

## 統計モデリング入門 新潟大 2015 (5)

なんでも“割算”するな! — GLM の工夫

久保拓弥 [kubo@ees.hokudai.ac.jp](mailto:kubo@ees.hokudai.ac.jp), @KuboBook

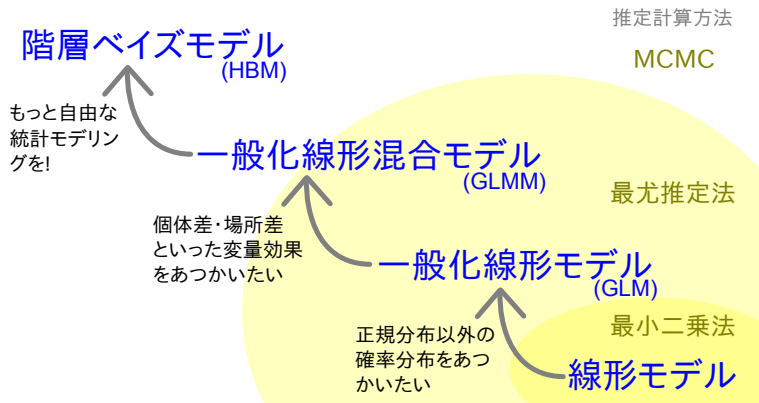
新潟大学集中講義 <http://goo.gl/m8HSBM>

2015-05-26

ファイル更新時刻: 2015-05-18 16:42

# この授業であつかう統計モデルたち

## 線形モデルの発展



データの特徴にあわせて線形モデルを改良・発展させる

一般化線形モデルって何だろう？

## 一般化線形モデル (GLM)

- ポアソン回帰 (Poisson regression)
- **ロジスティック回帰 (logistic regression)**
- 直線回帰 (linear regression)
- .....

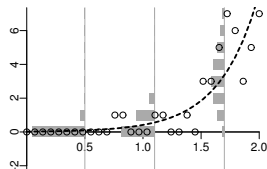
# 一般化線形モデルを作る

## 一般化線形モデル (GLM)

- 確率分布は?
- 線形予測子は?
- リンク関数は?

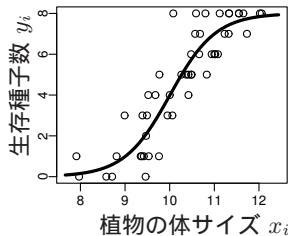
## ポアソン回帰のモデル

- 確率分布: **ポアソン分布**
- 線形予測子: e.g.,  $\beta_1 + \beta_2 x_i$
- リンク関数: **対数リンク関数**



## ロジスティック回帰のモデル

- 確率分布: 二項分布
- 線形予測子: e.g.,  $\beta_1 + \beta_2 x_i$
- リンク関数: logit リンク関数



# この時間に説明したいこと

- ① “ $N$  個のうち  $k$  個が生きてる” タイプのデータ  
上限のあるカウントデータ
- ② “ $N$  個のうち  $k$  個が生きてる” タイプのデータ  
 $y_i \in \{0, 1, 2, \dots, 8\}$
- ③ ロジスティック回帰の部品  
二項分布 binomial distribution と logit link function
- ④ 何でも「割算」するな!  
脱「割算」の offset 項わざ

# 1. “ $N$ 個のうち $k$ 個が生きてる” タイプのデータ

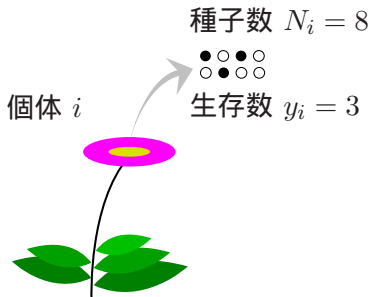
上限のあるカウントデータ

ポアソン分布ではなく二項分布で



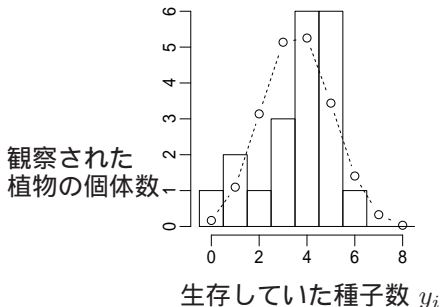
## 例題：植物の種子の生存確率

- 架空植物の種子の生存を調べた
- 種子: 生きていれば発芽する
  - どの個体でも **8 個** の種子を調べた
- 生存確率: ある種子が生きている確率
- データ: 植物 **20** 個体, 合計 **160** 種子の生存の有無を調べた
- 73 種子が生きていた — このデータを統計モデル化したい



たとえばこんなデータが得られたとしましょう

個体ごとの生存数	0	1	2	3	4	5	6	7	8
観察された個体数	1	2	1	3	6	6	1	0	0



これは個体差なしの均質な集団

## 生存確率 $q$ と二項分布の関係

- 生存確率を推定するために**二項分布** という確率分布を使う
- 個体  $i$  の  $N_i$  種子中  $y_i$  個が生存する確率

$$p(y_i | q) = \binom{N_i}{y_i} q^{y_i} (1 - q)^{N_i - y_i},$$

- ここで仮定していること
  - **個体差はない**
  - つまり **すべての個体で同じ生存確率  $q$**

ゆうど

尤度: 20 個体ぶんのデータが観察される確率

- 観察データ  $\{y_i\}$  が確定しているときに
- パラメータ  $q$  は値が自由にとりうると考える
- 尤度は 20 個体ぶんのデータが得られる確率の積, パラメータ  $q$  の関数として定義される

$$L(q|\{y_i\}) = \prod_{i=1}^{20} p(y_i | q)$$

---

個体ごとの生存数	0	1	2	3	4	5	6	7	8
観察された個体数	1	2	1	3	6	6	1	0	0

---

## 対数尤度方程式と最尤推定

- この尤度  $L(q \mid \text{データ})$  を最大化するパラメータの推定量  $\hat{q}$  を計算したい
- 尤度を対数尤度になおすと

$$\begin{aligned} \log L(q \mid \text{データ}) &= \sum_{i=1}^{20} \log \binom{N_i}{y_i} \\ &+ \sum_{i=1}^{20} \{y_i \log(q) + (N_i - y_i) \log(1 - q)\} \end{aligned}$$

- この対数尤度を最大化するように未知パラメーター  $q$  の値を決めてやるのが**最尤推定**

# 最尤推定 (MLE) とは何か

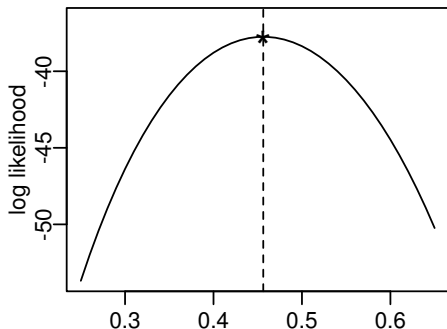
- 対数尤度  $L(q \mid \text{データ})$  が最大になるパラメーター  $q$  の値をさがすこと

- 対数尤度  $\log L(q \mid \text{データ})$  を  $q$  で偏微分して 0 となる  $\hat{q}$  が対数尤度最大

$$\partial \log L(q \mid \text{データ}) / \partial q = 0$$

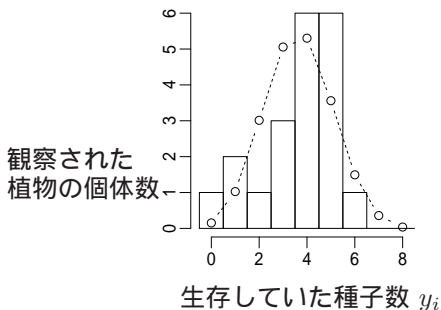
- 生存確率  $q$  が全個体共通の場合の最尤推定量・最尤推定値は

$$\hat{q} = \frac{\text{生存種子数}}{\text{調査種子数}} = \frac{73}{160} = 0.456 \text{ ぐらい}$$



## 二項分布で説明できる 8 種子中 $y_i$ 個の生存

$$\hat{q} = 0.46 \text{ なので } \binom{8}{y} 0.46^y 0.54^{8-y}$$



## 2. “ $N$ 個のうち $k$ 個が生きてる” タイプのデータ

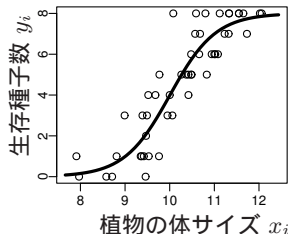
$$y_i \in \{0, 1, 2, \dots, 8\}$$



GLM のひとつである **logistic 回帰モデル** を指定する

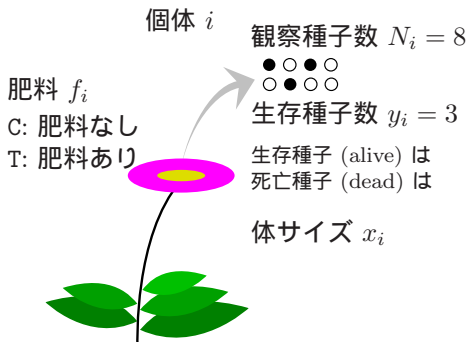
## ロジスティック回帰のモデル

- 確率分布: 二項分布
- 線形予測子: e.g.,  $\beta_1 + \beta_2 x_i$
- リンク関数: logit リンク関数



# またいつもの例題? ..... ちょっとちがう

8 個の種子のうち  $y$  個が **発芽可能** だった! ..... というデータ



# データファイルを読みこむ

data4a.csv は CSV (comma separated value) format file なので, R で読みこむには以下のようにする:

```
> d <- read.csv("data4a.csv")
```

or

```
> d <- read.csv(  
+ "http://hosho.ees.hokudai.ac.jp/~kubo/stat/2014/Fig/binomial/data4a.csv")
```

データは d と名付けられた data frame (表みたいなもの) に格納される

## data frame d を調べる

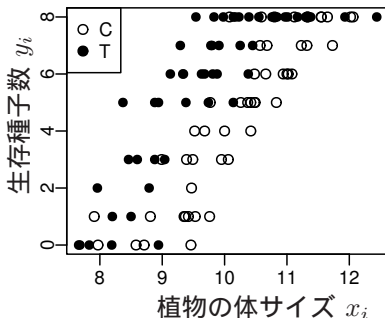
```
> summary(d)
```

	N	y	x	f
Min.	:8	Min. :0.00	Min. : 7.660	C:50
1st Qu.:	:8	1st Qu.:3.00	1st Qu.: 9.338	T:50
Median	:8	Median :6.00	Median : 9.965	
Mean	:8	Mean :5.08	Mean : 9.967	
3rd Qu.:	:8	3rd Qu.:8.00	3rd Qu.:10.770	
Max.	:8	Max. :8.00	Max. :12.440	

## まずはデータを図にしてみる

```
> plot(d$x, d$y, pch = c(21, 19)[d$f])
```

```
> legend("topleft", legend = c("C", "T"), pch = c(21, 19))
```



今回は施肥処理 がきいている？

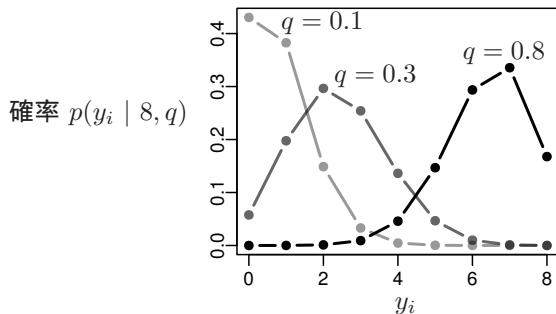
### 3. ロジスティック回帰の部品

二項分布 binomial distribution と logit link function

## 二項分布: $N$ 回のうち $y$ 回, となる確率

$$p(y | N, q) = \binom{N}{y} q^y (1 - q)^{N-y}$$

$\binom{N}{y}$  は  $N$  個の観察種子の中から  $y$  個の生存種子を選び出す場合の数

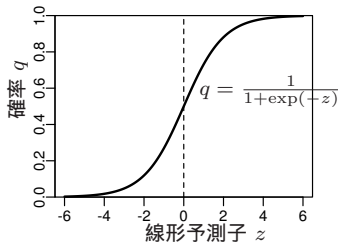


# ロジスティック曲線とはこういうもの

ロジスティック関数の関数形 ( $z_i$ : 線形予測子, e.g.  $z_i = \beta_1 + \beta_2 x_i$ )

$$q_i = \text{logistic}(z_i) = \frac{1}{1 + \exp(-z_i)}$$

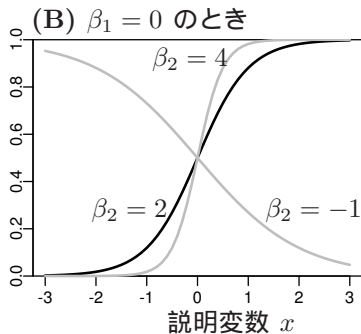
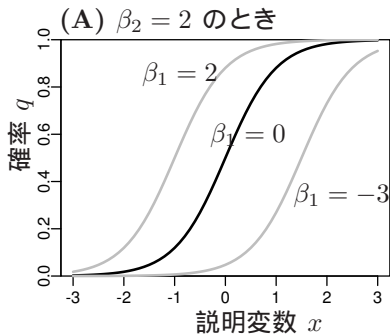
```
> logistic <- function(z) 1 / (1 + exp(-z)) # 関数の定義  
> z <- seq(-6, 6, 0.1)  
> plot(z, logistic(z), type = "l")
```





# パラメーターが変化すると.....

黒い曲線は  $\{\beta_1, \beta_2\} = \{0, 2\}$  . (A)  $\beta_2 = 2$  と固定して  $\beta_1$  を変化させた場合 .  
 (B)  $\beta_1 = 0$  と固定して  $\beta_2$  を変化させた場合 .



パラメーター  $\{\beta_1, \beta_2\}$  や説明変数  $x$  がどんな値をとっても確率  $q$  は  $0 \leq q \leq 1$   
 となる便利な関数

# logit link function

- logistic 関数

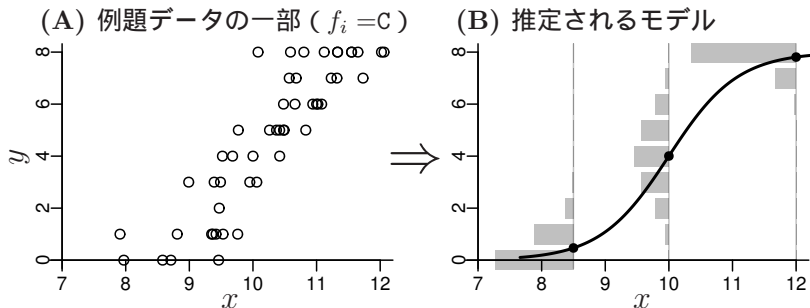
$$q = \frac{1}{1 + \exp(-(\beta_1 + \beta_2 x))} = \text{logistic}(\beta_1 + \beta_2 x)$$

- logit 変換

$$\text{logit}(q) = \log \frac{q}{1 - q} = \beta_1 + \beta_2 x$$

logit は logistic の逆関数 , logistic は logit の逆関数

logit is the inverse function of logistic function, vice versa

R でロジスティック回帰 —  $\beta_1$  と  $\beta_2$  の最尤推定

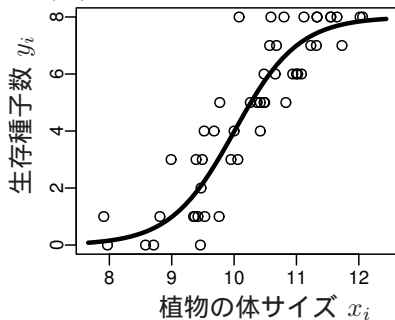
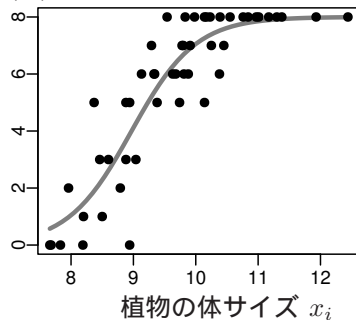
```
> glm(cbind(y, N - y) ~ x + f, data = d, family = binomial)
```

```
...
```

```
Coefficients:
```

(Intercept)	x	fT
-19.536	1.952	2.022

## 統計モデルの予測: 施肥処理によって応答が違う

(A) 施肥処理なし ( $f_i = C$ )(B) 施肥処理あり ( $f_i = T$ )

## 4. 何でも「割算」するな!

脱「割算」の offset 頂わざ

ポアソン回帰を強めてみる

## 割算値ひねくるデータ解析はなぜよくないのか?

- 観測値 / 観測値 がどんな確率分布にしたがうのか見とおしが悪く、さらに説明要因との対応づけが難しくなる
- 情報が失われる: 10 打数 3 安打 と 200 打数 60 安打, どちらも 3 割バッターと言ってよいのか?
- 割算値を使わないほうが見とおしのよい, 合理的なデータ解析ができる (今回の授業の主題)
- したがって割算値を使ったデータ解析は不利な点ばかり, そんなことをする必要はどこにもない

# 避けられるわりざん

- 避けられる割算値

- 確率

例:  $N$  個のうち  $k$  個にある事象が発生する確率

対策: ロジスティック回帰など二項分布モデルで

- 密度などの指数

例: 人口密度, specific leaf area (SLA) など

対策: offset 頂わざ — このあと解説!

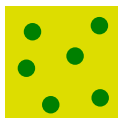
# 避けにくいわりざん

- 避けにくい割算値
  - 測定機器が内部で割算した値を出力する場合
  - 割算値で作図せざるをえない場合があるかも

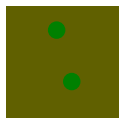


## offset 項の例題: 調査区画内の個体密度

- 何か架空の植物個体の密度が明るさ  $x$  に応じて どう変わるかを知りたい
- 明るさは  $\{0.1, 0.2, \dots, 1.0\}$  の 10 段階で観測した



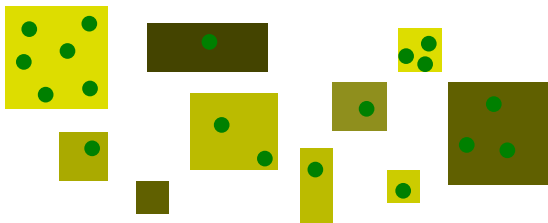
$x$ 大  
明るい



$x$ 小  
暗い

これだけなら単純に `glm(..., family = poisson)` とすればよいのだが .....

## 場所によって調査区の面積を変えました?!



- 明るさ  $x$  と面積  $A$  を同時に考慮する必要あり
- ただし密度 = 個体数 / 面積といった割算値解析はやらない!
- `glm()` の `offset` 頂わざでうまく対処できる
- ともあれその前に観測データを図にしてみる

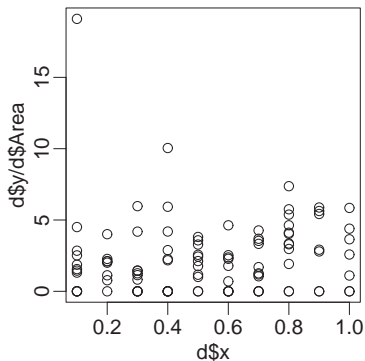
R の data.frame: 面積 Area, 明るさ x, 個体数 y

```
> load("d2.RData")
> head(d, 8) # 先頭 8 行の表示
```

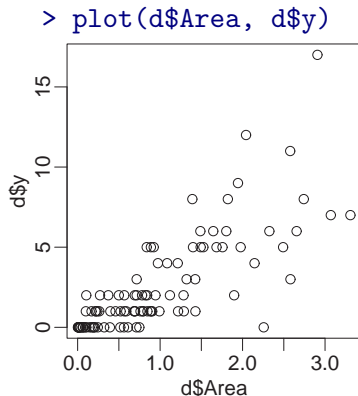
	Area	x	y
1	0.017249	0.5	0
2	1.217732	0.3	1
3	0.208422	0.4	0
4	2.256265	0.1	0
5	0.794061	0.7	1
6	0.396763	0.1	1
7	1.428059	0.6	1
8	0.791420	0.3	1

## 明るさ vs 割算値図の図

```
> plot(d$x, d$y / d$Area)
```



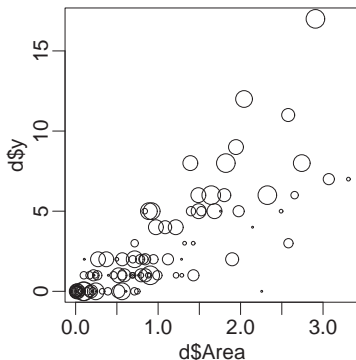
いまいちよくわからない

面積  $A$  vs 個体数  $y$  の図

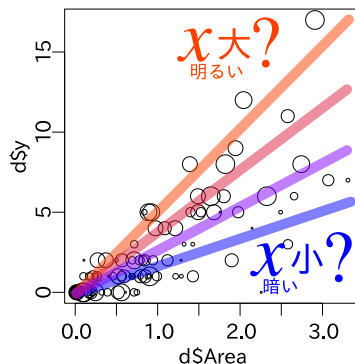
面積  $A$  とともに区画内の個体数  $y$  が増大するようだ

# 明るさ $x$ の情報 (マルの大きさ) も図に追加

```
> plot(d$Area, d$y, cex = d$x * 2)
```



同じ面積でも明るいほど個体数が多い?

密度が明るさ  $x$  に依存する統計モデル

- 区画内の個体数  $y$  の平均は面積  $\times$  密度
- 密度は明るさ  $x$  で変化する

# 平均個体数 = 面積 × 密度モデル

1. ある区画  $i$  の応答変数  $y_i$  は平均  $\lambda_i$  のポアソン分布にしたがうと仮定:

$$y_i \sim \text{Pois}(\lambda_i)$$

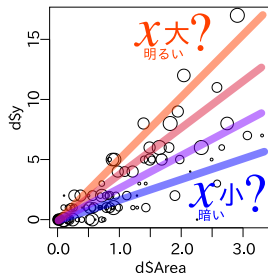
2. 平均値  $\lambda_i$  は面積  $A_i$  に比例し, 密度は明るさ  $x_i$  に依存する

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

つまり  $\lambda_i = \exp(\beta_1 + \beta_2 x_i + \log(A_i))$  となるので

$\log(\lambda_i) = \beta_1 + \beta_2 x_i + \log(A_i)$  線形予測子は右辺のようになる

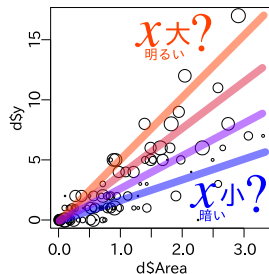
このとき  $\log(A_i)$  を offset 項とよぶ (係数  $\beta$  がない)





# この問題は GLM であつかえる!

- family: poisson, ポアソン分布
  - link 関数: "log"
  - モデル式:  $y \sim x$
  - offset 項の指定:  $\log(\text{Area})$
- **線形予測子**  $z = \beta_1 + \beta_2 x + \log(\text{Area})$   
 $a, b$  は推定すべきパラメーター
  - **応答変数の平均値**を  $\lambda$  とすると  $\log(\lambda) = z$   
 つまり  $\lambda = \exp(z) = \exp(\beta_1 + \beta_2 x + \log(\text{Area}))$
  - **応答変数** は平均  $\lambda$  のポアソン分布に従う:



# glm() 関数の指定

```
fit <- glm(  
  y ~ x,  
  family = poisson(link = "log"),  
  data = d,  
  offset = log(Area)  
)
```

結果を格納するオブジェクト

関数名

モデル式

確率分布の指定

offset の指定

リンク関数の指定 (省略可)

## R の glm() 関数による推定結果

```
> fit <- glm(y ~ x, family = poisson(link = "log"), data = d,  
  offset = log(Area))  
> print(summary(fit))
```

Call:

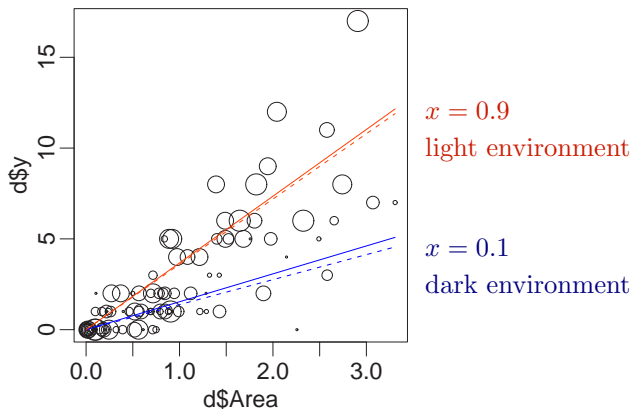
```
glm(formula = y ~ x, family = poisson(link = "log"), data = d,  
  offset = log(Area))
```

(... 略 ...)

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.321	0.160	2.01	0.044
x	1.090	0.227	4.80	1.6e-06

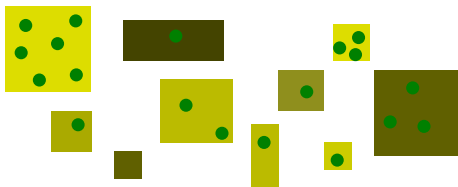
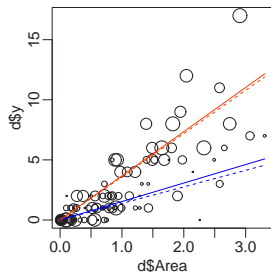
# 推定結果にもとづく予測を図にしてみる



- 実線は  $glm()$  の推定結果にもとづく予測
- 破線はデータ生成時に指定した関係

# まとめ: glm() の offset 頂わざで脱割算

- 平均値が面積などに比例する場合は, この面積などを **offset 項** として指定する
- 平均 = 面積 × 密度, というモデルの**密度** を exp(線形予測子) として定式化する



# 統計モデルを工夫してわりざんやめよう

- 避けられる割算値

- 確率

例:  $N$  個のうち  $k$  個にある事象が発生する確率

対策: ロジスティック回帰など二項分布モデルで

- 密度などの指数

例: 人口密度, specific leaf area (SLA) など

対策: offset 頂わざ — 統計モデリングの工夫!