定は何でも最小二乗法でよい」の世界で，(まぎらわしいのですが) **一般線形モデル** (general linear model) なるものが提唱されました．これはひらたく言ってしまえば「統計学といえば**何でも正規分布**の皆さん，あなたたちはデータの種類によって ANOVA とか重回帰とか ANCOVA とか使いわけてるみたいですけど，それって統計モデルの見地からすれば

- ばらつきをあらわす確率分布が正規分布
- 確率分布の平均 = 線形予測子 [2]
- 確率分布の分散は一定

という同じ統計モデルを何だか別ものあつかいしちゃってるのでは? これらは general linear model として統一的に理解して，アタマの中をすっきり整

1. 統計モデルの場合，考えつかれた時期からずいぶんと時間が経過してからふつーのデータ解析者たちに使われるようになったりします．新しい統計モデル普及の重要な理由のひとつは個人用計算機がここ 20 年ぐらいでずいぶんと高速化した，ということがあると思います．

2. あとで説明します．

理しましょうよ」といった趣旨なのだろうと思います．しかし一般線形モデルなる用語はあまり使われていません．というのも，一般線形モデルもその中に含んでしまう，より強力な統計モデルが普及していったからでしょう．

ここで一般線形モデルの例として，直線回帰 (直線のあてはめ) の概念を図 2 に示してみます． [3]

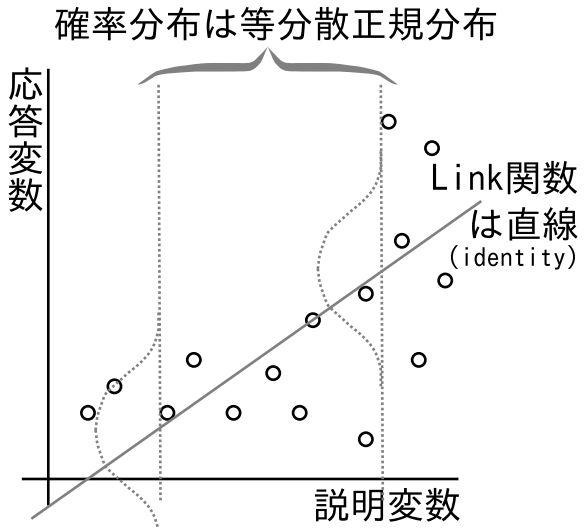


図 2: 古典的な直線あてはめ (一般線形モデル)

この講義では,「データを図にして，ばらつきをよく見て，統計モデルの部品である確率分布を選んで……」といった統計モデリングのお作法をおススめしております．ということで，それが実現できそうな統計モデリングとして，とりあえずてはじめに一般化線形モデル (図 3) から始めましょう，というのが今日のハナシです．

一般化線形モデル (generalized linear model; GLM) は「正規分布だけ」の一般線形モデルよりさらに強化 [4] された統計モデルです:

- 確率分布は正規分布だけでなく，ポアソン分布・二項分布・ガンマ分布などが選択できる [5]
- 確率分布の平均は線形予測子だけでなく link 関数にも依存する
- 確率分布の分散は使っている確率分布にあわせて変化しうる (例: ポアソン分布なら分散は平均とともに増大)

この GLM の中で今回はポアソン分布をつかった統計モデルの例，次回は二項分布をつかった統計モデルの例を説明します．どちらもカウントデータ [6] の統計モデリングに使います．

3. この図 2 で「直線回帰でダメな場合もけっこうあるんでは?」と言いたかったポイントとしては……たとえば応答変数が 0, 1, 2, 3, ... と数えられるカウントデータだったとしましょう．その場合，(1) 期待値 (直線) がマイナスになるってヘンじゃない? (2) 図でみると「ばらつき一定」ではなさそう? といったところでしょう．で，やっぱり「分布をよく見て」確率分布を選んで統計モデリングしましょう，というハナシにつなげたいわけです．

4. いわば現実のデータ解析にあわせた，といいますか

5. これらは指数関数族の確率分布です．

6. これは「何でも正規分布」世界ではうまくあつかえません．

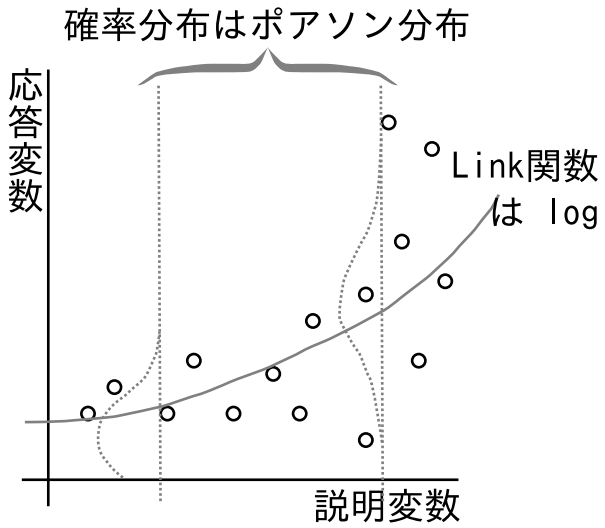


図 3: 一般化線形モデルによるあてはめの例

一般化線形モデルという名前は，こういうモデリングを普及したかったヒトたちが「ポアソン分布や二項分布をつかった統計モデリングも，皆さんよくご存知の線形モデルをちょっと拡張したものにすぎませんよ」といった意図もこめてつけたのかもしれません．いっぽうで (とくに正規分布に固執してるわけではない) 統計学的手法ユーザーからすると，(線形の) **ポアソン回帰** [7] や **logistic 回帰** [8] や正規分布の統計モデルをタバねたものを「一般化線形モデル」と呼んでいる，という理解でもよいかと思います． [9]

7. ポアソン分布をつかった統計モデルによるパラメーター推定

8. 二項分布 + logistic モデルをつかった統計モデルによるパラメーター推定

9. なぜ「線形」モデルなのか? 線形にしとくとわかりやすくて推定計算もやりやすい，といったご利益があります.

2. GLM の部品: 確率分布，link 関数，線形予測子

GLM を構成する部品は**確率分布** (probablistic distribution)・**link 関数** (link function)・**線形予測子** (linear predictor)・です．GLM で使える，正確にいえば R の `glm()` 関数で使える確率分布は前回の講義のーとでも示しましたが，下のとおりです.

	確率分布	乱数生成	パラメーター推定
(離散)	ベルヌーイ分布	`rbinom()`	`glm(family = binomial)`
	二項分布	`rbinom()`	`glm(family = binomial)`
	ポアソン分布	`rpois()`	`glm(family = poisson)`
	負の二項分布	`rnbinom()`	`glm.nb()`
(連続)	ガンマ分布	`rgamma()`	`glm(family = gamma)`
	正規分布	`rnorm()`	`glm(family = gaussian)`

link 関数 (link function) について説明する前に，まず**線形予測子**について簡単に説明してみます．最小二乗法でパラメーター推定する (正規分布でばらつきを説明する) 統計モデルでは平均 μ が

$$\mu = \beta_1 + \beta_2 \times (\text{要因 } 1) + \beta_3 \times (\text{要因 } 2) + \cdots$$

となっている必要があり，パラメーター $\{\beta_j\}$ を最小二乗法で推定します．要因というのは，より具体的には**説明変数** (explanatory variable) の値です．見ればわかるように右辺が

$$\sum_j \beta_j \times (1 \text{ もしくは説明変数の値})$$

というふうに推定したいパラメーターの線形結合になっているので，この統計モデルは**線形モデル**と名づけられています．また

$$z = \sum_j \beta_j \times (1 \text{ もしくは説明変数の値})$$

と定義すると，z は**線形予測子**とよばれたりします．GLM とはモデル内のどこかにこの線形予測子をもつ統計モデルのことです．[10]

なお，説明変数に対して，統計モデルで「説明される」ほうは応答変数 (response variable) とよばれます．[11]

一般化線形モデルでは単純な $\mu = z$ の場合[12] だけでなく，たとえば **log link 関数**を適用すると，平均を $\log \mu = z$ と定義してもさしつかえない，つまり

$$\mu = \exp\left(\beta_1 + \beta_2 \times (\text{要因 } 1) + \beta_3 \times (\text{要因 } 2) + \cdots\right)$$

といった統計モデルのパラメーター推定も可能になる，ということです．パラメーターの推定値は，前回の講義で説明した最尤推定法 (数値的な最尤推定法) によって計算されます．

GLM では「この確率分布なら，この link 関数を使うことが多い」[13] という組みあわせがあります．また確率分布 (R の `glm()` 関数の `family` 指定) と平均値などがきまると，そのばらつき (分散) もきまります．[14]

	確率分布	`glm()` の `family`	よく使う `link` 関数	分散 (m は平均)
(離散)	ベルヌーイ分布	`binomial`	`logit`	$\mu(1-\mu)$
	二項分布	`binomial`	`logit`	$\mu(1-\mu)$ (注)
	ポアソン分布	`poisson`	`log`	μ
(連続)	ガンマ分布	`gamma`	`log` かな?	μ^2
	正規分布	`gaussian`	`identity`	一定

10. ただし，好きなように確率分布や link 関数を選べない，という制約はありますけど.

11. 独立変数･従属変数といういいかたはちょっとまぎらわしいので，最近は説明変数・応答変数とよばれることが多いです.

12. これは R の `glm()` の identity link 関数つまり「そのまま」 link 関数に該当します.

13. canonical link function などと呼ばれていたり．この canonical 以外の link 関数が使える場合もあります.

14. 表中の二項分布の分散の (注): 正確には，$Np(1-p)$，ただし N は最大生起数， p は生起確率.

3. GLM の例題: 架空植物のデータ解析

さて，ここから図 4 に示されてる架空植物の種子数データの統計モデリングをとおして GLM を使ったデータ解析について考えてみましょう．この節では，統計モデリングに着手する前の準備として，この例題であつかう架空データについてまず説明してみます．ここではこの架空植物の個体ごとの種

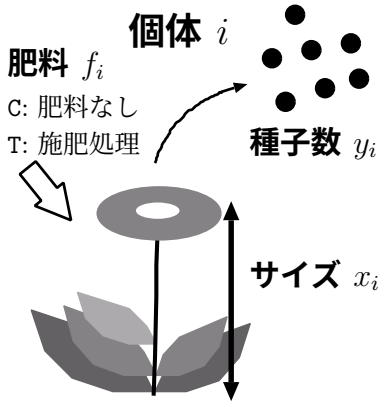


図 4: 架空植物の第 i 番目の個体 ($i = 1, 2, \cdots, 100$)

子数がどのように決まるか (統計モデルでどう表現するのがよいのか) をあつかいたいとします．個体は i という記号であらわされ ($i = 1, 2, 3, \cdots, 100$, つまり 100 個体います)，その種子数は y_i; 「サイズ」[15] は x_i; また全個体のうち 50 個体 ($i = 1, 2, \cdots, 50$) は特に何もしてないけど (C 個体)，残り 50 個体 ($i = 51, 52, \cdots, 100$) には施肥処理と称して肥料をあげた (T 個体)，とします．

この架空植物の観測データが何か spread sheet ソフトウェア[16] に格納されていたとして，これを「別名で保存」とかで CSV (comma separated values) 形式で保存したとします．[17]

もしそのファイル名が data3a.csv [18] だとすると，R では [19]

```
> d <- read.csv("data3a.csv")
```

と命じるだけでファイルを読みこんで data.frame というデータ構造に格納してくれます．ここでは d という名前が付けられています．[20] data.frame はとりあえず「table のようにあつかえるデータ構造」と考えてください．

R 内ではどのようにデータが格納されているのでしょうか? それを確認するためには，d とだけ指示して ENTER を押すと全データ表示されます．

15. どうせ架空植物なので具体的には何も考えてなくて……「高さ」でも「重量」でも何でもいいです．

16. たとえば，皆さんが好きで (そして私が好きではない)「ゑくせる」など．

17. CSV はテキストファイルになっているので，多くのソフトウェアでも読みこむことができます．

18. これは講義 web page からダウンロードできます．

19. このデータ読みこみ操作をする前に，R メニューの「ディレクトリの変更...」または R 関数の setwd() を使って data3a.csv が置いてあるディレクトリに移動する必要があります．

20. 入力がラクだから，という理由のためです．

```
    y     x  f
1   6  8.31  C
2   6  9.44  C
3   6  9.50  C
... (中略) ...
99  7 10.86  T
100 9  9.97  T
```

100 個体ぶんのデータが表示されます．[21] これを見ればわかるように，全 100 個体ぶんのデータが 100 行の data.frame として格納されていて，さらに最初の列 y には種子数，x にはサイズ，f には施肥処理の値が入っています．

21. head(d) とすると最初の 6 行だけが表示，head(d, 10) とすると最初の 10 行だけが表示されます．

まだあまり R にも慣れていないでしょうから，この d の列ごとにデータを表示させてみましょう．x と y 列は

```
> d$x
  [1]  8.31  9.44  9.50  9.07 10.16  8.32 10.61 10.06
  [9]  9.93 10.43 10.36 10.15 10.92  8.85  9.42 11.11
... (中略) ...
 [97]  8.52 10.24 10.86  9.97

> d$y
  [1]  6  6  6 12 10  4  9  9  9 11  6 10  6 10 11  8
 [17]  3  8  5  5  4 11  5 10  6  6  7  9  3 10  2  9
... (中略) ...
 [97]  6  8  7  9
```

まあ，こんなものかというかんじですが，f 列はちょっと様子がちがっていて，

```
> d$f
  [1] C C C C C C C C C C C C C C C C C C C C C C C C C
 [26] C C C C C C C C C C C C C C C C C C C C C C C C C
 [51] T T T T T T T T T T T T T T T T T T T T T T T T T
 [76] T T T T T T T T T T T T T T T T T T T T T T T T T
Levels: C T
```

これは **因子 (factor) 型** [22] という形式でデータが格納されているためです．R の read.csv() 関数は CSV 形式の table に C だの T だのといった文字を含む列 [23] は factor に変換します．つまりこの f 列は C と T の 2 水準 (levels) からなる (文字データではなく) **値** で構成されている，と上の表示は示しています．値は小さい順に C, T となっていて，これは read.csv() 関数が「水準の順番はアルファベット順でいいだろ」と判断したためです．[24]

22. 正確には「型」ではなく「クラス」とよぶべきなんでしょうが……

23. 文字・数字ごっちゃになってる列も factor になります．

24. もちろんユーザーが指示してこの水準の順番を変更できます．

R の class() 関数を使うと，あるデータオブジェクトがどういう型 (正確

にはクラス) に属しているか調べられます. [25]

25. R ではコメントマーク # から行末までは読みとばされます. ねんのため.

```
> class(d) # d そのものは data.frame であり, ……
[1] "data.frame"
> class(d$y) # y 列は整数だけの integer 型
[1] "integer"
> class(d$x) # x 列は実数も含むので numeric 型
[1] "numeric"
> class(d$f) # そして f 列は factor 型
[1] "factor"
```

さて, R の summary() 関数を使ってこの d と名づけられた data.frame の概要を把握してみましょう.

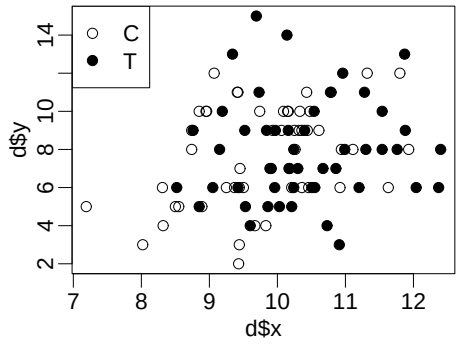
```
> summary(d)
       y                x           f
 Min.   : 2.00   Min.   : 7.190   C:50
 1st Qu.: 6.00   1st Qu.: 9.428   T:50
 Median : 8.00   Median :10.155
 Mean   : 7.83   Mean   :10.089
 3rd Qu.:10.00   3rd Qu.:10.685
 Max.   :15.00   Max.   :12.400
```

data.frame の summary はこのように列ごとに表示されます. [26] どのように "summary" されてしまうか, それは列の型に依存しています.

26. f 列の表示は, C が 50 個, T が 50 個ありますよ, と示しています. factor な vector を summary() するとカウントしてくれるんですね!

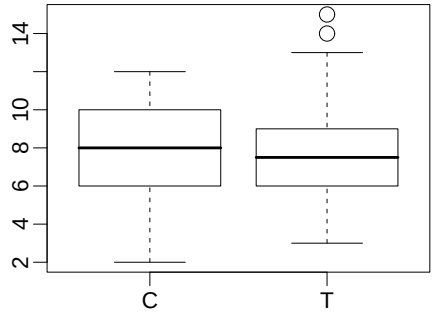
やはりデータ全体を「見る」には plot() 関数なんかで図にしてみるのが一番わかりやすいですね. これは横軸に x 列, タテ軸に y 列をとった散布図 (scatter plot) です.

```
> plot(d$x, d$y, pch = c(21, 19)[d$f])
> legend("topleft", legend = c("C", "T"), pch = c(21, 19))
```



横軸を因子型の f 列とした場合の表示はこうなります.

```
> plot(d$f, d$y)
```



R の plot() は横軸が因子型の場合は自動的に箱ひげ図 [27] (box-whisker plot) にするんですよね.

ちょっとハナシが先ばしりますが，この架空植物のデータでは上の散布図や箱ヒゲ図などで示されているように，**施肥効果があまり影響してないらしい**ということをちょっと覚えていてください.

データを R に読みこんだら，統計モデリングだの何だのやる前に plot() や summary() などなどを使って，データの概要を把握するようにしましょう．これはデータ解析においてたいへん重要なことです. [28]

27. ハコの上中下はそれぞれ 75%, 50%, 25% 点，上下のヒゲの範囲が (**おおよその**) 上下 2.5% 点，マルはその (おおよその) 95% 区間からはみだしたデータ, をあらわしています.

28. またこのときに，データ列どうしの割り算値などでっちあげて作図するのはあまりおススメできません.

4. 統計モデリング: glm(..., family = poisson)

さて，統計モデリングです．ここでやりたいことは d の y 列にある個体ごとの種子数がサイズ x や施肥処理 f にどう影響されているのか (あるいはされていないのか)，種子数のばらつきも含めて統計モデルのひとつである**一般化線形モデル**として表現したい，ということです.

個体 i の種子数 y_i は

- 0 個, 1 個, 2 個と数えられるカウントデータ
- 値に上限があるのか (このデータからは) はっきりしない

ということで，前回紹介したポアソン分布を部品とする統計モデルが作れそうです．ある個体 i の**種子数の平均**が

$$\lambda_i = \exp(\beta_1 + \beta_2 x_i)$$

このように**サイズ** x_i を使ってかける，と**仮定**します．先ほどの図をみると施肥効果 f_i はあまり種数に影響なさそうなので，ここでは**無視**します. [29]

29. あとから f_i をいれたモデルなども登場します.

上の平均種子数は

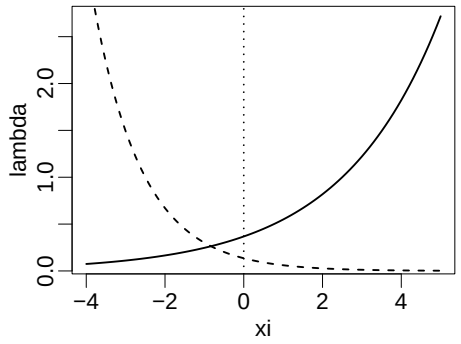
$$\log \lambda_i = \beta_1 + \beta_2 x_i$$

というふうにも書けますから，**link 関数** は log link 関数で**線形予測子** は $z = \beta_1 + \beta_2 x_i$ となっている，とわかると思います.

なぜ log link 関数をここで持ち出すのでしょうか? それはそうすると「計算に都合よく」かつ「わかりやすい」からです．ポアソン分布の平均パラメーター λ_i はつねに $\lambda_i > 0$ でなければなりません．$\lambda_i = \exp(z_i)$ というふうに線形予測子 z $(= \beta_1 + \beta_2 x_i)$ を $\exp()$ で囲っておけば z がどんな値をとろうとも，λ_i はつねに非負の値となります． [30]

30. 下の例は λ_i をてきとうに計算させている例で，β_1, β_2, x_i がどんな値をとろうとも，$\lambda_i \geq 0$ である，と示しています.

```
> xi <- seq(-4, 5, 0.1)
> plot(xi, exp(-1 + 0.4 * xi), type = "l", ylab = "lambda", lwd = 2)
> lines(xi, exp(-2 - 0.8 * xi), lwd = 2, lty = 2) # 破線
> abline(v = 0, lwd = 2, lty = 3)
```



また「わかりやすい」というのは要因の影響が積で表される，ということです．これはあとで施肥処理の効果もいれたモデルを検討するときに説明します.

確率分布の平均を $\lambda_i = \exp(\beta_1 + \beta_2 x_i)$ にしよう，ときめたのである個体 i で種子数が y_i である確率はポアソン分布の確率密度関数を使って

$$p(y_i \mid \{\beta_j\}, \{x_i\}) = \frac{\lambda_i^{y_i} \exp(-\lambda_i)}{y_i!}$$

と書けます．全個体の尤度は

$$L(\{\beta_j\} \mid \{y_i\}, \{x_i\}) = \prod_{i=1}^{100} \frac{\lambda_i^{y_i} \exp(-\lambda_i)}{y_i!}$$

対数尤度は

$$\log L(\{\beta_j\} \mid \{y_i\}, \{x_i\}) = \sum_{i=1}^{100} \log \left\{ \frac{\lambda_i^{y_i} \exp(-\lambda_i)}{y_i!} \right\}$$

これを最大化するような $\hat{\beta}_1$ と $\hat{\beta}_2$ を求める，というのが一般化線形モデル (GLM) のパラメーターの最尤推定，ということになります．

R ではたいへんお手軽に GLM のパラメーター推定ができるようになっていて，

```
> fit <- glm(y ~ x, data = d, family = poisson)
```

とするだけでモデルの**あてはめ** (fitting) とその推定結果 [31] が (またしても私がてきとうに名づけた) `fit` なるオブジェクトに格納されます．

31. だけでなくその他いろいろなものが…… 興味があるヒトは `names(fit)` としてみてください．

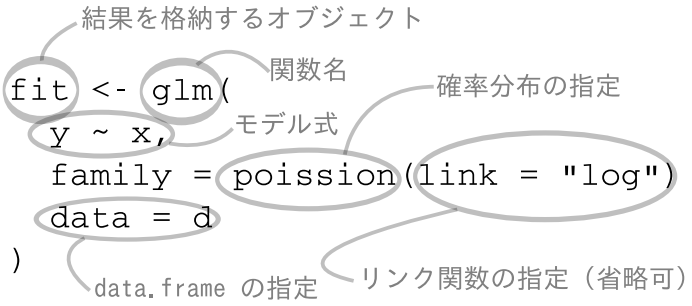


図 5: `glm()` 関数の指定

いろいろ指定している内容は図 5 のとおりなのですが……ここで `family = poisson` は「分布はポアソン分布を使ってね」という指示です．これは正式には `family = poisson(link = "log")` というように link 関数も指定すべきなのですが，`poisson` family における default link function [32] は `"log"` なので log link 関数を使いたいときはとくにわざわざ指定する必要ありません．

32. あるいは canonical link function (正準 link 関数)，あとでまた出ます．

それでは `fit` を表示してみると

```
> fit # これはいつもと同じく print(fit) した場合と同じ動作です

Call:  glm(formula = y ~ x, family = poisson, data = d)

Coefficients:
(Intercept)            x
    1.29172      0.07566

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      89.51
Residual Deviance: 84.99        AIC: 474.8
```

さらに summary() 関数を使うと，もっと詳細な推定結果を表示してくれます.

```
> summary(fit)

Call:
glm(formula = y ~ x, family = poisson, data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3679  -0.7348  -0.1775   0.6987   2.3760

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.29172    0.36369   3.552 0.000383 ***
x            0.07566    0.03560   2.125 0.033580 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 89.507  on 99  degrees of freedom
Residual deviance: 84.993  on 98  degrees of freedom
AIC: 474.77

Number of Fisher Scoring iterations: 4
```

さてさて……あちこちに Deviance なるコトバが登場しますが，これについてはまたあとで説明することにしましょう．さて，上の summary() 出力のなかで係数 (coefficient) の情報が示されている部分，

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.29172    0.36369   3.552 0.000383 ***
x            0.07566    0.03560   2.125 0.033580 *
```

あたりの読みかただけを先に簡単に説明してみましょう．まず Coefficients (係数) というのは推定したいパラメーター，つまり (Intercept) (「切片」) に対応する係数は (さきほどの定式化でいうと) β_1 で x に対応する係数は β_2 ということです.

Estimate は前回の講義でもでてきた**推定値**のことで，つまりこの場合ですと最尤推定値，$\hat{\beta}_1 \approx 1.29$ で $\hat{\beta}_2 \approx 0.0757$ ということです.

さーてこれの次からアヤしい世界に突入するのですが……Std. Errror はパラメーターの標準誤差の推定値です.「標準誤差とは何か?」なんてことをマジメに議論するとけっこうたいへんなので，ここでは「最尤推定値は得られたけど，じつはデータをちょっと変えると推定値もけっこう変わるんだよね，その**推定値のばらつき**」ぐらいの**いいかげんな理解**ですませることに

しましょう．この場合ですと $\hat{\beta}_1$ の標準誤差がでかくて，$\hat{\beta}_2$ の標準誤差が小さいみたいだ，つまり $\hat{\beta}_1$ のほうがふらつくのね，といった意味だと思ってください．

なぜこれが「アヤしい」かといえば，標準誤差の推定計算にある方法 [33] を使っているのですが，ホントにそれでいいのか? という疑問があるからです．だからといって他によい方法がないのでたいていの統計ソフトウェアではそれを採用しています．

で，めんどうなことにこの標準誤差 (SE) は最尤推定値に比例するといったハナシもあるので，`z value` [34] なる (最尤推定値) / SE と定義される割算値 (Wald 統計量) を計算しています．そのココロをてきとーに図示してみますと図 6 のごとくで，$\hat{\beta}_*$ たちがゼロから十分に離れているかどうか，「$\hat{\beta}_1$ は SE のわりにはゼロから離れてるね」「$\hat{\beta}_2$ ゼロに近いわかりには SE がでかいね」ぐらいのキモチをを表現しているわけです．で，これを「検定」しようというのが Wald 検定 [35] なるもので，`Pr(>|z|)` ってのは `z value` の絶対値というか $\hat{\beta}_*$ の「分布のスソ」がゼロにどれぐらいかかっているか，をあらわしてます．

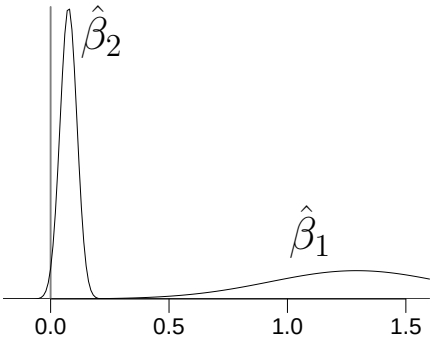


図 6: 最尤推定値の分布?

で，[36] 私はこの係数 (パラメーターの推定値) の Wald 検定なるものはアテにならないいいかげんなもの，R の `summary(glm(...))` で出てくるのは，まあちょっとした「めやす」みたいなモノ，だと考えてます．[37]

こういうアヤしげな「検定」より重要なことは，推定結果を使って統計モデルの**予測 (prediction) がいかなるものか**，を見てみることです．R で図示してみると……

33. 最尤推定値付近での対数尤度の「とがりかた」から最尤推定値の「標準偏差」を推定しています．

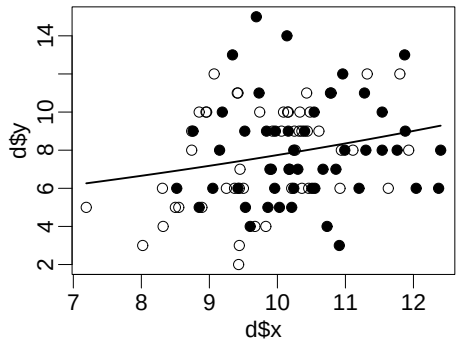
34. これはさきほど線形予測子を定義してた z とは無関係です．

35. 考えてみると，これってふつーの (Neyman-Pearson 的なわくぐみ) の検定ではありませんね．信頼区間を評価というか……

36. このあたりの私のイヤイヤな口調からわかるように

37. あるパラメーターがゼロと見なすべきかどうかはあとで登場する**モデル選択**の手法で決めてしまえばよいのです．

```
> plot(d$x, d$y, pch = c(21, 19)[d$f])
> xx <- seq(min(d$x), max(d$x), length = 100)
> lines(xx, exp(1.29 + 0.0757 * xx), lwd = 2)
```



ここで「予測された平均種子数」は`lines(xx, exp(1.29 + 0.0757 * xx), lwd = 2)`というふうに推定値を代入して平均値 λ を計算しています．これがめんどくさければ，

```
> yy <- predict(fit, newdata = data.frame(x = xx), type = "response")
> lines(xx, yy, lwd = 2)
```

というふうに `predict()` 関数を使う，というよりお手軽な手法なんかもあります．

最尤推定値は対数尤度 $\log L(\{\beta_j\} \mid \text{データ})$ を最大化するような $\{\hat{\beta}_i\}$ です．つまり $\{\hat{\beta}_i\}$ のもとでは対数尤度 $\log L(\{\beta_j\} \mid \text{データ})$ は最大化されています．このモデルで**最大化された対数尤度**は `logLik(fit)` で評価することができて，

```
> logLik(fit) # fit <- glm(y ~ x, ...)
'log Lik.' -235.3863 (df=2)
```

最大化された対数尤度は -235.4 ぐらい，とわかります．(df=2) ってのは使っているパラメーター数，つまり β_1 と β_2 の 2 個つかってますよ，ということですね．

5. 因子型の説明変数の場合

ここまでは「図にしたときのわかりやすさ」を重視して，説明変数が「植物のサイズ」こと x_i だけのモデルについて説明してきました．

次に「なかったこと」あつかいだった施肥効果 f_i だけをくみこんだモデ

ルも説明してみましょう．前に示したように f 列は因子 (factor) 型になっていますが,

```
> d$f
  [1] C C C C C C C C C C C C C C C C C C C C C C C C C
 [26] C C C C C C C C C C C C C C C C C C C C C C C C C
 [51] T T T T T T T T T T T T T T T T T T T T T T T T T
 [76] T T T T T T T T T T T T T T T T T T T T T T T T T
Levels: C T
```

R の glm() のありがたいところは，説明変数がどういう型だろうが [38] とくに何もせずに同じように統計モデルを指定できるところにあります．その指定と推定結果をながめてみましょう．

38. とはいえ numeric, integer, factor に限定されますが.

```
> fit.f <- glm(y ~ f, data = d, family = poisson)
> fit.f

Call:  glm(formula = y ~ f, family = poisson, data = d)

Coefficients:
(Intercept)           fT
    2.05156      0.01277

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      89.51
Residual Deviance: 89.48        AIC: 479.3
```

このようにモデルの指定は説明変数が因子型であっても glm(y ~ f, ...) とすればよいのです．また Coefficients 出力をみると (Intercept) (「切片」) は先ほどと同じですが，施肥効果 f の係数の名前は fT になっています．これは f には C (肥料なし) と T (施肥処理) の 2 水準があり (順番もこのとおり)，1 番目の水準である C を基準つまりゼロとおき，2 番目以降 [39] の水準の「ずれ」をみてやろう，ということです．つまり，もし個体 i の f が C ならば

39. ここでは 2 番目までしかありませんが.

$$\lambda_i \approx \exp(2.05 + 0) = \exp(2.05)$$

であり，もし T ならば

$$\lambda_i \approx \exp(2.05 + 0.0128) = \exp(2.0628)$$

である，という予測になります．ここでは「肥料をやると種子数がほんの少しだけ増える」という予測になっています.

このモデルで最大化された対数尤度はこうなっています.

```
> logLik(fit.f) # fit.f <- glm(y ~ f, ...)
'log Lik.' -237.6273 (df=2)
```

6. 説明変数が数量型 + 因子型のモデル

施肥効果 f はごく簡単に glm() であつかえたので，だったらサイズ x といっしょに統計モデル化，つまりサイズと施肥効果が同時に種子数に影響をあたえるモデルなんかの推定計算もできるのでは，というのがこれになります.

```
> fit.full <- glm(y ~ x + f, data = d, family = poisson)
> fit.full

Call:  glm(formula = y ~ x + f, family = poisson, data = d)

Coefficients:
(Intercept)            x           fT
    1.26311      0.08007     -0.03200

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:      89.51
Residual Deviance: 84.81        AIC: 476.6
```

こんどは肥料の効果 fT がマイナスだと推定されていますね.

それはともかくとして，先ほどから放置していた問題「log link 関数では要因の効果が積であらわされる」ということの解釈についてここで説明したいと思います.

上の推定結果があらわしていることは，もし個体 i のサイズを x_i とすると，施肥処理が C ならば

$$\lambda_i \approx \exp(1.26 + 0.08x_i)$$

であり，もし T ならば

$$\lambda_i \approx \exp(1.26 + 0.08x_i - 0.032)$$

という予測になっています．これからわかるように

$$\begin{aligned} \exp(1.26 + 0.08x_i - 0.032) &= \exp(1.26 + 0.08x_i) \times \exp(-0.032) \\ &= (\text{切片とサイズ}) \times (\text{施肥処理の効果}) \end{aligned}$$

となっていますから，施肥処理の効果は glm(y ~ x + f, ...) と定義したにもかかわらず，足し算できいてるのではなくかけ算で影響してくる，とい

うことです．この場合ですと $\exp(-0.032) \approx 0.969$ ですから，肥料をやると種子数の平均が 0.969 倍になるという予測になります．

このモデルで最大化された対数尤度はこうなっています．

```
> logLik(fit.full) # fit.full <- glm(y ~ x + f, ...)
'log Lik.' -235.2937 (df=3)
```

7. Deviance って何なの?

さてさて，ここまで

- サイズの効果のみのモデル
- 施肥効果のみのモデル
- サイズの効果 × 施肥効果のモデル

を調べてきてサイズの効果はともかく，施肥効果については首尾一貫しない結果が得られました．いったいどのモデルが「よい」のでしょうか? それを決めるのが**モデル選択**でこれは「あてはまりの良さ」と「モデルの複雑さ」のかねあいで決まります．そして「あてはまりの良さ」とは，この最尤推定法の世界においては**最大化された対数尤度**があらわしています．[40]

モデル選択にハナシをすすめる前に，ここまで登場してきたけれどトバされていた最大化された対数尤度，つまり deviance について検討してみましょう．[41]

最尤推定値は対数尤度 $\log L(\{\beta_j\} \mid \text{データ})$ ……えーい書くのがめんどうなので $\log L$ を最大化するような $\{\hat{\beta}_i\}$ です．このように最大化された対数尤度を $\log L^*$ としましょうか．モデルごとの $\log L^*$ の大小を比較すれば，どのモデルがあてはまりがよいか，といったことが検討できそうです．

しかしながら，ここまで示してきた `summary(glm(y ...))` では $\log L^*$ は示されず[42] そのかわり deviance なる数量が表示されています．

さて **deviance** はこういった目的のために対数尤度を組み合わせた指標で

$$D = -2 \log L^*$$

と定義されます．つまり対数尤度に -2 をかけているだけです．[43]

たとえば最初のモデル，つまり架空植物のサイズ x_i だけで平均種子数

$$\lambda_i = \exp(\beta_1 + \beta_2 x_i)$$

40. すでに各モデルごとに `logLik()` 関数で計算させてますよね．これはモデル間で比較可能な数量です……ただし離散確率分布モデルと連続確率分布モデルの比較，といった「原理的にありえない」状況では当然ながら比較できません．

41. 訳語としては尤離度（ゆうりど）ってのがあるのですが，あまり普及してないのでここでは deviance と呼びます．

42. わざわざ `logLik()` 関数を使って $\log L^*$ を評価してましたね．

43. なんで -2 なんかをかけるの? というのは…… まあ歴史的経緯というやつでして，この場合は尤度比検定の検定統計量と関係があります．統計学にはこういう歴史的なんちゃらがいろいろあります．

と定義されるモデルについて説明してみましょう．これを仮に「**サイズだけモデル**」とよびます．

サイズだけモデルの最大化対数尤度 $\log L^*$ は -235.4 ぐらいでしたから deviance ($D = -2\log L^*$) は 470.8 ぐらいになります．しかしながら，このモデルの `glm()` の出力結果をみても，

```
... (中略) ...
Null Deviance:      89.51
Residual Deviance: 84.99        AIC: 474.8
```

どこにも 470.8 なる数値はでてきません．そのかわり `Null Deviance` だとか `Residual Deviance` だとか `AIC` といった数値が示されています．

ここで「観測データに統計モデルがあてはまっているか」をあらわしているのは `Residual Deviance` なので，[44] まずはこれについて説明してみましょう．Residual deviance はある deviance D に対して

$$D - (\text{ポアソン分布で可能な最小 } D)$$

と定義されます．[45] この「(ポアソン分布で可能な最小 D)」ってのは何なのでしょうか? それは "**full model**" などと呼ばれているモデル，たとえば **データ数が 100 個ならパラメーターを 100 個 (!)** 使って「あてはめた」モデルということです．[46]

この例題の場合ですと，`data.frame` である `d` に格納されている種子数は

```
> d$y
  [1]  6  6  6 12 10  4  9  9  9 11  6 10  6 10 11  8
... (後略) ...
```

となっていました.「サイズだけモデル」はサイズ x_i で決まる平均種子数 λ_i でこの `d$y` が説明できると考えていました．これにたいして full model は

- 1, 2, 3 個目の `y` は 6 なので平均 6 のポアソン分布
- 4 個目の `y` は 12 なので平均 12 のポアソン分布
- 5 個目の `y` は 10 なので平均 10 のポアソン分布
- ... (以下略) ...

というふうに 100 個のデータに対して 100 個のパラメーターを準備して「あてはめる」モデルです．このようなモデルはぜんぜんデータの情報を圧縮していない,「種子数はどのように決まるか?」にマトモに答えていない価値の

44. Residual というのは統計学な用語で「残差」と訳されます.

45. Deviance や residual deviance は大きいほどあてはまりが悪い，つまり deviance は「あてはまりの悪さ」をあらわしています.

46. これに対して「サイズだけモデル」は β_1 と β_2 の 2 個しかパラメーターを思いだしてください.

ない統計モデルです．しかしこのモデルを適用したときに，対数尤度は最大化され，[47]

```
> sum(dpois(d$y, lambda = d$y, log = TRUE))
[1] -192.8898
```

「サイズだけモデル」の $\log L^*$ が -235.4 でしたからこれは「ケタちがい」[48] に大きい最大化対数尤度になっています．対数尤度が最大化される，ということは，つまりそれは deviance $D = -2\log L^* \approx 385.8$ が最小化される，ということです．これが「(ポアソン分布で可能な最小 D)」というもなので，「サイズだけモデル」の residual deviance は

$$D - (\text{ポアソン分布で可能な最小 } D) \approx 470.8 - 385.8 = 85.0$$

となるので先に示した `glm()` のアウトプット

```
Residual Deviance: 84.99        AIC: 474.8
```

と一致していることがわかりますね．つまり residual deviance とは「すごーくあてはまりが良い」場合に最小値ゼロをとるような値となります．[49]

それでは residual deviance の最大値はどのように決まるのでしょうか? Residual deviance が最大になるのは「もっともあてはまりの悪いモデル」であり，それは (ポアソン分布を部品として使っているときに) 「もっともパラメーター数の少ないモデル」つまり平均種子数が

$$\lambda_i = \exp(\beta_1)$$

というふうに 1 パラメーターだけで決まっているモデルです．これは R の中では “**null model**” と呼ばれています．[50] それではこの null model を使った推定計算を試みてみましょう．

```
> fit.null <- glm(formula = y ~ 1, family = poisson, data = d)
> fit.null

Call:  glm(formula = y ~ 1, family = poisson, data = d)

Coefficients:
(Intercept)
      2.058

Degrees of Freedom: 99 Total (i.e. Null);  99 Residual
Null Deviance:      89.51
Residual Deviance: 89.51        AIC: 477.3
```

ポアソン分布を仮定しているときの最大 residual deviance である 89.5 といった値が示されていて，これは null model のもとでの最大化対数尤度

47. このことから統計モデリングにおいてパラメーターを増やしまくって「あてはまりの良さ」だけをひたすら追及するのは無意味らしい，という気がしてきますよね．

48. 対数の世界のコトなので．

49. しかし residual deviance がゼロになるような統計モデルは，さきほどの full model ごときもので観測された現象をできるだけ簡単に説明しようとする統計モデルとしては無価値です．

50. Null というと帰無仮説 (null hypothesis) なるコトバを連想させるわけですが……null model は「なんとなく帰無仮説的なかんじのモデル」というぐらいの意味なのでしょう．

```
> logLik(fit.null) # fit.null <- glm(y ~ 1, ...)
'log Lik.' -237.6432 (df=1)
```

から算出されています.

じつは null model の residual deviance は他のモデル，たとえば「サイズだけモデル」の glm() アウトプットにも含まれていて,

```
...（中略）...
Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      89.51
Residual Deviance: 84.99        AIC: 474.8
```

この Null Deviance というのがそれに該当します．つまりこれは

- 「サイズだけモデル」の residual deviance は 85.0 ぐらい
- 説明変数何もないモデルの residual deviance 89.5 ぐらい
- ということで，サイズ x_i という説明変数を追加することで residual deviance が 4.1 ぐらい [51] 改善されたね

51. これはモデル間の対数尤度の差になります.

といったことを示しているわけです.

ついでに (最尤推定後も残されている) degree of freedom (自由度) に関する

```
Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
```

このアウトプットもいまや理解できるわけで,

- Null model の (残されている) 自由度は 99 (データ数 100 - パラメーター数 1)
- 「サイズだけモデル」の自由度は 98 (100 - パラメーター数 2)

という意味です．ここまで登場したモデルのパラメーター数 (k)，最大化対数尤度 ($\log L^*$)，deviance ($D = -2\log L^*$)，residual deviance ($-2\log L^* -$ 最大 D) をまとめてみます.

Model	k	$\log L^*$	**Deviance** $-2\log L^*$	**Residual deviance**
NULL	1	-237.6	475.3	89.5
f	2	-237.6	475.3	89.5
x	2	-235.4	470.8	85.0
x + f	3	-235.3	470.6	84.8
FULL	100	-192.9	385.8	0.0

この table をみると，パラメーター数さえ増やせば residual deviance はどんどん小さくなる，つまり「あてはまりが良くなる」ことがわかります.

8. 「よい」モデルを選ぶモデル選択

統計モデルにおいて「あてはまりの良さ」つまり対数尤度の最大化 (= deviance の最小化) だけを**ひたすら**おいもとめるのはどうもヘンらしい，ということが先ほどの full model の検討でわかってきました. [52]

52. 正規分布を使った回帰などで R^2 最大化をひたすら追求するのも同様にばかばかしいことです.

統計モデリングでは

- できるだけ最大化対数尤度が大きいあてはまりのよいモデル
- できるだけパラメーター数が少ない簡単なモデル

という矛盾したふたつの要求をバランスよく満たすような統計モデルを見つけることを **モデル選択** (model selection) とよび，このときにモデルのよしあしを決めるのがモデル選択規準です．たとえば良く使われてるモデル選択規準 AIC (Akaike's information criterion) は

$$\begin{aligned} \mathrm{AIC} &= -2(\text{最大化対数尤度}) + 2(\text{そのモデルで使ってるパラメーター数}) \\ &= -2\log L^* + 2k \end{aligned}$$

と定義され，これが**一番小さいモデルがよいモデル**として選択されます．今回登場したいろいろなポアソン分布モデルについて AIC を評価してみると，

Model	k	$\log L^*$	**Deviance** $-2\log L^*$	**Residual deviance**	**AIC**
`NULL`	1	-237.6	475.3	89.5	477.3
`f`	2	-237.6	475.3	89.5	479.3
`x`	2	-235.4	470.8	85.0	474.8
`x + f`	3	-235.3	470.6	84.8	476.6
`FULL`	100	-192.9	385.8	0.0	585.8

となり,「サイズだけモデル」(Model `x`) があてはまりの良さとモデルの単純さを兼ねそなえた「よい」統計モデルとして選ばれました. [53]

53. ヒトことだけ説明するなら，あてはまり**ではなく**「予測力がよい」という意味で「よい」モデルなのです.

次のページに示していますが，R では `stepAIC()` 関数を使うことで，手軽に AIC による `glm()` のモデル選択ができます.

```
> library(MASS) # stepAIC を定義する MASS package よみこみ
> stepAIC(fit.full) # 「全部いりモデル」を使う
Start:  AIC=476.59
y ~ x + f

       Df Deviance    AIC
- f     1    84.99 474.77
<none>       84.81 476.59
- x     1    89.48 479.25

Step:  AIC=474.77
y ~ x

       Df Deviance    AIC
<none>       84.99 474.77
- x     1    89.51 477.29

Call:  glm(formula = y ~ x, family = poisson, data = d)

Coefficients:
(Intercept)            x
    1.29172      0.07566

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      89.51
Residual Deviance: 84.99        AIC: 474.8
```

モデル選択については次の回 & 次の次の回でさらにとりあげていく予定なので，今日のところはこれぐらいで終えることにしましょう.

9. 今日のまとめ

今日は以下にあげているようなことを説明してみました.

- 一般化線形モデル (GLM) はポアソン回帰・ロジスティック回帰・正規分布を仮定する直線回帰や (いわゆる) ANOVA を統合した呼びかたである
- R では `glm()` 関数でまとめてあつかえる
- GLM の基本部品: 確率分布・link 関数・線形予測子
- ポアソン回帰は確率分布 = ポアソン分布 (`poisson`)，link 関数 = `log`
- `glm()` 関数は GLM のパラメーターを最尤推定する

- 推定結果を使って予測 (prediction) ができる
- Deviance は -2 × 最大化対数尤度，あてはまりの悪さ
- AIC = Deviance + 2 × (パラメーター数)，あてはまりの良さとパラメーター数の「バランス」をとる

次回はまた別の「正規分布では説明できない現象の統計モデリング」として，二項分布と logit link 関数を使う GLM である**ロジスティック回帰**について説明します.