統計・データ解析セミナーの予習

GCOE　アジア保全生態学

2010.11.24　　粕谷英一（理・生物・生態）

本日のメニュー

R
一般化線形モデル（Generalized Linear Models、略して GLM）
R で GLM を使う
R でグラフを描く

説明しないこと：R でできること全般―たくさんあるので時間的に無理
　　R でするプログラミング－データ解析なら使いやすい

R

【起動と終了】
起動は他のアプリケーションと同じ
終了は、コマンド画面（R の基本的かつ主な画面）で、q()　と入力するか、メニューから終了を選ぶ。終了時には、作業スペースを保存するか（オブジェクトなどが保存される）どうか聞いてくる。

関数などがどんなものかわからないとき
　? 関数の名前
　help(関数の名前)
　?? 関数の名前

【簡単な例】データを入力し、相関係数を計算し、グラフを書く
データの入力と相関係数の計算

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
x1<-c(1.52,2,3.01,9,2,6.3,5,11.2)
y1<-c(4,0.21,-1.5,8,2,6,9.915,5.2)
cor(x1,y1)
cor.test(x1,y1)
```

- -
と入力すると以下のようになる
- -

```
> x1<-c(1.52,2,3.01,9,2,6.3,5,11.2)
> y1<-c(4,0.21,-1.5,8,2,6,9.915,5.2)
> cor(x1,y1)
[1] 0.5594521
> cor.test(x1,y1)

        Pearson's product-moment correlation

data:  x1 and y1
t = 1.6533, df = 6, p-value = 0.1494
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2397297  0.9066828
sample estimates:
      cor
0.5594521
```

- -

さらに
- -

```
hist(x1)
hist(y1)
plot(x1,y1)
```

- -
と（ゆっくり）入力すると

```
> hist(x1)
> hist(y1)
> plot(x1,y1)
```

とメインのウィンドウには出て、同時に自動的に別ウィンドウが開いて x1 のヒストグラム、y1 のヒストグラム、x1 が横軸で y1 が縦軸の散布図が表示される。

<-代入
+足し算　　-引き算　　*掛け算　　/割り算　　^べき乗
c()　ベクトルを作る
cor()　相関係数を計算する
cor.test()　相関係数を計算して信頼区間を求め、検定する

【簡単な例：続き　基本的な統計量の計算】
以下のように入力すると、
- -
mean(x1)
median(x1)
var(x1)
sd(x1)
range(x1)
- -
以下のようになる
- -
```
> mean(x1)
[1] 5.00375
> median(x1)
[1] 4.005
> var(x1)
[1] 12.88577
> sd(x1)
[1] 3.589675
> range(x1)
[1]  1.52 11.20
```
- -
基本的な統計量を計算する関数
mean()平均　median()中央値　var()分散
sd()標準偏差　range()　範囲
- -
　キー操作　上下の矢印キーは履歴の移動

【簡単な例：続き　オブジェクトの中味】

- -

```
cortest1<-cor.test(x1,y1)
cortest1
str(cortest1)
cortest1$estimate
```

- -

と入力すると、以下のようになる

- -

```
> cortest1<-cor.test(x1,y1)
> cortest1
        Pearson's product-moment correlation

data:  x1 and y1
t = 1.6533, df = 6, p-value = 0.1494
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2397297  0.9066828
sample estimates:
      cor
0.5594521

> cortest1$estimate
      cor
0.5594521

> str(cortest1)
List of 9
 $ statistic  : Named num 1.65
  ..- attr(*, "names")= chr "t"
 $ parameter  : Named num 6
  ..- attr(*, "names")= chr "df"
 $ p.value    : num 0.149
 $ estimate   : Named num 0.56
  ..- attr(*, "names")= chr "cor"
 $ null.value : Named num 0
```

```
> cortest1$estimate
      cor
0.5594521
```

- -

str()　その名前のものーオブジェクトーの中味の構造を表示
$　オブジェクトを構成するもの

【簡単な例：続き　ベクトルの操作】

- -

```
x1[1]
x1[5]
x1[x1<=8.5]
x1[1:5]
z1<-x1+y1*2
z1
z2<-(x1-x1/y1)
z2
z2[8] <-(216.5)
z2
z2<-(216.5)
z2
```

- -

```
> x1[1]
[1] 1.52
> x1[5]
[1] 2
> x1[x1<=8.5]
[1] 1.52 2.00 3.01 2.00 6.30 5.00
> x1[1:5]
[1] 1.52 2.00 3.01 9.00 2.00
> z1<-x1+y1*2
> z1
[1]  9.52  2.42  0.01 25.00  6.00 18.30 24.83 21.60
```

```
> z2<-(x1-x1/y1)
> z2
[1]     1.140000   -7.523810      5.016667      7.875000      1.000000      5.250000
4.495714
[8]    9.046154
> z2[8]<-(216.5)
> z2
[1]     1.140000    -7.523810      5.016667      7.875000      1.000000      5.250000
[7]     4.495714 216.500000
> z2<-(216.5)
> z2
[1] 216.5
```

- -

[]内の数字はベクトル内での要素の位置を表す
数字:数字　は連続した整数を表す

【簡単な例：続き　データフレーム】
　データフレームとはベクトルを複数まとめたようなもの（オブジェクトの形式の１つ）　統計的なデータ解析ではよく使われて便利
　　＜＜例の書き方を簡単にします＞＞

- -

```
> newdata1<-data.frame(x1,y1)
> newdata1
       x1        y1
1    1.52    4.000
2    2.00    0.210
3    3.01 -1.500
4    9.00    8.000
5    2.00    2.000
6    6.30    6.000
7    5.00    9.915
8 11.20    5.200
> names(newdata1)<-c("haba","nagasa")
> newdata1
    haba nagasa
```

```
1    1.52    4.000
2    2.00    0.210
3    3.01 -1.500
4    9.00    8.000
5    2.00    2.000
6    6.30    6.000
7    5.00    9.915
8 11.20    5.200
> summary(newdata1)
         haba                  nagasa
 Min.    : 1.520    Min.    :-1.500
 1st Qu.: 2.000    1st Qu.: 1.552
 Median : 4.005    Median : 4.600
 Mean    : 5.004    Mean    : 4.228
 3rd Qu.: 6.975    3rd Qu.: 6.500
 Max.    :11.200    Max.    : 9.915
> newdata1$haba
[1]    1.52    2.00    3.01    9.00    2.00    6.30    5.00 11.20
> newdata1$nagasa
[1]    4.000    0.210 -1.500    8.000    2.000    6.000    9.915    5.200
> newdata1[1,]
   haba nagasa
1 1.52          4
> newdata1[,1]
[1]    1.52    2.00    3.01    9.00    2.00    6.30    5.00 11.20
> newdata1[,"haba"]
[1]    1.52    2.00    3.01    9.00    2.00    6.30    5.00 11.20
> newdata1.01<-subset(newdata1,haba<=8.1)
> newdata1.01
   haba nagasa
1 1.52    4.000
2 2.00    0.210
3 3.01 -1.500
5 2.00    2.000
6 6.30    6.000
7 5.00    9.915
```

```
> newdata1.02<-newdata1
> newdata1.02$menseki<-newdata1.02$haba*newdata1.02$nagasa
> newdata1.02$menseki
[1]   6.080   0.420 -4.515 72.000   4.000 37.800 49.575 58.240
> newdata1.02
    haba nagasa menseki
1   1.52   4.000    6.080
2   2.00   0.210    0.420
3   3.01 -1.500   -4.515
4   9.00   8.000   72.000
5   2.00   2.000    4.000
```

- -

data.frame()　データフレームを作る
summary()
$
subset()　条件に合うデータだけからなるデータフレームを作る

【データファイルの読み込み】
他のソフトウェアで作られたデータファイルを R のデータフレームに読み込む
　基本的な考え方：コンピューターができそうなことはコンピューターにやらせる－省力化とまちがい減らし
　コピーしてクリップボードにあるものを読み込む、他の統計ソフト専用ファイルを読む、MS-Excel のファイルを読むなどの関数は各種揃っているが、ここでは、テキストファイルを読む（応用がきくから）
　以下のデータを 1124test1.txt という名前で保存しておく。区切り記号はタブ。

- -

nage	ura	taion	takasa
4	2	36.1	12.5
8	3	36.0	14.3
12	3	36.2	10.8
9	8	39.0	14.4
6	3	36.5	14.0
24	15	38.0	11.8
8	0	35.8	13.8
18	3	36.3	13.3

```
11       4        36.9    11.9
10       6        37.0    14.2
8        3        37.1    14.4
3        2        38.4    12.5
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> mydata01<-read.table("1124test1.txt")
> mydata01
       V1   V2      V3       V4
1   nage ura taion takasa
2       4    2   36.1    12.5
3       8    3   36.0    14.3
4      12    3   36.2    10.8
5       9    8   39.0    14.4
6       6    3   36.5    14.0
7      24   15   38.0    11.8
8       8    0   35.8    13.8
9      18    3   36.3    13.3
10     11    4   36.9    11.9
11     10    6   37.0    14.2
12      8    3   37.1    14.4
13      3    2   38.4    12.5
> mydata01<-read.table("1124test1.txt",header=T)
> mydata01
    nage ura taion takasa
1       4    2   36.1    12.5
2       8    3   36.0    14.3
3      12    3   36.2    10.8
4       9    8   39.0    14.4
5       6    3   36.5    14.0
6      24   15   38.0    11.8
7       8    0   35.8    13.8
8      18    3   36.3    13.3
9      11    4   36.9    11.9
10     10    6   37.0    14.2
11      8    3   37.1    14.4
12      3    2   38.4    12.5
```

- -
まずここまでやってみる。ディレクトリとファイル名の確認を忘れずに。
- -

```
> mydata01$uraritu<-(mydata01$ura/mydata01$nage)
> mydata01
    nage ura taion takasa     uraritu
1      4   2   36.1   12.5 0.5000000
2      8   3   36.0   14.3 0.3750000
3     12   3   36.2   10.8 0.2500000
4      9   8   39.0   14.4 0.8888889
5      6   3   36.5   14.0 0.5000000
6     24  15   38.0   11.8 0.6250000
7      8   0   35.8   13.8 0.0000000
8     18   3   36.3   13.3 0.1666667
9     11   4   36.9   11.9 0.3636364
10    10   6   37.0   14.2 0.6000000
11     8   3   37.1   14.4 0.3750000
12     3   2   38.4   12.5 0.6666667
> plot(mydata01)
```

- -
read.table()　表の形になっているファイルを読み込む

【R とパッケージ】R には特定の目的のための関数などを集めたパッケージと呼ばれるものが多数ある。いずれかのパッケージに含まれている分析手法の数は膨大なので、自分がしたい分析はまずどこかのパッケージにないか探すといい。パッケージに含まれている関数などを使いたいときには、そのパッケージをネット上から自分のコンピューターに入れておき（パッケージのインストール)、使うときにインストールされているパッケージをロードする（パッケージの読み込み)。たまたま、異なるパッケージに同じ名前の関数があると、後から読み込まれた方が使われる（警告が出る)。

【R と GUI】R は、基本的にはコマンドラインに文字を入力して動かす（実は統計ソフトの多くはそうである)。しかし、もっとグラフィカルインターフェースっぽく使いたいときには、そのようなパッケージなどもある。代表的には R commander（Rcmdr とも呼ばれる)。

==========

R 補足

#全オブジェクト消去（みな消えてしまえ）

rm(list=ls(all=TRUE))

あるいは

rm(list=ls())

一般化線形モデル

回帰を拡張（一般化）したものです

回帰とは

回帰のパーツ説明　説明変数（昔は独立変数）　目的変数（応答変数、昔は従属変数）　誤差（残差）

説明変数の式が目的変数の期待値を決め、実際の目的変数の値はその期待値のまわりにばらつく　目的変数の期待値と実際の値の差を誤差と呼ぶ

たとえば、y=3x-1 で、x=3 で y=7 というデータがあったとすると、
x=3 に対する目的変数 y の期待値は 8 で、残差は-1 である。3x-1 の 3 を回帰係数、-1 を切片という。

直線回帰で、説明変数を複数にすると（線形）重回帰
重回帰で、説明変数に名義変数まで拡張すると、一般線形モデル（正規線形モデル）
一般線形モデルで、説明変数の一次式以外のもの（の一部）と目的変数の分布を等分散の正規分布以外のものまで拡張すると、一般化線形モデル

一般化線形モデルのパーツ：３つの構成要素
・線形予測子 linear predictor　説明変数の一次式のこと

・リンク関数（連結関数）link funtion　説明変数の一次式と目的変数の予測値（期待値）の関係
　　リンク関数（目的変数の予測値）＝線形予測子

・誤差分布（構造）error structure　目的変数の予測値（期待値）のまわりのばらつきの分布

リンク関数の例：identity（そのまま）、log（対数）、logit（ロジット）、inverse

（逆数)、(相補的 log-log)

誤差分布の例　等分散の正規分布（分散が一定)、ポアソン分布（分散は平均と等しい)、二項分布（分散=観察された個数×確率×(1-確率))、ガンマ分布（分散は平均の二乗に比例)
ポアソン分布：単位時間（単位面積）あたり一定の率で生じるイベントの回数（ものの個数)。回数なので非負の整数。イベント回数や個体数を分析するときの基本
二項分布：一定の確率で２つのできごとのどちらかが起こる現象を n 回繰り返したときの片方のできごとが起こる回数。生存 vs 死亡やメス vs オス、ある場所にいる vs 他のところにいるといったデータを分析するときの基本

一般化線形モデルの例：
リンク関数が identity、誤差分布が等分散の正規分布
　直線回帰、(元々の意味での）重回帰、分散分析
リンク関数が logit、誤差分布が二項分布
　ロジスティック回帰、対数線形モデルの一部
リンク関数が 対数、誤差分布がポアソン分布
　ポアソン回帰

一般化線形モデルにおける必須知識
線形予測子関係
　名義変数はダミー変数にして扱う
　オフセット　いつも回帰係数が 1 の説明変数
　交互作用（interaction）　ある説明変数が目的変数に与える効果が、他の説明変数の値が変わると変わる　両変数の積を説明変数にする

（偏）回帰係数の意味：他の説明変数の値をすべて一定に保って、その説明変数の値を 1 増やしたときに目的変数の期待値に与える効果

尤度（likelihood)
　確率ないしそれに準じるもの（連続的な量だと特定の値の確率は 0 なので確率密度を使う)
　確率かそれに準じるものなので、たいていは 1 より小さい
　尤度を最大にするように回帰係数や切片など（パラメーター）を決める（最尤法という)

対数尤度（log likelihood）

検定　Wald 検定　尤度比検定　スコア検定
　普通の（帰無）仮説
　　その説明変数が変化しても目的変数の期待値に変化なし
　　回帰係数（パラメーター）が 0 か
　Wald 検定や尤度比検定やスコア検定はサンプル数が大きいときに正しい
　サンプル数が少ないときにはパラメトリック・ブートストラップなど

最適な予測式（モデル選択）
　　AIC（赤池情報量規準）
　　自由パラメーター数×2＋最大対数尤度のマイナス 2 倍

誤差構造関係
　overdispersion（分散が過大）：二項分布やポアソン分布では平均値を決める確率や単位当たり発生率が決まれば分散も決まる。しかし、実際には、平均に対して、現実の分散はこの分布からの理論値よりも大きいことが多い。そのため、二項分布やポアソン分布のつもりで分析すると、偏りを過大に評価してしまう
　quasi-likelihood（擬似尤度、準尤度）　平均と分散の関係はデータ解析上重要なので、平均が変わったとき分散がどう変化するかだけを問題にした尤度もどき。

多重共線性　説明変数の中に相関が非常に強いものがあると、結果が不安定になる（データのごくわずかなちがいで、回帰係数の値などが大きく変化）

R で GLM を使う

glm()という関数
　３構成要素　線形予測子を formula（記号~）、誤差を family、リンク関数を link で指定する。

glm(目的変数~説明変数,family= かんとか(link="何とか "))

等分散の正規分布、identity リンクの例（直線回帰と同じ）

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> gx1<-c(1,3,2,5.1,4.02,2.8,5,6,7,8)
> gy1<-c(4.02,2.2,5.1,4.2,3.5,1,7,8.5,9.1,7.3)
>##ここまではデータの準備
> res.n01<-glm(gy1~gx1,family=gaussian(link="identity"))
> res.n01

Call:  glm(formula = gy1 ~ gx1, family = gaussian(link = "identity"))

Coefficients:
(Intercept)          gx1
     1.3779       0.8684

Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
Null Deviance:      65.68
Residual Deviance: 31.66        AIC: 45.9
> summary(res.n01)

Call:
glm(formula = gy1 ~ gx1, family = gaussian(link = "identity"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8095  -1.5474   0.1274   1.7410   1.9852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3779     1.4449   0.954   0.3682
gx1           0.8684     0.2962   2.932   0.0189 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 3.957456)
```

```
    Null deviance: 65.682  on 9  degrees of freedom
Residual deviance: 31.660  on 8  degrees of freedom
AIC: 45.903

Number of Fisher Scoring iterations: 2

> anova(res.n01)
Analysis of Deviance Table

Model: gaussian, link: identity

Response: gy1

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev
NULL                     9     65.682
gx1   1   34.022         8     31.660

> anova(res.n01,test="F")
Analysis of Deviance Table

Model: gaussian, link: identity

Response: gy1

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev      F    Pr(>F)
NULL                     9     65.682
gx1   1   34.022         8     31.660 8.597 0.01894 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- -

gaussian は等分散の正規分布の意味

family を指定して、リンク関数を指定しないとデフォールトのリンク関数が使われる（たまたまそれを使いたければ、link を省略できる）

- -

```
> res.n02<-glm(gy1~gx1,family=gaussian)
> res.n02

Call:  glm(formula = gy1 ~ gx1, family = gaussian)

Coefficients:
(Intercept)          gx1
     1.3779       0.8684

Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
Null Deviance:      65.68
Residual Deviance: 31.66        AIC: 45.9
> summary(res.n02)

Call:
glm(formula = gy1 ~ gx1, family = gaussian)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8095  -1.5474   0.1274   1.7410   1.9852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3779     1.4449   0.954   0.3682
gx1           0.8684     0.2962   2.932   0.0189 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 3.957456)
```

```
    Null deviance: 65.682  on 9  degrees of freedom
Residual deviance: 31.660  on 8  degrees of freedom
AIC: 45.903

Number of Fisher Scoring iterations: 2
```

- -

この例は等分散の正規分布（のつもり）なので、一般線形モデル用の関数 lm() も使える

- -

```
> res.n03<-lm(gy1~gx1)
> res.n03

Call:
lm(formula = gy1 ~ gx1)

Coefficients:
(Intercept)          gx1
     1.3779       0.8684

> summary(res.n03)

Call:
lm(formula = gy1 ~ gx1)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8095 -1.5474  0.1274  1.7410  1.9852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3779     1.4449   0.954   0.3682
gx1           0.8684     0.2962   2.932   0.0189 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.989 on 8 degrees of freedom
Multiple R-squared: 0.518,          Adjusted R-squared: 0.4577
F-statistic: 8.597 on 1 and 8 DF,   p-value: 0.01894
```

- -

結果は同じ

説明変数が名義尺度のとき。
片方（以下の例では a）に 0、もう片方（以下の例では b）に 1 を割り当てるダミー変数を使って分析している。

- -

```
> gx2<-c("a","b","a","b","a","a","a","a","b","b")
> gx2
 [1] "a" "b" "a" "b" "a" "a" "a" "a" "b" "b"
> summary(gx2)
   Length      Class       Mode
       10  character  character
> res.n04<-glm(gy1~gx2,family=gaussian(link="identity"))
 警告メッセージ：
In model.matrix.default(mt, mf, contrasts) :
   変数 'gx2' は因子に変換されました
> res.n04

Call:  glm(formula = gy1 ~ gx2, family = gaussian(link = "identity"))

Coefficients:
(Intercept)         gx2b
     4.8533       0.8467

Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
Null Deviance:      65.68
Residual Deviance: 63.96        AIC: 52.94
> summary(res.n04)

Call:
glm(formula = gy1 ~ gx2, family = gaussian(link = "identity"))
```

```
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-3.8533  -1.4633  -0.2933   2.0100   3.6467

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.8533     1.1544   4.204  0.00298 **
gx2b          0.8467     1.8252   0.464  0.65510
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 7.995167)

    Null deviance: 65.682  on 9  degrees of freedom
Residual deviance: 63.961  on 8  degrees of freedom
AIC: 52.936

Number of Fisher Scoring iterations: 2
```

- -

途中のエラーメッセージはデータの作成手順によっては出ないこともある
回帰係数や切片の意味を確かめてみる

- -

```
> cka<-c(1,3,5,6,7,8)
> gx2[cka]
[1] "a" "a" "a" "a" "a" "a"
> ckb<-c(2,4,9,10)
> gx2[ckb]
[1] "b" "b" "b" "b"
> mean(gy1[cka])
[1] 4.853333
> mean(gy1[ckb])
[1] 5.7
> mean(gy1[ckb])- mean(gy1[cka])
[1] 0.8466667
```

- -

次は、目的変数が 2 項分布の例（ロジスティック回帰）
データは先ほどファイル読み込みで使った、

```
> mydata01
    nage ura taion takasa     uraritu
1      4    2   36.1    12.5 0.5000000
2      8    3   36.0    14.3 0.3750000
3     12    3   36.2    10.8 0.2500000
4      9    8   39.0    14.4 0.8888889
5      6    3   36.5    14.0 0.5000000
6     24   15   38.0    11.8 0.6250000
7      8    0   35.8    13.8 0.0000000
8     18    3   36.3    13.3 0.1666667
9     11    4   36.9    11.9 0.3636364
10    10    6   37.0    14.2 0.6000000
11     8    3   37.1    14.4 0.3750000
12     3    2   38.4    12.5 0.6666667
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
>res.b01<-glm(uraritu~taion+takasa,weight=nage,data=mydata01,family=b
inomial(link="logit"))

> res.b01

Call:  glm(formula = uraritu ~ taion + takasa, family = binomial(link =
"logit"),          data = mydata01, weights = nage)

Coefficients:
(Intercept)            taion           takasa
    -38.8384           1.0008           0.1169

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:          30.86
Residual Deviance: 9.084            AIC: 42.64
> summary(res.b01)

Call:
glm(formula = uraritu ~ taion + takasa, family = binomial(link = "logit"),
```

```
    data = mydata01, weights = nage)

Deviance Residuals:
      Min        1Q    Median        3Q       Max
-1.88153  -0.38548   0.06386   0.81892   1.21023

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -38.8384     9.4727  -4.100 4.13e-05 ***
taion         1.0008     0.2388   4.191 2.77e-05 ***
takasa        0.1169     0.1709   0.684    0.494
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 30.8621  on 11  degrees of freedom
Residual deviance:  9.0836  on  9  degrees of freedom
AIC: 42.643

Number of Fisher Scoring iterations: 4

> anova(res.b01,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: uraritu

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                      11    30.8621
taion   1  21.3021        10     9.5600 3.923e-06 ***
takasa  1   0.4764         9     9.0836     0.49
```

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
- -
family が binomial（二項分布の意味）のときはデフォールトのリンクは logit なので省略しても同じ
- -
>
res.b02<-glm(uraritu~taion+takasa,weight=nage,data=mydata01,family=binomial)
> res.b02

Call: glm(formula = uraritu ~ taion + takasa, family = binomial, data = mydata01,
       weights = nage)

Coefficients:
(Intercept)   taion   takasa
    -38.8384   1.0008   0.1169

Degrees of Freedom: 11 Total (i.e. Null); 9 Residual
Null Deviance: 30.86
Residual Deviance: 9.084   AIC: 42.64
> summary(res.b02)

Call:
glm(formula = uraritu ~ taion + takasa, family = binomial, data = mydata01,
    weights = nage)

Deviance Residuals:
      Min   1Q   Median   3Q   Max
-1.88153 -0.38548   0.06386   0.81892   1.21023

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -38.8384   9.4727 -4.100 4.13e-05 ***
taion   1.0008   0.2388   4.191 2.77e-05 ***
takasa   0.1169   0.1709   0.684   0.494

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 30.8621  on 11  degrees of freedom
Residual deviance:  9.0836  on  9  degrees of freedom
AIC: 42.643

Number of Fisher Scoring iterations: 4

> anova(res.b02,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: uraritu

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                      11     30.8621
taion   1  21.3021         10      9.5600 3.923e-06 ***
takasa  1   0.4764          9      9.0836     0.49
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
データはこんな与え方も可能
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
>
res.b04<-glm(cbind(ura,(nage-ura))~taion+takasa,data=mydata01,family=bi
nomial(link="logit"))

> res.b04
```

```
Call:  glm(formula = cbind(ura, (nage - ura)) ~ taion + takasa, family =
binomial(link = "logit"),          data = mydata01)

Coefficients:
(Intercept)            taion          takasa
    -38.8384           1.0008           0.1169

Degrees of Freedom: 11 Total (i.e. Null);   9 Residual
Null Deviance:          30.86
Residual Deviance: 9.084             AIC: 42.64

> summary(res.b04)

Call:
glm(formula = cbind(ura, (nage - ura)) ~ taion + takasa, family =
binomial(link = "logit"),
      data = mydata01)

Deviance Residuals:
       Min             1Q       Median             3Q          Max
-1.88153   -0.38548     0.06386     0.81892     1.21023

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept) -38.8384         9.4727    -4.100 4.13e-05 ***
taion              1.0008         0.2388      4.191 2.77e-05 ***
takasa             0.1169         0.1709      0.684       0.494
---
Signif. codes:   0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

      Null deviance: 30.8621   on 11   degrees of freedom
Residual deviance:   9.0836   on   9   degrees of freedom
AIC: 42.643
```

Number of Fisher Scoring iterations: 4

> anova(res.b04,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(ura, (nage - ura))

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	P(>|Chi|)
NULL			11	30.8621	
taion	1	21.3021	10	9.5600	3.923e-06 ***
takasa	1	0.4764	9	9.0836	0.49

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
- -
目的変数を、率（部分の数/全体の数）で与えたときには、全体の数によりデータの重さが異なるので weight を指定する必要がある。

quasi-likelihood による overdispersion 対策（実際には、この例では overdispersion になっていない）
- -
>
res.qb04<-glm(cbind(ura,(nage-ura))~taion+takasa,data=mydata01,family=quasibinomial)
> res.qb04

Call: glm(formula = cbind(ura, (nage - ura)) ~ taion + takasa, family = quasibinomial, data = mydata01)

Coefficients:
(Intercept) taion takasa

```
    -38.8384          1.0008          0.1169

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:      30.86
Residual Deviance: 9.084        AIC: NA
> summary(res.qb04)

Call:
glm(formula = cbind(ura, (nage - ura)) ~ taion + takasa, family = quasibinomial,
    data = mydata01)

Deviance Residuals:
      Min        1Q    Median        3Q       Max
-1.88153  -0.38548   0.06386   0.81892   1.21023

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.8384     8.8308  -4.398  0.00173 **
taion         1.0008     0.2226   4.496  0.00150 **
takasa        0.1169     0.1594   0.734  0.48190
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasibinomial family taken to be 0.8690688)

    Null deviance: 30.8621  on 11  degrees of freedom
Residual deviance:  9.0836  on  9  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4

> anova(res.qb04,test="F")
Analysis of Deviance Table

Model: quasibinomial, link: logit
```

Response: cbind(ura, (nage - ura))

Terms added sequentially (first to last)

	Df	Deviance	Resid. Df	Resid. Dev	F	Pr(>F)
NULL			11	30.8621		
taion	1	21.3021	10	9.5600	24.5114	0.0007903 ***
takasa	1	0.4764	9	9.0836	0.5482	0.4779178

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
- -

交互作用のある例　交互作用およびその説明変数そのものは*で指定
交互作用だけは:で指定（こういう分析に意味があることはまれ）
- -

```
>
res.bi01<-glm(uraritu~taion*takasa,weight=nage,data=mydata01,family=binomial(link="logit"))
>
res.bi02<-glm(uraritu~taion+takasa+taion:takasa,weight=nage,data=mydata01,family=binomial(link="logit"))
>
res.bi03<-glm(uraritu~taion:takasa,weight=nage,data=mydata01,family=binomial(link="logit"))
> res.bi01

Call:  glm(formula = uraritu ~ taion * takasa, family = binomial(link =
"logit"),          data = mydata01, weights = nage)

Coefficients:
 (Intercept)            taion           takasa    taion:takasa
       7.81352         -0.26075         -3.49177          0.09762
```

Degrees of Freedom: 11 Total (i.e. Null); 8 Residual
Null Deviance: 30.86
Residual Deviance: 8.845 AIC: 44.4
> res.bi02

Call: glm(formula = uraritu ~ taion + takasa + taion:takasa, family = binomial(link = "logit"), data = mydata01, weights = nage)

Coefficients:
 (Intercept) taion takasa taion:takasa
  7.81352 -0.26075 -3.49177 0.09762

Degrees of Freedom: 11 Total (i.e. Null); 8 Residual
Null Deviance: 30.86
Residual Deviance: 8.845 AIC: 44.4
> res.bi03

Call: glm(formula = uraritu ~ taion:takasa, family = binomial(link = "logit"), data = mydata01, weights = nage)

Coefficients:
 (Intercept) taion:takasa
  -3.273467 0.006233

Degrees of Freedom: 11 Total (i.e. Null); 10 Residual
Null Deviance: 30.86
Residual Deviance: 28.44 AIC: 60
- -
定数項（切片）だけ
- -
>
res.b06<-glm(uraritu~1,weight=nage,data=mydata01,family=binomial(link="logit"))
> res.b06

Call: glm(formula = uraritu ~ 1, family = binomial(link = "logit"),

data = mydata01, weights = nage)

Coefficients:
(Intercept)
-0.2829

Degrees of Freedom: 11 Total (i.e. Null); 11 Residual
Null Deviance: 30.86
Residual Deviance: 30.86 AIC: 60.42
- -
定数項（切片）が 0
- -
>
res.b07<-glm(uraritu~taion-1,weight=nage,data=mydata01,family=binomial(link="logit"))

> res.b07

Call: glm(formula = uraritu ~ taion - 1, family = binomial(link = "logit"), data = mydata01, weights = nage)

Coefficients:
taion
-0.007073

Degrees of Freedom: 12 Total (i.e. Null); 11 Residual
Null Deviance: 33.26
Residual Deviance: 31.2 AIC: 60.76
- -

目的変数がポアソン分布でリンク関数が対数の場合（ポアソン回帰）
以下が使ったデータ
- -
> pdata1
x1 y1
1 11.50 0

```
2   11.00   1
3   10.20   5
4   11.90   3
5   12.40   4
6   13.60   6
7   17.10   7
8   11.15   2
9   10.50   1
10 21.20   8
11 12.30   5
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> res.p01<-glm(y1~x1,data=pdata1,family=poisson(link="log"))
> res.p01

Call:  glm(formula = y1 ~ x1, family = poisson(link = "log"), data = pdata1)

Coefficients:
(Intercept)                  x1
      -0.3082           0.1201

Degrees of Freedom: 10 Total (i.e. Null);  9 Residual
Null Deviance:          22.12
Residual Deviance: 13.4              AIC: 48.63
> summary(res.p01)

Call:
glm(formula = y1 ~ x1, family = poisson(link = "log"), data = pdata1)

Deviance Residuals:
       Min             1Q       Median             3Q          Max
-2.41860   -0.81926   -0.03940     0.71451     1.38837

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.3082        0.5753   -0.536   0.59216
x1                  0.1201        0.0380     3.161   0.00157 **
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 22.117  on 10  degrees of freedom
Residual deviance: 13.396  on  9  degrees of freedom
AIC: 48.631

Number of Fisher Scoring iterations: 5

> anova(res.p01,test="Chisq")
Analysis of Deviance Table

Model: poisson, link: log

Response: y1

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                    10     22.117
x1    1   8.7216         9     13.396  0.003145 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- -

family が poisson（ポアソン分布の意味）のときはデフォールトのリンクは log なので省略しても同じ

- -

```
> res.p02<-glm(y1~x1,data=pdata1,family=poisson)
> res.p02

Call:  glm(formula = y1 ~ x1, family = poisson, data = pdata1)
```

```
Coefficients:
(Intercept)          x1
     -0.3082      0.1201

Degrees of Freedom: 10 Total (i.e. Null);  9 Residual
Null Deviance:      22.12
Residual Deviance: 13.4          AIC: 48.63
```

- -

quasi-likelihood による overdispersion 対策（この例では overdispersion の程度はごく弱い）

- -

```
> res.qp01<-glm(y1~x1,data=pdata1,family=quasipoisson)
> res.qp01

Call:  glm(formula = y1 ~ x1, family = quasipoisson, data = pdata1)

Coefficients:
(Intercept)          x1
     -0.3082      0.1201

Degrees of Freedom: 10 Total (i.e. Null);  9 Residual
Null Deviance:      22.12
Residual Deviance: 13.4          AIC: NA
> summary(res.qp01)

Call:
glm(formula = y1 ~ x1, family = quasipoisson, data = pdata1)

Deviance Residuals:
      Min        1Q    Median        3Q       Max
-2.41860  -0.81926  -0.03940   0.71451   1.38837

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.30817    0.62761  -0.491   0.6352
x1           0.12012    0.04146   2.897   0.0177 *
```

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 1.190285)

    Null deviance: 22.117 on 10 degrees of freedom
Residual deviance: 13.396 on 9 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5

> anova(res.qp01,test="Chisq")
Analysis of Deviance Table

Model: quasipoisson, link: log

Response: y1

Terms added sequentially (first to last)

     Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL   10   22.117
x1   1   8.7216   9   13.396 0.006792 **

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

- -
オフセットの説明
　たとえば、c1 が行動が見られた回数で、tt1 が観察時間だとする。観察時間当たりの行動回数(c1/tt1)を目的変数として分析したいとする。
　リンクは対数リンクだとする。説明変数を x とすると、
log(c1/tt1)＝ β *x+ α
という回帰式を考えていることになる。左辺を変形する
log(c1) - log(tt1)＝ β *x+ α

となり、整理して
log(c1)＝ β*x+α+log(tt1)
となる。offset(変数名)では、回帰係数が 1 になる変数を指定するので、この場合、
c1~x+offset(tt1)
ではなく
c1~x+offset(log(tt1))
となる。
　なお、「観察時間の効果を除く」とか思って、
c1~x+ tt1
としてしまうと、
log(c1)＝ β*x+α+γ*tt1、変形して log(c1)- γ*tt1＝ β*x+α、整理して
log(c1/exp(γ*tt1))＝ β*x+α
となって、分析の目的とはだいぶ遠いところに行ってしまう。
- -
何かの 2 乗を説明変数に入れたいとき
- -

```
> res.nq01<-glm(gy1~I(gx1^2),family=gaussian)
> res.nq01

Call:  glm(formula = gy1 ~ I(gx1^2), family = gaussian)

Coefficients:
(Intercept)       I(gx1^2)
      2.86637         0.09771

Degrees of Freedom: 9 Total (i.e. Null);  8 Residual
Null Deviance:          65.68
Residual Deviance: 28.92            AIC: 45

##上は 2 次の項と定数項だけ、次は 1 次の項もある普通の 2 次式
> res.nq02<-glm(gy1~I(gx1^2)+gx1,family=gaussian)
> res.nq02

Call:  glm(formula = gy1 ~ I(gx1^2) + gx1, family = gaussian)

Coefficients:
```

```
(Intercept)        I(gx1^2)             gx1
     3.2965          0.1218         -0.2287

Degrees of Freedom: 9 Total (i.e. Null);  7 Residual
Null Deviance:          65.68
Residual Deviance: 28.8             AIC: 46.96
```

- -

R でグラフを描く

グラフィックス用の関数や命令には、それだけで図ができる高水準のものと、線を引く点を打つなどの低水準のものがある。ここでは、plot()を中心に、高水準のものの使い方の基本例を説明する。

関数 plot()による散布図の例（以下は 1 行ずつ実行して結果を確認する）

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> plot(y1~x1,data=pdata1)
##横軸の名前を変える
> plot(y1~x1,data=pdata1,xlab="shoumou")
##縦軸の名前も変える
> plot(y1~x1,data=pdata1,xlab="shoumou",ylab="No. of events")
##横軸の範囲を指定
> plot(y1~x1,data=pdata1,xlim=c(0,20))
##縦軸の範囲を指定
> plot(y1~x1,data=pdata1,ylim=c(0,20))
##縦軸の範囲と横軸の範囲を指定
> plot(y1~x1,data=pdata1,xlim=c(0,30),ylim=c(0,20))

##線の太さを変えて記号の見かけ上の大きさを変えてみる
> plot(y1~x1,data=pdata1,lwd=30)
##記号の色を変えてみる
> plot(y1~x1,data=pdata1,col="blue")
##記号の色と見かけ上の大きさを変えてみる
> plot(y1~x1,data=pdata1,lwd=20,col="blue")
##直線で結ぶ
> plot(y1~x1,data=pdata1,type="l")
##点を直線で結ぶ
> plot(y1~x1,data=pdata1,type="b")
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

- -

エラーバーをつけてみる

　gx1 の上下に er1 の長さのエラーバーを付ける

- -

```
> c1<-1:10
> c1
 [1]  1  2  3  4  5  6  7  8  9 10
> plot(gx1~c1)
> er1<-c(0.2,0.5,0.4,0.6,0.8,1.1,1.5,1.3,0.5,1.9)
> gx1u<-gx1+er1
> gx1l<-gx1-er1
> plot(gx1~c1)
> arrows(c1,gx1u,c1,gx1l,length=.05,angle=90,code=3)
```

- -

```
## 1 つの変数だけ指定すると、その変数が縦軸、順序を横軸にした散布図になる
> plot(pdata1$x1)
```

- -

```
##density の結果を入れると、カーネル密度のグラフになる
> plot(density(gy1))
```

- -

一般化線形関数モデルの関数 glm()の結果を plot()の中に入れると、残差プロットがいくつか次々にできる

```
#enter を押すと次が描かれる
> plot(res.p01)
```

- -

関数 barplot()で棒グラフを描く

- -

```
##まず使うデータ
> bardata01<-c(3,2.1,6.5,2,2.5)
> bardata01
[1] 3.0 2.1 6.5 2.0 2.5

> barplot(bardata01)

#棒の隙間をなくす
> barplot(bardata01,space=0)
```

#棒の隙間を広く

> barplot(bardata01,space=0.5)

#棒の中に斜線を引く

> barplot(bardata01,density=10,angle=5)

#棒の中の斜線の角度を変える

> barplot(bardata01,density=10,angle=45)

#棒の中に斜線を密に引く

> barplot(bardata01,density=50,angle=45)

#棒の太さをちがえる

> barplot(bardata01,width=c(2,1.5,2,1,1))

#棒が横向き

> barplot(bardata01,horiz=T)

#棒をちがう色でぬる

> barplot(bardata01,col = c("blue", "black", "cyan","green","brown"))

- -

箱ひげ図を boxplot()で描く

　箱ひげ図（box plot）は、箱でデータの中央値と上下のヒンジを示し、それよりも広い範囲を直線で示す

- -

> boxplot(gy1~gx2)

- -

ヒストグラムを hist()で描く

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#使用するデータ
> hdata1<-c(1,7.1,2,3,4,5,9,2,3,4,5,6,8,5.2,4,3,2,4,5,8,9,2)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> hist(hdata1)
#breaks で区切りの値を与えることができる
> hist(hdata1,breaks=c(0.5,3.5,6.5,9.5))
#細かくしてみる
> hist(hdata1,breaks=c(0.5,1.5,2.5,3.5,4.5,5.5,6.5,7.5,8.5,9.5))
#
> hist(hdata1,breaks=c(0.2,2.2,4.2,6.2,8.2,10.2))
#区間幅は同じでなくてもいいー意味があるかどうかは別だが
> hist(hdata1,breaks=c(0.2,2.9,4.2,6.2,8.2,10.2))
==============================
```