用語の整理

- 一般線形モデル　General Linear Model
    - t検定や分散分析も線形モデルだよね
- 一般化線形モデル　Generalized Linear Model
    - 様々な分布を使えるモデリング

表 15.1 一般化線形モデルについて：予測変数 x の様々な尺度タイプに対する典型的な線形関数 lin(x)

予測変数 x の尺度タイプ

		量的		名義的	
単一の群	2 つの群	単一の予測変数	複数の予測変数	単一の因子	複数の因子
β_0	$\beta_{x=1}$ $\beta_{x=2}$	β_0 $+\beta_1 x$	β_0 $+\sum_k \beta_k x_k$ $+\sum_{j,k} \beta_{j\times k} x_j x_k$ $+\left[\begin{array}{c}\text{より高次の}\\\text{交互作用}\end{array}\right]$	β_0 $+\vec{\beta}\cdot\vec{x}$	β_0 $+\sum_k \vec{\beta}_k \cdot \vec{x}_k$ $+\sum_{j,k} \vec{\beta}_{j\times k} \cdot \vec{x}_{j\times k}$ $+\left[\begin{array}{c}\text{より高次の}\\\text{交互作用}\end{array}\right]$

値 $\mathrm{lin}(x)$ は表 15.2 に示されている関数によって被予測データに対してマッピングされる。

表 15.2 一般化線形モデルについて： 典型的なノイズ分布と被予測変数 y の様々な尺度タイプを記述するための逆リンク関数

被予測変数 y 尺度タイプ	典型的なノイズ分布 y ~ pdf ($\boldsymbol{\mu}$, [parameters])	典型的逆リンク関数 $\boldsymbol{\mu}$ = f (lin(x), [parameters])
量的	$y \sim$ 正規分布(μ, σ)	$\mu = \mathrm{lin}(x)$
二分値	$y \sim$ ベルヌーイ分布(μ)	$\mu =$ ロジスティック関数 $(\mathrm{lin}(x))$
名義	$y \sim$ カテゴリカル分布$(\ldots, \mu_k, \ldots)$	$\mu_k = \dfrac{\exp(\mathrm{lin}_k(x))}{\sum_c \exp(\mathrm{lin}_c(x))}$
順序	$y \sim$ カテゴリカル分布$(\ldots, \mu_k, \ldots)$	$\mu_k = \begin{array}{l} \Phi((\theta_k - \mathrm{lin}(x))/\sigma) \\ -\Phi((\theta_{k-1} - \mathrm{lin}(x))/\sigma) \end{array}$
カウント	$y \sim$ ポアソン分布(μ)	$\mu = \exp(\mathrm{lin}(x))$

値 μ は予測されたデータの中心傾向 (平均である必要はない)。予測変数は x，$\mathrm{lin}(x)$ は表 15.1 で示されているような x の線形関数。

ベルヌーイ回帰の例

- ロジスティック回帰だが分布の名前を冠するならベルヌーイ分布なので
- ベルヌーイ分布は試行回数が１の二項分布。コインの裏か表が出る確率，θ だと考えれば良い
- ただし，成功率 θ は0から1の範囲の数字

ベルヌーイ回帰の例

- ロジスティック関数で変換されたμがベルヌーイ分布に従う，という形でモデリングする

- ロジスティック関数

$$f(x) = \frac{1}{1 + exp(-x)}$$

- Stanにはinv_logit関数として実装されているのでそれを使ってコードを書く

#BSJ_S

Bonus Truck

```
data{
  int<lower=0> L;
  real X[L];
  int<lower=0> Y[L];
}

parameters{
  real beta;
}

transformed parameters{
  real<lower=0> theta[L];
  for(l in 1:L){
    theta[l] = inv_logit(X[l]*beta);
  }
}

model{
  for(l in 1:L){
    Y[l] ~ bernoulli(theta[l]);
  }
}
```

Bonus Truck

```
data{
  int<lower=0> L;
  real X[L];
  int<lower=0> Y[L];
}

parameters{
  real beta;
}


model{
  for(l in 1:L){
    Y[l] ~ bernoulli_logit(X[l]*beta);
  }
}
```

変換とセットになっている関数も

ポアソン回帰の例

- ポアソン分布
  - $Y_i \sim Poisson(\lambda)$
  - 平均パラメータλだけの分布
  - データが整数であること，パラメタが正の値しかとらないことに注意。

ポアソン回帰の例

- ポアソン分布
  - $Y_i \sim Poisson(\lambda)$
  - 平均パラメータλだけの分布
  - データが整数であること，パラメタが正の値しかとらないことに注意。

Bonus Truck

```
data{
  int<lower=0> L;
  real X[L];
  int<lower=0> Y[L];
}

parameters{
  real beta;
}

transformed parameters{
  real<lower=0> theta[L];
  for(l in 1:L){
    theta[l] = exp(X[l]*beta);
  }
}

model{
  for(l in 1:L){
    Y[l] ~ poisson(theta[l]);
  }
}
```

Bonus Truck

```
data{
  int<lower=0> L;
  real X[L];
  int<lower=0> Y[L];
}

parameters{
  real beta;
}


model{
  for(l in 1:L){
    Y[l] ~ poisson_log(X[l]*beta);
  }
}
```

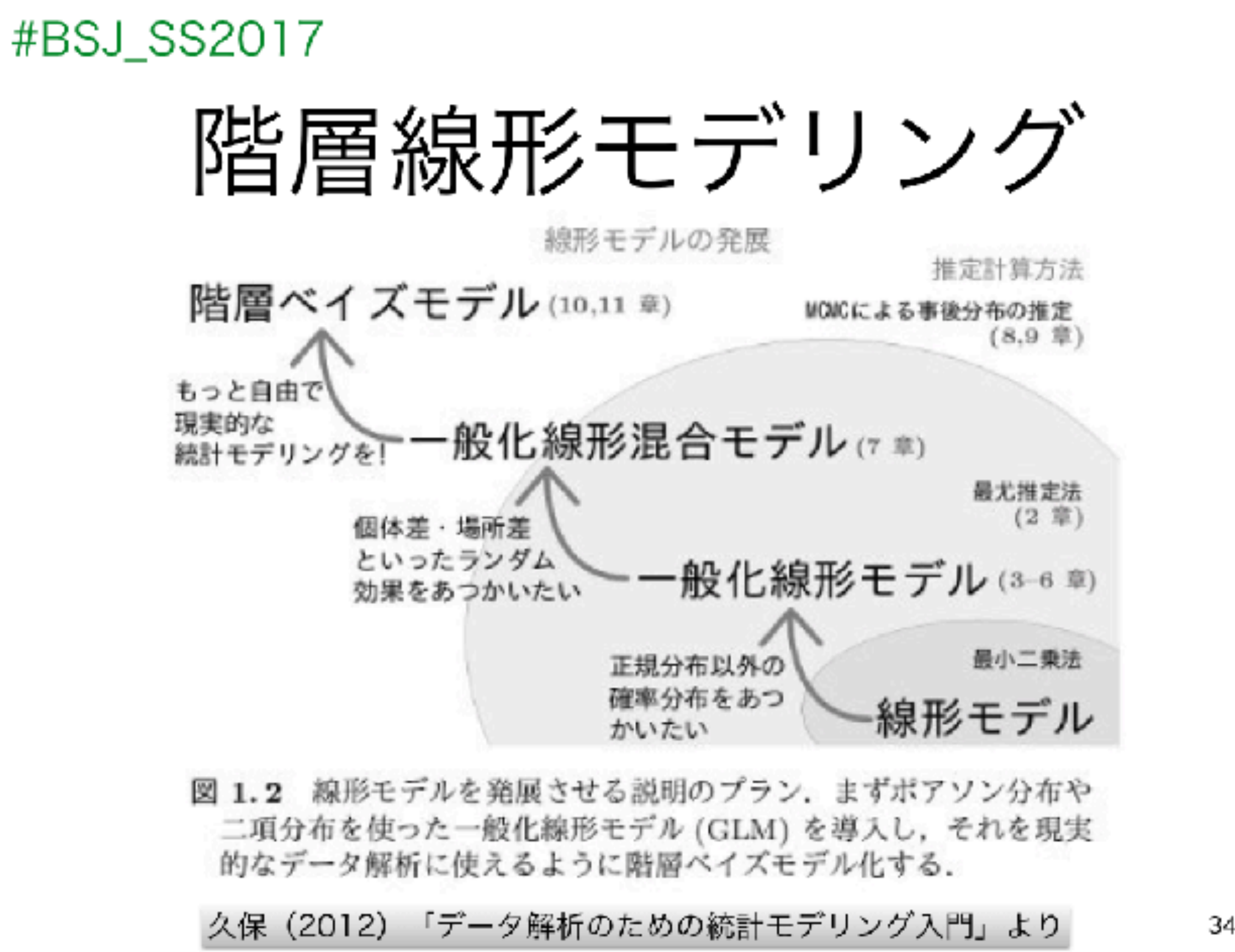
変換とセットになっている関数も

ちょっと追加

Bonus Truck

階層線形モデルの別解

- 階層線形モデルはベイズの特権ではない
- 従来の最尤法で複雑なモデルになると対応しきれないので，推定法としてベイズを利用するというながれもあった



階層線形モデルの別解

- 最尤法で推定するにはlmerパッケージをどうぞ
- ちゃんと米印も出してくれるlmerTestパッケージからの利用がオススメ

```
library(lmerTest)
result.HLM <- lmer(Y~X+(1|team),data=dat)
summary(result.HLM)
```

Bonus Truck

階層線形モデルの別解

- 最尤法で推定するにはlmerパッケージをどうぞ
- ちゃんと米印も出してくれるlmerTestパッケージからの利用がオススメ

```
library(lmerTest)
result.HLM <- lmer(Y~X+(1|team),data=dat)
summary(result.HLM)
```

切片のみ異なるモデル


```
Linear mixed model fit by REML t-tests use Satterthwaite ap
  degrees of freedom [lmerMod]
Formula: Y ~ X + (1 | team)
   Data: dat

REML criterion at convergence: 960

Scaled residuals: 
     Min       1Q   Median       3Q      Max 
-2.06753 -0.79008  0.04376  0.58145  2.41985 

Random effects:
 Groups   Name        Variance Std.Dev.
 team     (Intercept) 34497.7  185.74  
 Residual               119.1   10.91  
Number of obs: 120, groups:  team, 6

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)    
(Intercept) 753.94309   75.85388   5.09000   9.939  0.00016 ***
X             4.97348    0.01625 113.08000 306.117  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
  (Intr)
X -0.024
```

切片のみ異なるモデル

モデル式の記述

- モデル式の記述は次のようにする
  - 固定効果の独立変数と従属変数をチルダでつなぐ
  - 変量効果はカッコに入れて指定する

```
library(lmerTest)
result.HLM <- lmer(Y~X+(1|team),data=dat)
summary(result.HLM)
```

```
result.HLM2 <- lmer(Y~X+(1+X|team),data=dat)
summary(result.HLM2)
```

Bonus Truck

```
> summary(result.HLM2)
Linear mixed model fit by REML t-tests use Satt
  degrees of freedom [lmerMod]
Formula: Y ~ X + (1 + X | team)
   Data: dat

REML criterion at convergence: 989

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.17824 -0.59814 -0.01952  0.66576  2.13352

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 team     (Intercept) 21811.75 147.688
          X              62.21   7.888  0.68
 Residual               101.42  10.071
Number of obs: 120, groups:  team, 6

Fixed effects:
            Estimate Std. Error      df t value
(Intercept)  974.022     60.329   5.600  16.145
X             10.568      3.220   5.116   3.282
```

```
> fit8d
Inferen
4 chains, each with iter=50
post-warmup draws per chain

           mean se_mean
base0     975.47    1.09 67.
baseSig   156.51    0.96 58.
base[1]   809.68    0.06  4.
base[2]   859.15    0.06  4.
base[3]  1112.67    0.06  4.
base[4]   865.13    0.08  5.
base[5]  1041.70    0.08  5.
base[6]  1156.11    0.11  7.
beta0      10.62    0.06  3.
betaSig     8.61    0.05  3.
beta[1]     8.10    0.00  0.
beta[2]     3.34    0.00  0.
beta[3]    24.64    0.00  0.
beta[4]     3.94    0.00  0.
beta[5]    13.76    0.00  0.
beta[6]     9.63    0.00  0.
sig        10.20    0.01  0.
lp__     -385.78    0.06  3.
```

モデル式の記述

- Stanのチームが作ったrstanarmパッケージを使うと，lmerの書き方と同じモデルで変量効果モデルをベイズ推定できる

rstanarm-package {rstanarm} R Documentation

Applied Regression Modeling via RStan

Description

Stan Development Team

An appendage to the **rstan** package that enables some of the most common **applied** regression models to be estimated using Markov Chain Monte Carlo, variational approximations to the posterior distribution, or optimization. The **rstanarm** package allows these models to be specified using the customary R modeling