

# 心理統計学入門 II

## —尤度によるデータ生成過程の表現—

### 朝倉書店・副読本

早稲田大学 豊田秀樹

## 目次

1 単回帰モデル	2
2 重回帰モデル	10
3 偏回帰係数の解釈	12
4 ロジスティック回帰/メタ分析	17
5 ポアソンモデル	22
6 数種の分布による独立した1要因の推測	28
7 共分散分析/傾向スコア	35
8 さらに進んだ実験計画	41
9 階層線形モデル	47
10 間接質問法	53
11 項目反応理論	58
12 予測変数を直交化した重回帰分析	63
13 質的研究における飽和率・寡占度	65

# 1 単回帰モデル

## スクリプト

```
#####第1章 単回帰分析
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())           #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd を戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

##### 第1章で利用する関数 単回帰の初期値を最小2乗法で計算
init01<-function(y,x){    # y:基準変数, x:予測変数
  lmm<-lm(y ~x)           # 最小2乗法の解を格納する
  a<-as.numeric(coef(lmm)[1]) # 第1要素の切片の取り出し
  b<-as.numeric(coef(lmm)[2]) # 第2要素の回帰係数の取り出し
  sigma<-as.numeric(summary(lmm)$sigma) #  $\sigma$  e (1.5) 式の取り出し
  list(a=a,b=b, sigma=sigma) # リストでまとめて戻り値にする
}

##### 1.1 Galton (1886) の親子の身長データ

##### 表1.2 「身長データ」の一部 (cm)
親子身長 <- read.csv("./scrT/dat/親子身長.csv", header=T)
head(親子身長,n=14);tail(親子身長,n=14); #頭としっぽを見る

##### 表1.3 「身長データ」の数値要約
hsd<-function(x){sqrt(mean((x-mean(x))^2))}#標本分散による標準偏差を求める関数
round(apply(親子身長,2,mean),1)      # 平均
round(apply(親子身長,2,hsd),2)       # 標準偏差
round(apply(親子身長,2,quantile),1)  # 分位点

##### 図1.1 「身長データ」+ 乱数の散布図
set.seed(1234);                     #乱数の設定
R 親子身長<-親子身長+rnorm(928*2,0,0.8) #乱数の付加
plot(R 親子身長,type="p",xlim=c(155,190),ylim=c(155,190),pch=1,
     cex.axis=1.5,cex.lab=1.3)
#dev.copy2eps(file='z0101.eps',family='Japan1')

##### 図1.3 散布図と回帰直線 (図1.1に重ね書きする)
abline(0,1,lty=2,lwd=2.0)           #切片0, 傾き1の直線
abline(60.856,0.646,lty=1,lwd=2.0)  #点推定値による回帰直線を描く
#dev.copy2eps(file='z0103.eps',family='Japan1')

##### 単回帰モデルの stan コード
reg_01<-'
//単回帰モデル

data {                                //01
  int<lower=0> n;                      //02 データ数定義 p.2,1.5
  vector[n] y;                        //03 基準変数定義
  vector[n] X;                        //04 予測変数定義
}
parameters {                          //05
  real a;                             //06 切片定義 (1.3) 式
  real b;                             //07 回帰係数定義 (1.3) 式
  real<lower=0> sigma;                 //08 誤差標準偏差定義 (1.5) 式
}
transformed parameters {             //09
  vector[n] yhat;                     //10 予測値定義
  for(i in 1:n)                       //11 1 から n まで
    yhat[i] = a + X[i]*b;             //12 予測値 (1.2) 式
}
model {                               //13
  for(i in 1:n)                       //14 1 から n まで
    y[i] ~ normal(yhat[i],sigma);     //15 正規分布 (1.6) 式
}

par<-c("a","b","sigma")              #母数
dataSet_tankaiki01 <-list(n=nrow(親子身長),y=親子身長[,2], X=親子身長[,1] )

#####cmdstanr による実行
modfile_tankaiki01 <- write_stan_file(reg_01) #書き出す一時ファイル
mod_tankaiki01 <- cmdstan_model(modfile_tankaiki01) #コンパイル
csrfit_tankaiki01<-mod_tankaiki01$sample(data=dataSet_tankaiki01,chains=5,
  init = function(){init01(親子身長[,2], 親子身長[,1])},
  iter_sampling=20000,iter_warmup=1000,parallel_chains=5,seed=1234) #MCMC
fit_tankaiki01<-rstan::read_stan_csv(csrfit_tankaiki01$output_files())#stan 形式

#####rstan による実行
#fit_tankaiki01<-stan(model_code =reg_01,data=dataSet_tankaiki01,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit_tankaiki01,file="./scrT/obje/stan_obje0101")
#load(file="./scrT/obje/stan_obje0101"); #予め stan( ) で作った object をロード

ext_tankaiki01<-extract(fit_tankaiki01, par) #乱数取り出し

##### 表1.4 単回帰モデルの母数と、決定係数と相関係数の事後分布の要約 (前半)
gqcal(ext_tankaiki01$a)
gqcal(ext_tankaiki01$b)
gqcal(ext_tankaiki01$sigma)

hist(ext_tankaiki01$a,breaks=100) #切片の事後分布のヒストグラム (教科書にない)
hist(ext_tankaiki01$b,breaks=100) #回帰係数のヒストグラム (教科書にない)

#####関数を使った単回帰分析の実行
fit_tankaiki02<-Reg(親子身長[,2], 親子身長[,1],cha=5, ite=21000)
#save(fit_tankaiki02,file="./scrT/obje/stan_obje0102")
#load(file="./scrT/obje/stan_obje0102"); #予め stan( ) で作った object をロード
```

```
##### 表 1.4 単回帰モデルの母数と、決定係数と相関係数の事後分布の要約（全部）
out2<-print(fit_tankaiki02)
```

```
##### 1.4.2 「研究仮説が正しい確率」PHC
bp<-as.vector(fit_tankaiki02$b)
r2p<-as.vector(fit_tankaiki02$r2)
round(mean(bp >0.6),3); #phc(「回帰係数は 0.6 より大きい」) (1.21) 式 p.9,1.10
round(mean(r2p>0.2),3); #phc(「決定係数は 0.2 より大きい」) (1.21) 式 p.9,1.11
```

```
##### 表 1.5 回帰係数と決定係数の phc テーブル
phc01(seq(0.54,0.60,0.01),bp,cc="gtc",byoga="no")
phc01(seq(0.14,0.20,0.01),r2p,cc="gtc",byoga="no")
```

```
##### 図 1.4 回帰係数と決定係数の phc 曲線
par(mfrow=c(2,1)); #図 1.4
phc01(seq(0.5,0.75,0.01),bp,cc="gtc",byoga="yes",xlab="回帰係数")
phc01(seq(0.1,0.30,0.01),r2p,cc="gtc",byoga="yes",xlab="決定係数")
par(mfrow=c(1,1))
#dev.copy2eps(file="z0104qq.eps",family="Japan1")
```

```
##### 1.4.4 回帰効果
##### 表 1.6 予測値の事後分布の要約
out3<-print(fit_tankaiki02,degits=1,Xnew=seq(155,190,2))
```

```
##### 図 1.5 回帰直線・確信区間・予測区間
plot(R 親子身長,type="p",xlim=c(155,190),ylim=c(155,190),pch=1,
     cex.axis=1.5,cex.lab=1.3)
abline(60.856,0.646,lty=1,lwd=2.0)
lines(seq(155,190,2),out3$yhatc[, "0.025"],lty=2,lwd=2.0)
lines(seq(155,190,2),out3$yhatc[, "0.975"],lty=2,lwd=2.0)
lines(seq(155,190,2),out3$yastc[, "0.025"],lty=3,lwd=2.0)
lines(seq(155,190,2),out3$yastc[, "0.975"],lty=3,lwd=2.0)
#dev.copy2eps(file='z0104.eps',family='Japan1'); #図 1.5 ここまで
```

```
##### 表 1.7 基準変数の事後予測分布の要約
Xnew=seq(155,190,5)
out4<-print(fit_tankaiki02,degits=1,Xnew=Xnew)
```

```
##### 図 1.6 残差プロット（予測変数と残差の散布図）
plot(親子身長[,1],out2$resid,type="p",xlim=c(155,190),pch=1,
     cex.axis=1.5,cex.lab=1.3,xlab=' 両親',ylab=' 残差')
abline(h=0,lwd=2); #図 1.6 ここまで
#dev.copy2eps(file='z0105.eps',family='Japan1')
```

```
#####宿題
#「パスタ」データ入力
x1<- c(130,268,104,185,128,147,162, 68,142,175); #実測
x2<- c(110,232,176,207,122,202,191,124,193,250); #目測
x<-cbind(x1,x2)
```

## 1.1 基本方針

本書内に登場する統計モデルは stan というソフトウェアで分析されている。これまで R 言語を介して stan を利用する場合には、rstan というパッケージを使用するのが標準だった。しかし残念なことに、近年 rstan は重く、遅く、意味なく暴走するようになってしまった。この現状を鑑み、本書では cmdstanr というパッケージの使用をメインに据える。

**R 言語** オープンソース・フリーソフトウェアの統計解析向けのプログラミング言語及びその開発実行環境。ニュージーランドのオークランド大学の Ross Ihaka と Robert Clifford Gentleman により作られた。インタープリタ方式。公式サイトは <https://www.r-project.org/>

**stan** C++で書かれた統計的推論のための確率的プログラミング言語。Stan 言語では、対数確率密度関数を計算し、ベイジアン統計モデルを実装できる。アンドリュー・ゲルマンらによって 2012 年に開発され、モンテカルロ法の先駆者であるスタニスワフ・ウラムにちなんで名付けられた。コンパイル方式。公式サイトは <https://mc-stan.org/>

**cmdstanr** Stan への軽量な R インターフェイスを目指して開発された R のパッケージ。公式サイトは <https://mc-stan.org/cmdstanr/>

**rstan** Stan への R インターフェイスとして利用されてきた R のパッケージ。これは CRAN 上で rstan パッケージとして配布され公式サイトは <https://mc-stan.org/rstan/>

**インタープリタ方式** プログラミング言語の命令を 1 つずつ、機械語に解釈しながら実行する方式。作成したソースプログラムをただちに実行できるという点がこの方式のメリットである。一方、解釈しながら実行されるために、コンパイル型に比べると実行速度が遅いというのが欠点。

**コンパイル方式** ソースプログラムをいったん機械語に翻訳し、その機械語になったプログラムを実行する方式。翻訳することをコンパイル (compile)、翻訳するプログラムをコンパイラ (compiler) と言う。この方式のメリットは実行速度が速い点にある。デメリットとしては、コンパイルするという手間がかかることや、コンパイルした機械語のプログラムは他の環境 (OS や CPU が異なると) 実行できないことなどが挙げられる。

## 1.2 cmdstanr のインストール

筆者は windows10 に cmdstanr のインストールして使用している。mac へのインストールは未体験であるが同様に容易であるらしい (rstan のインストールの解説は割愛させていただく)。

### 1.2.1 ユーザー名が漢字の場合は、cmdstanr は動かない

ユーザー名がアルファベットの場合の、cmdstanr のインストールは、Getting started with CmdStanR (<https://mc-stan.org/cmdstanr/articles/cmdstanr.html>) の通りに何の問題もなくインストールされる。しかしユーザー名が漢字の場合は、cmdstanr は動かない。

### 1.2.2 漢字のユーザー名は変更しない

筆者の pc はユーザー名が漢字だったので、すこし手こずった。(1) まずユーザー名をアルファベットに変更した。でもフォルダ名は漢字のままなので、cmdstanr は動かない。(2) 次にユーザー名の漢字のフォルダ変更しようとしたが簡単には変更できない。(3) そしてレジストリを書き直してユーザー名の漢字のフォルダ名を書き換えることは、原理的には可能であることが分かった。しかし失敗するとシステム自体が不安定なる懸念がある。この方法はお勧めしない。

### 1.2.3 ユーザー名の追加をしよう

ではどうすればいいのか。漢字のユーザー名（フィルダ名）はそのままにして、新たにアルファベットの管理者権限のあるユーザー名を追加しよう。それでインストール問題は解決した。

### 1.2.4 便利なパッケージは使わないの？

cmdstanr や stan には、第 I 巻のスク립トに登場したようなパッラーがあります。たとえば brms パッケージは、線形式を書くだけで、それに相当する stan のスク립トを書いてくれる優れたツールです。しかしこの副読本では、brms のようなパッケージは使用しません。それは「尤度によるデータの生成過程の表

現」という学習目的のためには、1 つ 1 つ stan のスク립トを書いたほうが良いからです。もちろん便利なツールを否定しているのではありません。学習が進み、卒論や、データ分析の実践では、むしろ積極的に利用することをお勧めいたします。

## 1.3 以下の関数を自分で復習しよう

getwd, setwd, source, library, function, read.csv, head, tail, round, apply, mean, quantile, set.seed, rnorm, plot, abline, hist, as.vector, lines

## 1.4 stan を構成するブロック

**functions{ }** : 本ブロックでは自作関数を定義することができる。

**data{ }** : 変数宣言のみを行う。Stan 外部からの入力データについての変数を本ブロックに記述する。このブロックにはステートメントは記述しない。

**transformed data{ }** : サンプル実行中に値が変化しない変数（データ）を扱い、当該データを変形する。もし一部でも値が変化するならば、**transformed parameters{ }** : ブロックに記述する。

**parameters{ }** : 母数の変数宣言を行う。母数の変換に関するステートメントの記述は、次の **transformed parameters** ブロックに記述する。

**transformed parameters{ }** : 本ブロックでは母数変換のステートメントが評価される。サンプリングは行われない。サンプリングが行われるごとに本ブロックで記述したステートメントが実行される。

**model{ }** : モデル分布を記述する。

**generated quantities{ }** : 本ブロックで記述されたステートメントは MCMC が終わった後に計算される。生成量の計算に用いる。**transformed parameters** ブロックに置く必然性のないステートメントは、ここに置いたほうが計算効率がよい。

## 1.5 stan の基本ルール

- ある場所で宣言した変数のスコープは、後続するブロックすべてに渡る。反対に、変数宣言よりも先に当該変数の操作を記述することは許されない。たとえば `transformed data` において宣言、変換した変数は `model` ブロックで利用できる。一方、`generated quantities` ブロックで宣言した変数を、それ以前のブロックで用いることはできない。
- 各ブロックの中括弧内に具体的なコードを記述する。`model{ }` 以外の不必要なブロックは省略することもできるが、順番を入れ替えることはできない。必ず上記の順に記述しなければならない。
- Stan コードは、コード内で用いる変数の定義を行う変数宣言と、プログラムが処理を実行する内容を記述した文—ステートメント (statement) の記述から成り立っている。
- 「//」以降の当該行に記された文字列はすべてコメント文として扱われる。コメント文を除き、ステートメントは必ずブロック内にて記述する必要がある。コメント文はどこに置いてもよい。

## 1.6 単回帰モデルの stan コード

`data`, `parameters`, `transformed parameters`, `model` の4つのブロックのみがある。`functions`, `transformed data`, `generated quantities` の3つのブロックに関しては、のちの章で登場する。

### 1.6.1 data ブロック

`data` ブロックでは、分析に用いるデータに関係した変数定義を記述する。

//01 で `data` ブロックの開始を宣言している。//02 からは、変数定義を行っている。まず人数を表す変数を `n` とすることを宣言している。`n` は整数値をとるため、変数の型は `int` を指定。山括弧 `<>` は変数の定義域を宣言する際に用いられ、下限は `lower`、上限は `upper` で指定す。

```
data {                                     //01
  int<lower=0> n;                          //02 データ数定義 p.2,1.5
  vector[n] y;                            //03 基準変数定義
  vector[n] X;                            //04 予測変数定義
}
```

本例の場合、`n` は0未満の値をとらないため、変数の下限を `<lower=0>` と宣言している。なお変数定義の記述は「型 (範囲 [サイズ]) 変数名 ([サイズ]);」の順番に行う必要がある。

1つのステートメントの記述が終わった場合には、必ず文末に「;(セミicolon)」を記し、当該行におけるコードの記述が終了したことを宣言する。

//03, //04 では、各人のデータ値を表す基準変数 `y` と予測変数 `X` を定義している。`vector` は定義した変数が実数のベクトルであることを意味し、そのあとにある `x[数値]` はその次元である。`n` 人分の箱を用意している。

前のブロックで定義した変数は後のブロックでも用いることができたので、後続のブロックで同じ変数を再定義する必要はない。ただし、その逆は許されない。

### 1.6.2 parameters ブロック

本ブロックでは推定対象となる母数について変数宣言している。//05 で `parameters` ブロックの開始を宣言する。//06 と//07 と//08 は、切片・回帰係数・誤差標準偏差の変数定義を行っている。これらは実数値をとるため、変数の型は `real` を指定する。

```
parameters {                             //05
  real a;                                 //06 切片定義 (1.3) 式
  real b;                                 //07 回帰係数定義 (1.3) 式
  real<lower=0> sigma;                   //08 誤差標準偏差定義 (1.5) 式
}
```

### 1.6.3 transformed parameters ブロック

このブロックでは、`parameters` ブロックにおいて宣言した変数に対し、変換を施すステートメントを記述するためのブロックである。

```
transformed parameters {           //09
  vector[n] yhat;                  //10 予測値定義
  for(i in 1:n)                    //11 1 から n まで
    yhat[i] = a + X[i]*b;          //12 予測値 (1.2) 式
}
```

//09 で `transformed parameters` ブロックの開始を宣言している。//10 では、予測値  $\hat{y}$  の変数 `yhat` を定義している。予測値  $\hat{y}$  は観測対象ごとに計算されるから、次元数は `n` である。//11 続く //12 を、`i` を 1 から `n` まで動かして実行する（複数のステートメントを繰り返す場合には中括弧で囲む）。//11 と //12 で 1 つのステートメントなので、//11 にはセミコロンを打たない。//12 では、(1.2) 式により、予測値を計算している。

#### 1.6.4 model ブロック

`model` ブロックでは、具体的な統計モデルを記述する。また母数に関するサンプリングについても本ブロック内で記述する。

//13 で `model` ブロックの開始を宣言している。//15 では、データが (1.6) 式の正規分布に従っていることを記述している。

```
model {                             //13
  for(i in 1:n)                     //14 1 から n まで
    y[i] ~ normal(yhat[i],sigma);   //15 正規分布 (1.6) 式
}
```

データ全体の尤度は、 $n$  個ある //15 の総積（全部の掛け算）なのであるが、尤度が総積であることは言わずもがななので、掛け算は省略されている。

なお、Stan コード内においてサンプリング対象としつつも、明示的に事前分布を指定しなかった場合には、Stan によって一様分布が仮定される。スクリプト中に事前分布に関する記述がないのは、このためである。

本ブロックにおいて、前ブロックまでで記述していない変数を用いる場合には、新たに変数宣言を行う必要がある。ただし後に変数のサンプリング結果を出力するためには、(transformed) `parameters` ブロックに記述するか、もしくは生成量として `generated quantities` ブロックに記述する。

## 1.7 MCMC の実行

`stan` のスクリプトによって表現した統計モデルの母数等の事後分布を求めるための MCMC の実行方法を説明する。本節の流れは、基本的に後続する章でも共通しているので、しっかりと理解されたい。

### 1.7.1 R 側で乱数を使いたい母数・生成量の指定

`par<-c("a","b","sigma")` によって母数のベクトルを作っている。`stan` のスクリプト内の `parameters` ブロックの母数ばかりでなく、他のブロックで定義された生成量を指定する場合もある。要するに、事後分布や事後予測分布の乱数を R 側で使いたい母数・生成量を指定する。使わない母数は指定しないこともある。

### 1.7.2 サイズやデータの指定

`dataSet_tankaiki01 <-list(n=nrow(親子身長),y=親子身長[,2], X=親子身長[,1])` によってサイズや入力データをリストの形式で指定する。ここは `data` ブロックと、順番・内容を完全に一致させなければならない。具体的に、ここでは、データ数・基準変数・予測変数を、この順番で過不足なく指定しなくてはいけない。

### 1.7.3 cmdstanr による MCMC の実行

```
ファイルへのパス <- write_stan_file(コード)
コンパイル結果   <- cmdstan_model(ファイルへのパス)
cmdstanr 形式の MCMC 結果 <- コンパイル結果$sample(
  data      = サイズやデータで作成したリスト,
  chains    = チェーン数
  init      = function(){parameters ブロックの順番で初期値指定 (リスト形式)},
              (ここでは init_i01( ) という新たに定義し関数を利用している)
  iter_sampling = 1 つのチェーンから取り出す乱数、ウォームアップは除く,
  iter_warmup   = 1 つのチェーンのウォームアップ数,
  parallel_chains = 平行に走らせる 5, チェーン数, seed = 乱数の種)
```

「`cmdstanr` 形式の MCMC 結果」には多少の癖があるので、`rstan::read_stan_csv(cmdstanr 形式の MCMC 結果$output_files())` によって「`rstan` 形式の MCMC 結果」に変換して利用する。

### 1.7.4 rstan による MCMC の実行

```
rstan 形式の MCMC 結果 <- stan(  
  model_code = コード,  
  data       = サイズやデータで作成したリスト,  
  pars       = R 側で乱数を使いたい母数・生成量,  
  seed       = 乱数の種,  
  chains     = チェーン数,  
  warmup     = 1 つのチェーンのウォームアップ数,  
  iter       = 1 つのチェーンから取り出す乱数、ウォームアップ含む
```

cmdstanr ではコンパイルとサンプリングを分けていたが、rstan では一緒に実行している。cmdstanr による MCMC の実行をメインに据えるので、通常は、この部分はコメントアウトしてある。

### 1.7.5 あらかじめ実行した MCMC の結果をロードする

cmdstanr にしても、rstan にしても、コンパイルと MCMC には、それなりの時間がかかる。授業等の解説では、MCMC の待ち時間を節約したいときも少なくない。そのような場合には、以下の命令で、授業前に、あらかじめ実行した MCMC の結果を保存しておくといよい。

```
save(保存するオブジェクト,file="./scrT/obje/ファイル名")
```

授業中には、以下の命令で、一瞬で保存したオブジェクトを読み込むことができる。load 関数を実行することで、MCMC をせずにスムーズに授業をすることができる。

```
load(file="./scrT/obje/ファイル名")
```

通常は以下のようにコメントアウトしてあるから、必要に応じて利用していただきたい。

```
#save(fit_tankaiki01,file="./scrT/obje/stan_obje0101")  
#load(file="./scrT/obje/stan_obje0101")
```

この save と load の組み合わせは、ここばかりでなく、本書に登場するすべての MCMC の計算に、コメントアウトの形式で用意されている。

### 1.7.6 オブジェクトから乱数を取り出す

extract(rstan 形式の MCMC 結果, par) によって MCMC による乱数のみがリスト形式で格納されたオブジェクトを得る。

## 1.8 自習問題

1. スクリプトを崩して、どんなエラーがでるか確かめてみよう。よく出るエラーメッセージを知っておくことは、自分でスクリプトを書いたとき、バグをとるために大切である。
  - (a) R はセミコロンはあってもなくてもよい。stan はどうだろう。どこでもいいからセミコロンをとってみよう。
  - (b) 逆に、//11 にセミコロンをつけてみよう。
  - (c) R では変数定義は必要ない。//02 の変数定義をやめてみよう。
  - (d) R では整数と実数を区別しなくていい。//02 の int<lower=0> だけ取り除いてみよう。
2. transformed parameters ブロックをなくして、同等の結果を得るには、どう書き換えたらよいだろうか。stan で同じモデルを表現する場合にも、いろいろな書き方がある。

## 1.9 筆者が作った関数

### 1.9.1 gqcal: 生成量・母数・予測分布の要約統計量

```
gqcal(x,digits =3,probs=c(0.025,0.05,0.5,0.95,0.975))
```

### 引数

**x** (乱数× 1) の生成量ベクトル、または (乱数×生成量) の行列  
**digits** 整数、出力される小数点以下の桁数  
**probs** 数値ベクトル、報告するパーセンタイル

### 戻り値

**y** c(EAP, post.sd, 分位点)

## 1.9.2 Reg: 関数を使った回帰分析の実行

```
Reg(y,X,see=1234, cha=5, war=1000, ite=21000, pack="cmds")
```

### 引数

**y** 基準変数 (n)

**X** 予測変数行列 (n\*p)

**see, cha, war, ite** 乱数の種/チェーン数/ウォームアップ期間/生成数 M/

**pack** パッケージの指定。"cmds"は cmdstanr で規定値。"rstan"で rstan。

### 戻り値

**fit** stan の出力

**par** 母数リスト

**a** 切片 (乱数)

**b** 回帰係数ベクトル (p) (乱数)

**sigma** 誤差標準偏差

**r2** 決定係数

**r** 重相関係数

**sb** 標準偏回帰係数

**vyhat** yhat の分散

**log\_lik** 対数尤度

### 印刷のメソッド

```
print.Reg(x,degits=3, prob=c(0.025, 0.05, 0.5, 0.95, 0.975),Xnew=F)
```

**x** クラス 'Reg' のオブジェクト

**degits** 小数点以下の数字

**prob** 事後分布で報告する確率点

**Xnew** (n\*P) の行列、単回帰の場合はサイズ n のベクトル 予測分布用データ

### 出力

**fit** stan の出力

**waic,y,X** waic,y,X

**yhat** 予測値

**resi** 残差 (X による)

**Gc** 標準偏回帰係数

**yhata yhatc** 回帰式の事後分布 (Xnew による)

**yasta yastc** 予測分布 (Xnew による)

## 1.9.3 phc01: phc 曲線の描画, phc のテーブル

```
phc01(seq01, a, b=0, cc="gtc", byoga="yes", dedits=3, lwd=2, lty=1,  
      xlab="", ylab="", cex.lab=2.2, cex.axis=1.5)
```

### 引数

**seq01** 基準点 c (曲線の横軸) を表現した等差数列 (またはスカラー)

**a** 数値ベクトル、母数または生成量または予測分布の乱数

**b** ゼロまたは数値ベクトル、母数または生成量または予測分布の乱数

**cc** "gtc"の場合は  $a - b > c$  であり、"rope"の場合は  $abs(a - b) < c$  であり、  
それ以外は ("ltc"が想定されるが)  $a - b < c$

**byoga** "yes"の場合は phc 曲線を描き phc は出力しない、"no"等それ以外の場合は曲線は描画せずに phc を出力する

**dedits** 整数、出力される小数点以下の桁数

**lwd, lty, xlab, ylab, cex.lab, cex.axis** 通常のグラフィックパラメタ

### 戻り値

**av** phc のテーブル

## 1.10 phc の計算

phc は、それを計算する関数 phc01( ) 係でなく、1 点であれば、

**mean(MCMC で得られた乱数ベクトルで研究仮説を表現した論理式)**

によって計算できる。研究仮説を表現する論理式は True または False を返す。  
ただし True は 1 として扱われ、False は 0 として扱われるから、平均値を計算すれば、乱数ベクトル中の True の比率に一致し、phc が求まる。たとえば、  
phc(「回帰係数は 0.6 より大きい」)、(1.21) 式 p.9,l.10 ならば

**mean(bp >0.6)**

とする。



## 1.11 第 1 章 宿題

「統計学入門 II ー尤度によるデータ生成過程の表現ー」

P.15, 1.6 節 実習課題 をしなさい。

夏学期にデータを取った人はそれを使用してください。

## 2 重回帰モデル

### 2.1 スクリプト

```
##### 第2章 重回帰モデル
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
library(psych)
library(rgl)
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

##### 2.1 直腸がんデータの分析
##### 表 2.1 食物供給量と直腸がんの訂正死亡率
(can <- read.csv("./scrT/dat/直腸がん.csv",header=T))

##### 表 2.2 「直腸がんデータ」の要約統計量
(平均<-apply(can,2,"mean"));          #表 2.2
vari<-function(x){mean((x-mean(x))^2)} #標本分散を計算する関数の定義
(SD<-sqrt(apply(can,2,"vari")))        #標準偏差

##### 図 2.1 多変量散布図
pairs.panels(can,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)

##### 表 2.3 「直腸がんデータ」の共分散行列
round(var(can),0)

##### 表 2.4 「直腸がんデータ」の相関行列
round(cor(can),2)

##### 2.2 重回帰モデル

#####関数を使った重回帰分析の実行
out1<-Reg(can[,3],can[,c(2,1)]/100)
#save(out1,file="./scrT/obje/stan_obje0201")
#load(file="./scrT/obje/stan_obje0201"); #予め stan( ) で作った object をロード

##### 表 2.5 事後分布の要約
pri1<-print(out1,3)

##### 図 2.2 決定係数の事後分布 (左) と
##### 「乳製品」の標準偏回帰係数の事後分布 (右)
par(mfrow=c(1,2))
hist(out1$r2,breaks=50,cex.axis=2.0,cex.lab=2.0,main="",
```

```
      xlab="決定係数",ylab="")
hist(out1$sb[,1],breaks=50,cex.axis=2.0,cex.lab=2.0,main="",
      xlab="「乳製品」の標準偏回帰係数",ylab="")
par(mfrow=c(1,1))
#dev.copy2eps(file="z0202sw.eps",family="Japan1")
```

```
##### 図 2.3 「乳製品」の ROPE と「総熱量」の phc 曲線
par(mfrow=c(2,1))
phc01(seq(0,0.5,0.01),out1$sb[,1],cc="rope",byoga="yes",
      xlab="「乳製品」の標準偏回帰係数の ROPE")
phc01(seq(0.65,1.2,0.01),out1$sb[,2],cc="gtc",byoga="yes",
      xlab="「総熱量」の標準偏回帰係数")
par(mfrow=c(1,1))
#dev.copy2eps(file="z0203sw.eps",family="Japan1")
```

```
##### 表 2.6 回帰係数と決定係数の phc テーブル
phc01(seq(0,0.3,0.05),out1$sb[,1],cc="rope",byoga="no")
phc01(seq(0.68,0.8,0.02),out1$sb[,2],cc="gtc",byoga="no")
```

```
##### 図 2.4 3 次元散布図と回帰平面 (単位: kcal)
plot3d(can[, "総熱量"],can[, "乳製品"],can[, "直腸がん"],size=7,
      lwd=1.5,type="h",xlab="", ylab="", zlab="");
planes3d(a=0.00594,b=-0.00403,c=-1,d=-10.97805, alpha = 0.1)
#rgl.snapshot("z0202.png")
```

```
##### 図 2.5 残差プロット (予測値  $\hat{y}$  と残差  $e$  の散布図)
plot(pri1$yhat,pri1$resi,type="n",xlab="予測値",ylab="residual",
      cex.axis=2.0,cex.lab=2.0)
abline(h=0,lwd=2)
text(pri1$yhat,pri1$resi,rownames(can),cex=1.5)
#dev.copy2eps(file="z0205ss.eps",family="Japan1")
```

```
##### 表 2.7 予測値・事後標準偏差・事後分布 95%点・事後予測分布 95%点
Xnew<-matrix(c(240,2800,140,2700,340,2900),,2,T);#乳製品・総熱量の順番
XnewS<-Xnew/100;          #推定した時の状態に合わせて 100 で割っている。
pri3<-print(out1,Xnew=XnewS)
(tab1406<-data.frame(
  乳製品=Xnew[,1],
  総熱量=Xnew[,2],
  予測値=round(pri3$yhatc[, "EAP"],1),
  postsd=round(pri3$yhatc[, "post.sd"],1),
  # 確信 025=round(pri3$yhatc[, "0.025"],1),
  確信 95=round(pri3$yhatc[, "0.95"],1),
  sd=round(pri3$yastc[, "sd"],1),
  # 予測 025=round(pri3$yastc[, "0.025"],1),
  予測 95=round(pri3$yastc[, "0.95"],1)))
```

## 2.2 以下の関数を自分で復習しよう

`cor`, `sqrt`, `text`,

## 2.3 作図関数

### 2.3.1 `pairs.panels`: 多変量散布図を描く

```
pairs.panels(x, smooth = TRUE, density=TRUE, ellipses=TRUE, rug=TRUE)
```

#### 引数

**x** データフレームか行列

**smooth** 論理値 傾向線を引く

**density** 論理値 ヒストグラムに密度を描く

**ellipses** 論理値 確率楕円を描く

**rug** 論理値 ヒストグラムに rug を描く

### 2.3.2 `plot3d`: 3次元散布図を描く

```
plot3d(x, y, z, type = "p" )
```

#### 引数

**x, y, z**, プロットされる x 軸, y 軸, z 軸の座標ベクトル

**type** 'p' はポイント、'h' は z=0 の面から足を立てる

### 2.3.3 `planes3d` : 3次元空間に平面を描く

```
planes3d(a,b,c,d, alpha )
```

#### 引数

**a,b,c,d**  $ax + by + cz + d = 0$

**alpha** グレイスケール 0.0 は白、1.0 は黒

## 2.4 自習問題

1. 「総熱量」の標準偏回帰係数の事後分布のヒストグラムを描きなさい。
2. 重相関係数の ROPE の phc 曲線を区間 (0.0, 1.0) で描きなさい。
3. 重相関係数の ROPE の phc テーブルを `seq(0.6,0.9,0.05)` で作成しなさい。

## 2.5 第2章 宿題

夏学期に2要因実験の学習をした際に作成した錯視データの重回帰分析を以下の形式で実行し解釈しなさい。

基準変数：主観的等価点 (PSE)、予測変数 x1：斜線の長さ、予測変数 x2：斜線の角度、データ数 n：192 (32 × 6 セル)

1. 表 2.4 に相当する相関行列、特に予測変数間の相関係数を解釈せよ。
2. 表 2.5 の形式で、決定係数  $\eta^2$ 、2つの標準偏回帰係数  $b1^*$ ,  $b2^*$  の事後分布を示せ。1. で求めた基準変数と予測変数の相関と、ここで求めた標準偏回帰係数の関係を論じなさい。
3. 2. で求めた決定係数の rope の phc テーブルを示しなさい。そして「長さと角度で錯視現象を 1 % (0.01) 以下しか説明できないなら、自分にとって錯視など存在しない」という言明が正しい確率を求めなさい。

## 3 偏回帰係数の解釈

### 3.1 スクリプト

```
#####第3章 偏回帰係数の解釈
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd()) #working directory(wd) の確認
(setwd('./scrT')) #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd)) #wd 戻す
library(rstan) #パッケージ rstan の呼び出し
library(cmdstanr) #パッケージ cmdstanr の呼び出し
library(posterior) #パッケージ posterior の呼び出し
library(psych)
library(rgl)
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義
par<-c("bs", "bs12", "r2", "r", "indi", "tota");#母数は共通なのでこの章のみ先に宣言
```

```
##### 第3章で利用する関数ここから
###重回帰と単回帰の初期値
initf<-function(y,x1,x2){
  lm2<-lm(y ~x1+x2)
  lm1<-lm(x2~x1)
  ay<-as.numeric(coef(lm2)[1])
  a2<-as.numeric(coef(lm1)[1])
  b<-as.vector(coef(lm2)[c(2,3)])
  b12<- as.numeric(coef(lm1)[2])
  sigma_y<-as.numeric(summary(lm2)$sigma)
  sigma_2<-as.numeric(summary(lm1)$sigma)
  list(ay=ay,a2=a2,b=b, b12=b12,sigma_y=sigma_y, sigma_2=sigma_2)
}
###プリント関数
gqcalc3<-function(ext){
  print(' 偏回帰係数 y2');print(gqcal(ext$bs[,2]))
  print(' 偏回帰係数 21');print(gqcal(ext$bs12))
  print(' 直接効果 y1 ');print(gqcal(ext$bs[,1]))
  print(' 間接効果 ');print(gqcal(ext$indi))
  print(' 総合効果 ');print(gqcal(ext$tota))
  print(' 決定係数 ');print(gqcal(ext$r2))
}
##### 第3章で利用する関数ここまで
```

```
#####重回帰と単回帰モデルの stan コード
mu_si_reg<-
data {
  int<lower=0> n; //データ数
  vector[n] y; //基準変数
  matrix[n,2] X; //予測変数行列
}
transformed data {
  real sdy; //y の標準偏差
```

```
vector[2] sdX; //x1,x2 の標準偏差
sdy = sqrt((variance(y)*(n-1))/n); //01 y の標準偏差
sdX[1] = sqrt((variance(X[,1])*(n-1))/n); //02 x1 の標準偏差
sdX[2] = sqrt((variance(X[,2])*(n-1))/n); //03 x2 の標準偏差
}
parameters {
  real ay; //切片 (重回帰)
  real a2; //切片 (単回帰)
  vector[2] b; //偏回帰係数ベクトル
  real b12; //偏回帰係数
  real<lower=0> sigma_y; //誤差標準偏差 (重回帰)
  real<lower=0> sigma_2; //誤差標準偏差 (単回帰)
}
transformed parameters {
  vector[n] yhat; //yhat
  vector[n] x2hat; //x2hat
  yhat = ay + X*b; //04 重回帰式 (3.4) 式
  x2hat= a2 + X[,1]*b12; //05 単回帰式 (3.5) 式
}
model {
  y ~ normal(yhat, sigma_y); //06 重回帰モデル (2.3) 式
  X[,2]~ normal(x2hat, sigma_2); //07 単回帰モデル (1.6) 式
}
generated quantities{
  vector[2] bs;
  real bs12;
  real<lower=0> vyhat;
  real<lower=0,upper=1> r2;
  real<lower=0,upper=1> r;
  real indi;
  real tota;
  vyhat =(variance(yhat)*(n-1))/n; //08 yhat の分散
  r2 = vyhat/(vyhat+sigma_y^2); //09 決定係数 (1.15) 式, (2.8) 式
  r =sqrt(r2); //10 重相関係数 (2.9) 式
  bs[1] = (b[1]*sdX[1])/sdy; //11 標準偏回帰係数 1 (2.7) 式
  bs[2] = (b[2]*sdX[2])/sdy; //12 標準偏回帰係数 2 (2.7) 式
  bs12 = (b12*sdX[1])/sdX[2]; //13 標準回帰係数 (2.7) 式
  indi = bs12*bs[2]; //14 間接効果 (3.6) 式
  tota = indi+bs[1]; //15 総合効果 (3.6) 式
}
';
```

##### 3. 1 予測変数が多い場合の偏回帰係数の解釈の困難性

##### 3. 2 直接効果・間接効果・総合効果

```
##### 2. 1 直腸がんデータの分析 (第2章からの接続のため分類 (3) が先頭)
(can <- read.csv("../scrT/dat/直腸がん.csv",header=T))
dataSet03<-list(n=nrow(can),y=can[,3], X=can[,c(2,1)] ); #入力
ini<-function(){initf(y=can[,3],x1=can[,2],x2=can[,1])}; #初期値
```

```
#####cmdstanr による実行
modfile03 <- write_stan_file(mu_si_reg) #書き出す一時ファイル
```

```

mod03 <- cmdstan_model(modfile03)          #コンパイル
csrfit03 <- mod03$sample(data = dataSet03,chains = 5,iter_sampling = 20000,
  init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit03 <- rstan::read_stan_csv(csrfit03$output_files()) #stan 形式への変換

#####rstan による実行
#fit03<-stan(model_code =mu_si_reg,data=dataSetequ03,
#  init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit03, file="./scrT/obje/stan_obje0303")
#load(file="./scrT/obje/stan_obje0303"); #予め stan( ) で作った object をロード

ext03<-extract(fit03, par);                #乱数の取り出し
gqcalc3(ext=ext03);                        #推定値 p.30,b1.9

##### 3.3.1 (1) 抑制変数が存在する場合
d01 生物 <- read.csv("./scrT/dat/ch02/d01 生物_raw.csv",header=T)
##### 図 3.4 多変量散布図 (1)
pairs.panels(d01 生物,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet01<-list(n=nrow(d01 生物),y=d01 生物 [,3], X=d01 生物 [,c(1,2)] )#入力
ini<-function(){init(y=d01 生物 [,3],x1=d01 生物 [,1],x2=d01 生物 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit01 <- mod03$sample(data = dataSet01,chains = 5,iter_sampling = 20000,
  init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit01 <- rstan::read_stan_csv(csrfit01$output_files()) #stan 形式への変換
#####rstan による実行
#fit01<-stan(model_code =mu_si_reg,data=,dataSet03
#  init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit01, file="./scrT/obje/stan_obje0301")
#load(file="./scrT/obje/stan_obje0301"); #予め stan( ) で作った object をロード
ext01<-extract(fit01, par);                #乱数の取り出し
##### 図 3.6 パス図 (1)
gqcalc3(ext=ext01)
##### 図 3.5 3 次元散布図回帰平面 (1)
plot3d(d01 生物 [,1],d01 生物 [,2],d01 生物 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=-0.410,b= 0.749,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.2 (2) x1 から y への直接効果が正で, 相関・間接効果が負の場合
d02 偏見 <- read.csv("./scrT/dat/ch02/d02 偏見_raw.csv",header=T)
##### 図 3.7 多変量散布図 (2)
pairs.panels(d02 偏見,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet02<-list(n=nrow(d02 偏見),y=d02 偏見 [,3], X=d02 偏見 [,c(1,2)] )#入力
ini<-function(){init(y=d02 偏見 [,3],x1=d02 偏見 [,1],x2=d02 偏見 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit02 <- mod03$sample(data = dataSet02,chains = 5,iter_sampling = 20000,
  init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC

```

```

fit02 <- rstan::read_stan_csv(csrfit02$output_files()) #stan 形式への変換
#####rstan による実行
#fit02<-stan(model_code =mu_si_reg,data=dataSet02,
#  init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit02, file="./scrT/obje/stan_obje0302")
#load(file="./scrT/obje/stan_obje0302"); #予め stan( ) で作った object をロード
ext02<-extract(fit02, par);                #乱数の取り出し
##### 図 3.9 パス図 (2)
gqcalc3(ext=ext02)
##### 図 3.8 3 次元散布図回帰平面 (2)
plot3d(d02 偏見 [,1],d02 偏見 [,2],d02 偏見 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=0.341,b= 0.875,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.3 (4) x1 から y への間接効果が正で, 相関・直接効果が負の場合
d04 印象 <- read.csv("./scrT/dat/ch02/d04 印象_raw.csv",header=T)
##### 図 3.11 3 次元散布図回帰平面 (4)
pairs.panels(d04 印象,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet04<-list(n=nrow(d04 印象),y=d04 印象 [,3], X=d04 印象 [,c(1,2)] )#入力
ini<-function(){init(y=d04 印象 [,3],x1=d04 印象 [,1],x2=d04 印象 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit04 <- mod03$sample(data = dataSet04,chains = 5,iter_sampling = 20000,
  init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit04 <- rstan::read_stan_csv(csrfit04$output_files()) #stan 形式への変換
#####rstan による実行
#fit04<-stan(model_code =mu_si_reg,data=dataSet04,
#  init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit04, file="./scrT/obje/stan_obje0304")
#load(file="./scrT/obje/stan_obje0304"); #予め stan( ) で作った object をロード
ext04<-extract(fit04, par);                #乱数の取り出し
##### 図 3.12 パス図 (4)
gqcalc3(ext=ext04)
##### 図 3.11 3 次元散布図回帰平面 (4)
plot3d(d04 印象 [,1],d04 印象 [,2],d04 印象 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=-0.720,b=0.952,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.4 (5) x1 から y への間接効果が負で, 相関・直接効果が正の場合
d05 歩留 <- read.csv("./scrT/dat/ch02/d05 歩留_raw.csv",header=T)
##### 図 3.13 多変量散布図 (5)
pairs.panels(d05 歩留,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet05<-list(n=nrow(d05 歩留),y=d05 歩留 [,3], X=d05 歩留 [,c(1,2)] )#入力
ini<-function(){init(y=d05 歩留 [,3],x1=d05 歩留 [,1],x2=d05 歩留 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit05 <- mod03$sample(data = dataSet05,chains = 5,iter_sampling = 20000,
  init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC

```

```

fit05 <- rstan::read_stan_csv(csrfit05$output_files()) #stan 形式への変換
#####rstan による実行
#fit05<-stan(model_code =mu_si_reg,data=dataSet05,
# init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit05, file="./scrT/obje/stan_obje0305")
#load(file="./scrT/obje/stan_obje0305"); #予め stan( ) で作った object をロード
ext05<-extract(fit05, par); #乱数の取り出し
##### 図 3.15 パス図 (5)
gqcalc3(ext=ext05)
##### 図 3.14 3 次元散布図回帰平面 (5)
plot3d(d05 歩留 [,1],d05 歩留 [,2],d05 歩留 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=0.460,b= 0.540,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.5 (6) 正の相加効果がある場合
d06 中 3 <- read.csv("./scrT/dat/ch02/d06 中 3_raw.csv", header=T)
##### 図 3.16 多変量散布図 (6)
pairs.panels(d06 中 3, smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet06<-list(n=nrow(d06 中 3),y=d06 中 3[,3], X=d06 中 3[,c(1,2)] )#入力
ini<-function(){init(y=d06 中 3[,3],x1=d06 中 3[,1],x2=d06 中 3[,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit06 <- mod03$sample(data = dataSet06,chains = 5,iter_sampling = 20000,
init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit06 <- rstan::read_stan_csv(csrfit06$output_files()) #stan 形式への変換
#####rstan による実行
#fit06<-stan(model_code =mu_si_reg,data=dataSet06,
# init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit06, file="./scrT/obje/stan_obje0306")
#load(file="./scrT/obje/stan_obje0306"); #予め stan( ) で作った object をロード
ext06<-extract(fit06, par); #乱数の取り出し
##### 図 3.18 パス図 (6)
gqcalc3(ext=ext06)
##### 図 3.17 3 次元散布図回帰平面 (6)
plot3d(d06 中 3[,1],d06 中 3[,2],d06 中 3[,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=0.468,b=0.579 ,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.6 (7) 負の相加効果がある場合
d07 孤独 <- read.csv("./scrT/dat/ch02/d07 孤独_raw.csv",header=T)
##### 図 3.19 多変量散布図 (7)
pairs.panels(d07 孤独,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet07<-list(n=nrow(d07 孤独),y=d07 孤独 [,3], X=d07 孤独 [,c(1,2)] )#入力
ini<-function(){init(y=d07 孤独 [,3],x1=d07 孤独 [,1],x2=d07 孤独 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit07 <- mod03$sample(data = dataSet07,chains = 5,iter_sampling = 20000,
init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC

```

```

fit07 <- rstan::read_stan_csv(csrfit07$output_files()) #stan 形式への変換
#####rstan による実行
#fit07<-stan(model_code =mu_si_reg,data=dataSet07,
# init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit07, file="./scrT/obje/stan_obje0307")
#load(file="./scrT/obje/stan_obje0307"); #予め stan( ) で作った object をロード
ext07<-extract(fit07, par); #乱数の取り出し
##### 図 3.21 パス図 (7)
gqcalc3(ext=ext07)
##### 図 3.20 3 次元散布図回帰平面 (7)
plot3d(d07 孤独 [,1],d07 孤独 [,2],d07 孤独 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a=-0.657,b=0.433 ,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.7 (8) 多重共線性が生じている場合
d08 身長 <- read.csv("./scrT/dat/ch02/d08 身長_raw.csv",header=T)
##### 図 3.22 多変量散布図 (8)
pairs.panels(d08 身長,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet08<-list(n=nrow(d08 身長),y=d08 身長 [,3], X=d08 身長 [,c(1,2)] )#入力
ini<-function(){init(y=d08 身長 [,3],x1=d08 身長 [,1],x2=d08 身長 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit08 <- mod03$sample(data = dataSet08,chains = 5,iter_sampling = 20000,
init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit08 <- rstan::read_stan_csv(csrfit08$output_files()) #stan 形式への変換
#####rstan による実行
#fit08<-stan(model_code =mu_si_reg,data=dataSet08,
# init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit08, file="./scrT/obje/stan_obje0308")
#load(file="./scrT/obje/stan_obje0308"); #予め stan( ) で作った object をロード
ext08<-extract(fit08, par); #乱数の取り出し
##### 図 3.24 パス図 (8)
gqcalc3(ext=ext08)
##### 図 3.23 3 次元散布図回帰平面 (8)
plot3d(d08 身長 [,1],d08 身長 [,2],d08 身長 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a= 4.859,b= -4.641 ,c=-1,d=0, alpha = 0.1)

```

```

##### 3.3.8 (9) 予測変数間の相関が 0 に近い場合
d09 筆実 <- read.csv("./scrT/dat/ch02/d09 筆実_raw.csv", header=T)
##### 図 3.25 多変量散布図 (9)
pairs.panels(d09 筆実,smooth=F,density=F,ellipses=F,pch =16,rug=F,cex.axis=1.0)
dataSet09<-list(n=nrow(d09 筆実),y=d09 筆実 [,3], X=d09 筆実 [,c(1,2)] )#入力
ini<-function(){init(y=d09 筆実 [,3],x1=d09 筆実 [,1],x2=d09 筆実 [,2])}#初期値
#####cmdstanr による実行
#### コンパイルは 'mod03' を使用する
csrfit09 <- mod03$sample(data = dataSet09,chains = 5,iter_sampling = 20000,
init =ini,iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC

```

```
fit09 <- rstan::read_stan_csv(csrfit09$output_files()) #stan 形式への変換
#####rstan による実行
#fit09<-stan(model_code =mu_si_reg,data=DataSet09,
# init =ini,pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit09, file="./scrT/obje/stan_obje0309")
#load(file="./scrT/obje/stan_obje0309"); #予め stan( ) で作った object をロード
ext09<-extract(fit09, par); #乱数の取り出し
##### 図 3.27 パス図 (9)
gqcalc3(ext=ext09)
##### 図 3.26 3 次元散布図回帰平面 (9)
plot3d(d09 筆実 [,1],d09 筆実 [,2],d09 筆実 [,3],size=3,type="p"
,xlab="z1", ylab="z2", zlab="z3")
planes3d(a= 0.434,b= 0.626 ,c=-1,d=0, alpha = 0.1)
```

## 3.2 initi : 重回帰と単回帰の初期値を計算する関数

```
initi<-function(y,x1,x2)
```

### 引数

y 重回帰の基準変数  
x1 重回帰と単回帰の予測変数  
x2 重回帰の予測変数で、かつ単回帰の基準変数

### 戻り値

ay 重回帰切片  
a2 単回帰切片  
b 重回帰係数ベクトル  
b12 単回帰係数  
sigma\_y 誤差標準偏差 (重回帰)  
sigma\_2 誤差標準偏差 (単回帰)

## 3.3 この章で初めて登場した stan のブロック

### 3.3.1 transformed data{ } ブロック

//01, //02, //03 で、後に使用する標本標準偏差を計算している。サンプリング実行中に値が変化しない変数 (データ) を扱い、当該データを変形する。標本標準偏差は MCMC によって値が変化しないのでここに置いている。もし一部でも値が変化するならば、transformed parameters{ } ブロックに記述する。

### 3.3.2 generated quantities{ } ブロック

//08 から //15 で生成量の事後分布を求めている。本ブロックで記述されたステートメントは MCMC が終わった後に計算される。transformed parameters ブロックに置く必然性のないステートメントは、ここに置いたほうが計算効率が高い。

//04 と //05 の予測値は、//06 と //07 で使用され、MCMC の計算の最中に使用するから、このスクリプトでは generated quantities ブロックには置けない。

## 3.4 自習問題

stan コード mu\_si\_reg では、generated quantities ブロックで生成量を計算している。生成量は、このように stan の中で計算できるが、R に戻ってきてからも計算できる。R に戻ってから生成量を計算することは、分析の着想を広げたときに、しばしば行われる。

1. 生成量である間接効果と総合効果を R で計算し、gqcal によって事後分布の数値要約を下さい。ただし間接効果と総合効果以外の生成量は、mu\_si\_reg が作った生成量を利用してよい。
2. 間接効果と総合効果の phc 曲線を描きなさい。区間は任せる。
3. 間接効果と総合効果の phc テーブルを描きなさい。計算する点は任せる。

## 3.5 第 3 章 宿題

P.41, 3.6 節 実習課題 を下さい。論文中に  $y, x_1, x_2$  間の 3 つの相関  $r_{y1}, r_{y2}, r_{21}$  が示された論文を選ぶこと。

- $b_1^* = (r_{y1} - r_{y2} \times r_{21}) / (1 - r_{21}^2)$
- $b_2^* = (r_{y2} - r_{y1} \times r_{21}) / (1 - r_{21}^2)$
- $b_{21}^* = r_{21}$
- 総合効果 =  $r_{y1}$

- 決定係数  $= (r_{y1}^2 + r_{y2}^2 - 2 \times r_{y1} \times r_{y2} \times r_{21}) / (1 - r_{21}^2)$
- 間接効果  $= b_2^* \times b_{12}^*$

という最小 2 乗法による点推定値を得て、問 3) を解く。この宿題には事後分布は必要ないから、stan を使用する必要はない。



## 4 ロジスティック回帰/メタ分析

### 4.1 スクリプト

#####第4章 ロジスティック回帰/メタ分析

# Working directory が 'toyoda' であることの確認

```
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義
```

#####4.1 ロジスティック回帰 (ベルヌイ分布)

### 表 4.1 管理職か否かと年齢

(昇進<-read.csv("./scrT/dat/課長昇進.csv", header = TRUE))

### 表 4.2 年齢別の管理職者の人数

table(昇進); #表 4.2

###図 4.1 散点図と回帰直線

```
temp<-昇進[,2]+runif(200,-0.5,0.5); #図を見やすくする乱数付加
plot(temp, 昇進[,1],ylim=c(-0.2,1.2),xlim=c(30,46),
      xlab="年齢",ylab="管理職");
abline(a=-3.06335, b=0.09377, lwd=1.8);
text(40,0.6,"回帰直線",cex=1.5); #図 4.1 ここまで
```

### 4.1.2 ロジスティック変換

#オッズ変換 実数全体から確率への変換 (4.4) 式

```
odds<-function(prob){prob/(1-prob)}
plot(odds,0,1,xlab="確率",main="オッズ変換")
```

#ロジット変換 (対数オッズ変換) (4.5) 式

```
log_odds<-function(prob){log(prob/(1-prob))}
plot(log_odds,0,1,xlab="確率",main="ロジット変換")
```

#ロジスティック変換 (ロジット変換の逆変換) (4.7) 式

```
logistic<-function(yhat){1/(1+exp((-1)*yhat))}
```

###図 4.2 逆ロジット関数 logit-1( )

```
plot(logistic,-3,3,xlab="yhat",main="ロジスティック変換")
```

#####ロジスティック回帰 (ベルヌイ分布) の stan コード

```
logistic_reg_01<-`
data {
  int<lower=0> n;          //データ数
  int<lower=0,upper=1> y[n]; //基準変数 (0-1 データ)
  vector[n] X;            //予測変数ベクトル
```

```
}
parameters {
  real a;                //切片
  real b;                //偏回帰係数
}
transformed parameters {
  vector<lower=0,upper=1>[n] prob;
  for(i in 1:n)
    prob[i] = inv_logit(a + X[i]*b); //反応確率 (4.10) 式
}
model {
  for(i in 1:n)
    y[i] ~ bernoulli(prob[i]); //ベルヌイ分布モデル (4.8) 式
}
`;
```

```
par<-c("a","b") #母数
dataSet01 <-list(n=nrow(昇進),y=昇進[,1], X=昇進[,2]) #入力
```

#####cmdstanr による実行

```
modfile01 <- write_stan_file(logistic_reg_01) #書き出す一時ファイル
mod01 <- cmdstan_model(modfile01) #コンパイル
csrfit01 <- mod01$sample(data = dataSet01,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit <- rstan::read_stan_csv(csrfit01$output_files()) #stan 形式への変換
```

#####rstan による実行

```
#fit<-stan(model_code =logistic_reg_01,data=dataSet01,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit, file="./scrT/obje/stan_obje0401")
#load(file="./scrT/obje/stan_obje0401"); #予め stan( ) で作った object をロード
```

```
ext<-extract(fit, par); #乱数の取り出し
```

#####事後分布のヒストグラム (教科書にはない)

```
hist(ext$a,breaks=100)
hist(ext$b,breaks=100)
```

#####生成量

```
p0_5<- (-1)*ext$a/ext$b; #50%の確率で課長になれる年齢 p.46,1.3
oddshi<-exp(ext$b); #年齢が1歳変わった時のオッズ比 (4.13) 式
p35<- 1/(1+exp((-1)*(ext$a+ext$b*35))); #35歳で昇進できる確率 (4.14) 式
```

###表 4.3 母数と生成量の事後分布の要約

```
gqcal(ext$a);
gqcal(ext$b);
gqcal(p0_5);
gqcal(oddshi);
gqcal(p35)
```

###図 4.3 年齢の上昇に伴う昇進確率の変化

```
plot(temp, 昇進[,1],ylim=c(-0.2,1.2),xlim=c(30,46),
```

```

xlab="年齢",ylab="管理職",main="ロジスティック回帰")
par(new=T)
curve(1/(1+exp((-1)*((-18.679)+0.492*x))),30,46,lwd=1.8,
ylim=c(-0.2,1.2),xlim=c(30,46),xlab="",ylab=""); #図 4.3 ここまで

```

#### ##### 4.2 ロジスティック回帰 (2 項分布)

##### ###表 4.4 失語症の会話促進データ

```
(失語症訓練<-read.csv("./scrT/dat/失語症訓練.csv", header = TRUE));#表 4.4
```

##### ###図 4.4 訓練回数と正答率の折れ線グラフ

```

(ab<-coef(glm(失語症訓練[,2]/失語症訓練[,1]~失語症訓練[,3])))
plot(失語症訓練[,3], 失語症訓練[,2]/失語症訓練[,1],
ylim=c(0,1),type="o",xlab="訓練回数",ylab="正解率")
abline(a=ab[1],b=ab[2]); #図 4.4 ここまで

```

##### #####図 4.5 2 項分布の確率関数を 1 つずつ見る

```

size<-10;
for (prob in (1:9)/10 ){
x<-c(0:size)
plot(x, dbinom(x, size, prob), type="o",
main=paste("2 項分布 size:",size,"", prob:" ",prob) )
locator(); #右クリックして"停止"
}

```

##### #####図 4.5 2 項分布の確率関数

```

size<-10;x<-c(0:size)
plot(x, dbinom(x, size, 0.1), type="o",ylim=c(0,0.4),xlab="",ylab="")
text(1,0.3,paste("p=",prob,sep=""))
for (prob in c(0.3,0.5,0.7,0.9)){
par(new=T)
plot(x, dbinom(x, size, prob), type="o",ylim=c(0,0.4),xlab="",ylab="")
text(prob*10,0.3,paste("p=",prob,sep=""))
}

```

##### #####試行数が多い場合 (正規分布と変わらなくなる) 教科書にはない

```

size<-100;
for (prob in (1:9)/10 ){
x<-c(0:size)
plot(x, dbinom(x, size, prob), type="o",
main=paste("2 項分布 size:",size,"", prob:" ",prob) )
locator(); #右クリックして"停止"
}

```

##### #####ロジスティック回帰 (2 項分布) の stan コード

```

binom_reg<-'
data {
int<lower=0> n; //データ数
int<lower=0> t[n]; //試行数
int<lower=0> y[n]; //成功数

```

```

vector[n] X; //予測変数ベクトル
}
parameters {
real a; //切片
real b; //偏回帰係数
}
transformed parameters {
vector<lower=0,upper=1>[n] prob;
for(i in 1:n)
prob[i] = inv_logit(a + X[i]*b); //反応確率 (4.17) 式
}
model {
for(i in 1:n)
y[i] ~ binomial(t[i], prob[i]); //2 項分布モデル (4.16) 式,(4.17) 式
}
';

```

```

par<-c("a","b") #母数
dataSet02 <-list(n=nrow(失語症訓練),t=失語症訓練[,1],y=失語症訓練[,2],
X=失語症訓練[,3]) #入力

```

##### #####cmdstanr による実行

```

modfile02 <- write_stan_file(binom_reg) #書き出す一時ファイル
mod02 <- cmdstan_model(modfile02) #コンパイル
csrfit02 <- mod02$sample(data = dataSet02,chains = 5,iter_sampling = 20000,
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit02 <- rstan::read_stan_csv(csrfit02$output_files()) #stan 形式への変換

```

##### #####rstan による実行

```

#fit02<-stan(model_code =binom_reg,data=dataSet02,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit02, file="./scrT/obje/stan_obje0402")
#load(file="./scrT/obje/stan_obje0402"); #予め stan( ) で作った object をロード

```

```
ext<-extract(fit02, par); #乱数の取り出し
```

##### #####事後分布と生成量

```

p0_5<- (-1)*ext$a/ext$b; #半分表現できるようになる訓練回数 p,50,1.1
oddshi<-exp(ext$b); #1 回訓練をした場合のオッズ比 p,50,1.4
p35<- 1/(1+exp((-1)*(ext$a+ext$b*35))); # (4.14) 式に 35 を代入 p,50,1.6
p35aste<-rbinom(length(p35),15,p35); # (4.19) 式

```

##### #####表 4.5 母数と生成量の事後分布の要約

```

gqcal(ext$a ,3); #切片
gqcal(ext$b ,3); #傾き
gqcal(p0_5 ,1); #半分表現できるようになる訓練回数
gqcal(oddshi ,2); #1 回訓練をした場合のオッズ比
gqcal(p35 ,3); #あと 5 回訓練した場合の正答率
gqcal(p35aste,1); #35 回目の基本語数が 15 の場合の表現数の事後分布

```

##### #####図 4.6 ロジスティック回帰分析 (2 項分布)

```
plot(失語症訓練[,3], 失語症訓練[,2]/失語症訓練[,1],
```

```

ylim=c(0,1),xlim=c(0,36),type="o",xlab="訓練回数",ylab="正解率")
par(new=T)
curve(1/(1+exp((-1)*((-1.803)+0.099*x))),0,36,lwd=1.8,
ylim=c(0,1),xlim=c(0,36),xlab="",ylab="");
lines(c(35,35),c(0.747,0.905),lwd=1.8)
text(33,0.95,"95 %確信区間");
#図 4.6 ここまで

```

```

#####図 4.7 x = 35, n = 15 のときの「表現数」の事後予測分布
hist(p35aste)

```

#### ##### 4.3 メタ分析

##### #####4.3.1 前向き研究

##### #####表 4.10 BCG の有効性に関する前向き研究のレビュー

```

(前向<-read.csv("./scrT/dat/結核前 BCG.csv", header = TRUE))

```

```

round(risk 実前<-前向 [1,3]/(前向 [1,3]+前向 [1,4]),4); #リスク (4.20) 式
round(risk 対前<-前向 [1,5]/(前向 [1,5]+前向 [1,6]),4); # (4.21) 式
round(risk 実前/risk 対前,3); #リスク比
round((前向 [1,3]*前向 [1,6])/(前向 [1,4]*前向 [1,5]),3); #オッズ比 (4.22) 式

```

##### #####4.3.2 後ろ向き研究

##### #####表 4.12 BCG の有効性に関する後ろ向き研究のレビュー

```

(後向<-read.csv("./scrT/dat/結核後 BCG.csv", header = TRUE)) #表 4.12
round(risk 実後<-後向 [1,3]/(後向 [1,3]+後向 [1,4]),4); #リスク (4.23) 式
round(risk 対後<-後向 [1,5]/(後向 [1,5]+後向 [1,6]),4); # (4.24) 式
round((後向 [1,3]*後向 [1,6])/(後向 [1,4]*後向 [1,5]),3); #オッズ比 (4.25) 式

```

#### #####メタ分析 (オッズ比) の stan コード

```

meta_ana<- '
data {
  int<lower=0> m; //研究数
  int<lower=0> u1[m]; //オッズ比の分子成功数
  int<lower=0> n1[m]; //オッズ比の分子試行数
  int<lower=0> u0[m]; //オッズ比の分母成功数
  int<lower=0> n0[m]; //オッズ比の分母試行数
}
parameters {
  real a[m]; //切片
  real b[m]; //係数
  real mu_a; //切片平均
  real mu_b; //係数平均
  real sigma_a; //切片 SD
  real sigma_b; //係数 SD
}
model {
  for(i in 1:m){
    a[i] ~ normal(mu_a, sigma_a); //切片の事前分布 (4.30) 式
    b[i] ~ normal(mu_b, sigma_b); //係数の事前分布 (4.31) 式
    u1[i] ~ binomial(n1[i], inv_logit(a[i] + b[i])); //(4.26),(4.28),(4.34) 式
    u0[i] ~ binomial(n0[i], inv_logit(a[i] )); //(4.26),(4.29),(4.35) 式
  }
}

```

```

}
}
generated quantities{
  real Odds_r[m];
  real Odds_r_mu;
  Odds_r =exp(b); //研究ごとのオッズ比 (4.13) 式,p.55,1.4
  Odds_r_mu=exp(mu_b); //オッズ比の平均 p.55,1.7
}
';

```

#### #####4.3.4 前向き研究のメタ分析

```

par<-c("Odds_r","Odds_r_mu","mu_b","sigma_b") #母数
dataSet 前 <-list(m=nrow(前向),
u1=前向$u1,n1=前向$u1+前向$u12,
u0=前向$u21,n0=前向$u21+前向$u22) #入力

#####cmdstanr による実行
modfile 前 <- write_stan_file(meta_ana) #書き出す一時ファイル
mod 前 <- cmdstan_model(modfile 前) #コンパイル
csrfit 前 <- mod 前$sample(data = dataSet 前,chains = 5,iter_sampling = 20000,
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 前 <- rstan::read_stan_csv(csrfit 前$output_files()) #stan 形式への変換

```

```

#####rstan による実行
fit 前<-stan(model_code =meta_ana,data=dataSet 前,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 前, file="./scrT/obje/stan_obje0403")
#load(file="./scrT/obje/stan_obje0403"); #予め stan( ) で作った object をロード

```

```

ext 前<-extract(fit 前, par); #乱数の取り出し

```

```

#####表 4.10 の Odds/EAP
print(fit 前,digits_summary=3,prob=prob)

```

```

#####オッズ比の平均の事後分布のヒストグラム (教科書にはない)
hist(ext 前$Odds_r_mu,breaks=200,xlim=c(0,1.0),main="")

```

```

#####オッズ比の事後予測分布 (4.33) 式のヒストグラム (教科書にはない)
Odds_r_aste<-exp(rnorm(length(ext 前$mu_b),ext 前$mu_b,ext 前$sigma_b))
hist(Odds_r_aste,breaks=1000,xlim=c(0,4.0),main="")

```

#### #####表 4.11 前向き研究のオッズの平均の事後分布と事後予測分布の要約

```

gqcal(exp(ext 前$mu_b),3); #表 4.11
gqcal(Odds_r_aste,3)

```

```

#####phc p,55,b1.3
mean(1.0<Odds_r_aste)
mean(Odds_r_aste<0.8)

```

```

#「各前向き研究のオッズ比の事後分布の箱ひげ図」(教科書にはない)
aa<-ext 前$Odds_r[,1]; for (i in 2:13){aa<-c(aa,ext 前$Odds_r[,i])};

```

```
boxplot(aa~rep(1:13,each=100000),outline=F,horizontal=T,cex.axis=1.2);
abline(v=1.0,lwd=2.0)
```

#### #####4.3.5 後ろ向き研究のメタ分析

```
par<-c("Odds_r","Odds_r_mu","mu_b","sigma_b") #母数
dataSet 後 <-list(m=nrow(後向),
  u1=後向$u11,n1=後向$u11+後向$u21,
  u0=後向$u12,n0=後向$u12+後向$u22) #入力
```

#### #####cmdstanr による実行

```
#コンパイルは"mod 前"
csrfit 後 <- mod 前$sample(data = dataSet 後,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 後 <- rstan::read_stan_csv(csrfit 後$output_files()) #stan 形式への変換
```

#### #####rstan による実行

```
#fit 後<-stan(model_code =meta_ana,data=dataSet 後,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 後, file="./scrT/obje/stan_obje0404")
#load(file="./scrT/obje/stan_obje0404"); #予め stan( ) で作った object をロード
```

```
ext 後<-extract(fit 後, par); #乱数の取り出し
```

#### ####表 4.12 の Odds/EAP

```
print(fit 後,digits_summary=3,prob=prob)
Odds_r_aste<-exp(rnorm(length(ext 後$mu_b),ext 後$mu_b,ext 後$sigma_b))
```

#### ####表 4.13 後ろ向き研究のオッズの平均の事後分布と事後予測分布の要約

```
gqcal(exp(ext 後$mu_b),3)
gqcal(Odds_r_aste,3)
```

```
#####phc p.57.b1.1
mean(1.0<Odds_r_aste)
mean(Odds_r_aste<0.8)
```

#### ####図 4.8 各後ろ向き研究のオッズ比の事後分布

```
aa<-ext 後$Odds_r[,1]; for (i in 2:10){aa<-c(aa,ext 後$Odds_r[,i])}
boxplot(aa~rep(1:10,each=100000),outline=F,horizontal=T,cex.axis=1.5);
abline(v=1.0,lwd=2.0)
```

#### #####実習課題

```
### 1.
#load(file="./scrT/obje/stan_obje0401"); #予め stan( ) で作った object をロード
par<-c("a","b")
ext<-extract(fit, par); #乱数の取り出し
p0_5<- (-1)*ext$a/ext$b; #50%の確率で課長になれる年齢
oddshi<-exp(ext$b); #年齢が 1 歳変わった時のオッズ比
p35<- 1/(1+exp((-1)*(ext$a+ext$b*35))); #35 歳で昇進できる確率
gqcal(p0_5)
```

```
gqcal(oddshi)
```

```
#
# 関数 phc01 使った部分は割愛
#
```

```
### 2.
#load(file="./scrT/obje/stan_obje0402"); #予め stan( ) で作った object をロード
par<-c("a","b")
ext<-extract(fit02, par); #乱数の取り出し
p35<- 1/(1+exp((-1)*(ext$a+ext$b*35))); #あと 5 回訓練した場合の正答率
gqcal(p35)
```

```
#
# 関数 phc01 使った部分は割愛
#
```

```
### 3.
#load(file="./scrT/obje/stan_obje0404"); #予め stan( ) で作った object をロード
par<-c("Odds_r","Odds_r_mu","mu_b","sigma_b")
ext 後<-extract(fit 後, par); #乱数の取り出し
gqcal(ext 後$Odds_r_mu)
phc01(seq(0.25,0.70,0.01),ext 後$Odds_r_mu,cc="gtc",byoga="yes",
  xlab="オッズの平均")
phc01(seq(0.25,0.70,0.05),ext 後$Odds_r_mu,cc="gtc",byoga="no")
Odds_r_aste<-exp(rnorm(length(ext 後$mu_b),ext 後$mu_b,ext 後$sigma_b))
gqcal(Odds_r_aste,3)
phc01(seq(0.05,2.1,0.01),ext 後$Odds_r_mu,cc="gtc",byoga="yes",
  xlab="オッズの事後予測分布")
phc01(seq(0.05,0.3,0.05),ext 後$Odds_r_mu,cc="gtc",byoga="no")
```

## 4.2 以下の関数を自分で復習しよう

```
table, runif, exp, par(new=T), curve, boxplot,
dbinom(x, size, prob), locator,
```

## 4.3 ロジスティック回帰（ベルヌイ分布）の stan コード

```
prob[i] = inv_logit(a + X[i]*b);
```

$$p = \frac{1}{1 + \exp(-(a + bx))}$$

```
y[i] ~ bernoulli(prob[i]); y_i ~ Bernoulli(p_i)
```

## 4.4 ロジスティック回帰（2項分布）

### 4.4.1 glm: 一般線形モデル

```
glm(y~x)
```

#### 引数

**y** 基準変数

**x** 予測変数

### 4.4.2 coef: 係数を取り出す

```
coef(x)
```

#### 引数

**x** クラス gml のオブジェクト

#### 戻り値

一般線形モデルの母数

## 4.5 ロジスティック回帰（2項分布）の stan コード

```
y[i] ~ binomial(t[i], prob[i]);      y_i ~ binomial(n_i, p_i)
```

## 4.6 メタ分析（オッズ比）の stan コード

```
u1[i] ~ binomial(n1[i], inv_logit(a[i] + b[i]));
```

binomial( ) と inv\_logit( ) の組み合わせ

## 4.7 自習問題

以下の生成量の phc 曲線と phc テーブルを作成しなさい。

1. 「昇進データ」から計算した「昇進する確率が五分五分の年齢」「1 年後に昇進しているオッズ比」
2. 「失語症の会話促進データ」から計算した「あと 5 回訓練した場合の正答率」

## 4.8 第 4 章 宿題

RPG ゲーム X では、登場する 10 匹のモンスターをできるだけ多くやっつけることが求められている。100 人のプレイヤーが、このステージに 7 回挑んだときに「何匹のモンスターをやっつけたか」が "RPGX.csv" に記録されている。データは次のように読み込む。

```
read.csv("scrT/dat/RPGX.csv", header = T, row.names=1)
```

行名はプレイヤー番号、変数  $v_i$  ( $i=1\cdots 7$ ) は  $i$  回目に倒したモンスターの匹数である。あなたの学籍番号の下 2 桁 \*\* に相当するプレイヤー  $s^{**}$  のデータを以下のように分析せよ。

1. 分析するプレイヤーの番号、生データを示せ。
2. 何回目の挑戦かを予測変数、10 匹中倒したモンスターを基準変数としてロジスティック回帰分析（2 項分布）をして、母数  $a, b$  の事後分布を求めよ。
3. 半分（5 匹）はやっつけられる回数の事後分布を求めよ。
4. 1 回ステージにトライすると、どれくらい腕前は向上するだろうか。オッズ比の事後分布を求めよ。
5. 8 回目にトライしたら何割倒せるだろう。その確率の事後分布を求めよ。
6. 図 4.6 に相当する、「トライ数×やっつけ率」の折れ線グラフと、ロジスティック回帰曲線を描きなさい。ただし確信区間は描かなくてよい。

## 5 ポアソンモデル

### 5.1 スクリプト

```
#####第5章 ポアソンモデル／対数線形モデル
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))        #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))            #wd 戻す
library(rstan)           #パッケージ rstan の呼び出し
library(cmdstanr)        #パッケージ cmdstanr の呼び出し
library(posterior)       #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

#####5.1 ポアソン分布

###図 5.1 ポアソン分布の確率関数
par(mfrow=c(1,2))
u<-c(0:5);p061<-dpois(u, 0.61);names(p061)<-u
barplot(p061); text(4,0.3,"λ=0.61",cex=2.0)
x<-c(0:10)
plot(x, dpois(x, 2.0), type="o",xlab="",ylab="",xlim=c(0,10),ylim=c(0,0.3))
for (lambda in seq(3.0,6.0,1.0)){
  par(new=T)
  plot(x,dpois(x,lambda),type="o",xlab="",ylab="",xlim=c(0,10),ylim=c(0,0.3))
}
text(c(2,3,4,5,6),c(0.29,0.25,0.22,0.19,0.18),
     paste("λ=",2:6,sep=""),cex=1.5)
par(mfrow=c(1,1));          #図 5.1 ここまで
#dev.copy2eps(file="z0401.eps",family="Japan1")

####λ が大きくなると正規分布に近似 (教科書にはない)
size<-100; x<-c(0:size);
plot(x, dpois(x, 50), type="o",main=paste("ポアソン分布 λ:",50))

###5.1.1 数値例 (馬に蹴られて死亡した兵士数)
###表 5.1 プロシア陸軍で馬に蹴られて死亡した兵士数
Horse<-c(dpois(0, 0.61),dpois(1, 0.61),dpois(2, 0.61),
         dpois(3, 0.61),dpois(4, 0.61),dpois(5, 0.61));
print(0:5);print(c(109, 65, 22, 3, 1, 0));round(Horse*200,1)

###5.1.2 数値例 (当たりの本数の確率)
ice<-c(dpois(0, 0.25),dpois(1, 0.25),dpois(2, 0.25),          #(5.7) 式
       dpois(3, 0.25),dpois(4, 0.25));round(ice,3)

#####5.2 ポアソン分布の推測
####5.2.1 爆弾命中数と区画数のデータ
爆弾<-c(rep(0,229),rep(1,211),rep(2,93),rep(3,35),rep(4,7),rep(5,1))
```

###表 5.2 ロンドンにおける爆弾命中数と事後予測チェック の前半  
table(爆弾)

```
barplot(table(爆弾),xlab="命中数",ylab="区画数");#ヒストグラム (教科書にはない)
round(mean(爆弾),3);          #平均 p.62,1.2
round(mean((爆弾-mean(爆弾))^2),3); #分散 p.62,1.3
```

```
#####ポアソン分布の推定の stan コード
Poisson<-'
data {
  int<lower=0> n;          //データ数
  int<lower=0> u[n];       //計数データ
}
parameters {
  real lambda;           //母数
}
model {
  for(i in 1:n)
    u[i] ~ poisson(lambda); //ポアソン分布 (5.5) 式
}
generated quantities{
  int<lower=0> uaste;      //予測分布
  uaste = poisson_rng(lambda); //ポアソン乱数 (5.8) 式
}
';

par<-c("lambda","uaste") #母数
dataSet 爆 <-list(n=length(爆弾),u=爆弾) #入力

#####cmdstanr による実行
modfile 爆 <- write_stan_file(Poisson)          #書き出す一時ファイル
mod 爆 <- cmdstan_model(modfile 爆)             #コンパイル
csrfit 爆 <- mod 爆$sample(data = dataSet 爆,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 爆 <- rstan::read_stan_csv(csrfit 爆$output_files()) #stan 形式への変換

#####rstan による実行
#fit 爆<-stan(model_code =Poisson, data=dataSet 爆,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 爆, file="./scrT/obje/stan_obje0501")
#load(file="./scrT/obje/stan_obje0501"); #予め stan( ) で作った object をロード

ext 爆<-extract(fit 爆, par);          #乱数の取り出し

#####表 5.3 母数と生成量の事後分布の要約
sd 爆<-sqrt(ext 爆$lambda);          #標準偏差
gqcal(ext 爆$lambda,3);              #平均事後分布
gqcal(sd 爆 ,3);                    #sd 事後分布

###表 5.2 ロンドンにおける爆弾命中数と事後予測チェック の後半
round(table(ext 爆$uaste)/length(ext 爆$uaste),3); #確率
```

```

round(table(ext 爆$uaste)/length(ext 爆$uaste)*576,1);    #事後予測値

#####5.2.2 授業欠席データ
#####表 5.4 「心理統計学」の授業の欠席者数
(欠席<-c(0,1,2,0,0,1,2,0,2,0,1,0,0,1))

round(mean(欠席),3);          #p.63,1.3
round(mean((欠席-mean(欠席))^2),3); #p.63,1.3

par<-c("lambda","uaste")      #母数
dataSet 欠 <-list(n=length(欠席),u=欠席) #入力

#####cmdstanr による実行
#                               コンパイル済の "mod 爆" を使用する
csrfit 欠 <- mod 爆$sample(data = dataSet 欠,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 欠 <- rstan::read_stan_csv(csrfit 欠$output_files()) #stan 形式への変換

#####rstan による実行
#fit 欠<-stan(model_code =Poisson,data=dataSet 欠,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 欠, file="./scrT/obje/stan_obje0502")
#load(file="./scrT/obje/stan_obje0502"); #予め stan( ) で作った object をロード

ext 欠<-extract(fit 欠, par);          #乱数の取り出し

#####表 5.5 母数と生成量の事後分布の要約
sd 欠<-sqrt(ext 欠$lambda)
gqcal(ext 欠$lambda,3)
gqcal(sd 欠,3)

#####表 5.6 欠席者の事後予測チェック
round(table(ext 欠$uaste)/length(ext 欠$uaste),3);    #表 5.6
round(table(ext 欠$uaste)/length(ext 欠$uaste)*14,1)

#表 5.7 母数と生成量の事後分布の要約 (事後予測値の分布の比較)
予測確率 2 爆<-dpois(2,ext 爆$lambda)
予測確率 2 欠<-dpois(2,ext 欠$lambda)
予測値 2 爆<-予測確率 2 爆*576
予測値 2 欠<-予測確率 2 欠*14
gqcal(予測確率 2 爆,3)
gqcal(予測値 2 爆,1)
gqcal(予測確率 2 欠,3)
gqcal(予測値 2 欠,2)

##### 5.3 2つのポアソン分布の比較

##### 表 5.8 鏡映描写課題における逸脱数の分布
u1<-c(1,0,2,3,2,3,3,1,4,3,1,0,0,1,2,1,3,2,1,4,1,1,1,1,3,2,3,1,1,5)
u2<-c(1,3,2,7,4,4,2,1,0,2,1,2,1,0,1,5,5,3,4,2,1,3,1,1,1,2,3,1,1,3,0)
mean(u1);mean(u2)

```

```

table(u1);table(u2)

##### 2つのポアソン分布の推測の stan コード
Poisson2<-‘
data {
  int<lower=0> n1;          //データ数
  int<lower=0> n2;          //データ数
  int<lower=0> u1[n1];      //計数データ
  int<lower=0> u2[n2];      //計数データ
}
parameters {
  real lambda1;            //母数 1
  real lambda2;            //母数 2
}
model {
  for(i in 1:n1) u1[i] ~ poisson(lambda1); (5.10) 式
  for(i in 1:n2) u2[i] ~ poisson(lambda2); (5.10) 式
}
’;
par<-c("lambda1","lambda2")      #母数
dataSet 鏡 2 <-list(n1=length(u1),n2=length(u2),u1=u1,u2=u2) #入力

#####cmdstanr による実行
modfile 鏡 2 <- write_stan_file(Poisson2)          #書き出す一時ファイル
mod 鏡 2 <- cmdstan_model(modfile 鏡 2)            #コンパイル
csrfit 鏡 2 <- mod 鏡 2$sample(data = dataSet 鏡 2,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 鏡 2 <- rstan::read_stan_csv(csrfit 鏡 2$output_files()) #stan 形式への変換

#####rstan による実行
#fit 鏡 2<-stan(model_code =Poisson2, data=dataSet 鏡 2,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 鏡 2, file="./scrT/obje/stan_obje0503")
#load(file="./scrT/obje/stan_obje0503"); #予め stan( ) で作った object をロード

ext 鏡 2<-extract(fit 鏡 2, par);          #乱数の取り出し

#####表 5.9 2群のポアソンモデルの母数の事後分布の要約
gqcal(ext 鏡 2$lambda1,3)
gqcal(ext 鏡 2$lambda2,3)

##### 表 5.10 ROPE である phc, ROPE でない phc
sequ<-seq(0,1.2,0.2);a<-ext 鏡 2$lambda2;b<-ext 鏡 2$lambda1
phc01(sequ,a,b,cc="rope",byoga="no")
1-phc01(sequ,a,b,cc="rope",byoga="no")

##### 表 5.11 λ 2 のほうが大きいという phc
phc01(sequ,a,b,cc="gtc", byoga="no")

#####5.4 ポアソン回帰

```

##### 表 5.12 鏡映描写課題における逸脱数

```
(X<-data.frame(
  試行数=c(1:15),
  逸脱数=c(6,4,3,2,2,2,1,1,1,0,0,2,1,0,0)))
```

##### 図 5.3 15 試行における逸脱数

```
plot(X[,1],X[,2],type="o",xlab="試行数",ylab="逸脱数")
abline(a=coef(glm(X[,2]~X[,1]))[[1]],coef(glm(X[,2]~X[,1]))[[2]])
```

##### 5. 4. 1 指数変換 実数全体から正の実数への変換

##### 図 5.4 指数変換  $\exp(\cdot)$

```
plot(exp,-2,2,xlab="a+bx",ylab="λ",main="", lwd=2.0,cex.lab=1.5)
```

##### ポアソン回帰の stan コード

```
Poisson_reg<-'
data {
  int<lower=0> n; //データ数
  int<lower=0> y[n]; //基準変数 (計数データ)
  vector[n] X; //予測変数ベクトル
}
parameters {
  real a; //切片
  real b; //偏回帰係数
}
transformed parameters {
  vector<lower=0>[n] lambda; //ラムダ
  for(i in 1:n)
    lambda[i] = exp(a + X[i]*b); // (5.14) 式, (5.19) 式
}
model {
  for(i in 1:n)
    y[i] ~ poisson(lambda[i]); //ポアソン分布モデル (5.13) 式, (5.15) 式
}
generated quantities{
  int<lower=0> yaste[n]; //予測分布
  for(i in 1:n)
    yaste[i] = poisson_rng(lambda[i]); //ポアソン分布モデル (5.20) 式
}
';
```

```
par<-c("a","b","lambda","yaste") #母数
dataSet 回帰 <-list(n=nrow(X),y=X[,2], X=X[,1]) #入力
```

##### cmdstanr による実行

```
modfile 回帰 <- write_stan_file(Poisson_reg) #書き出す一時ファイル
mod 回帰 <- cmdstan_model(modfile 回帰) #コンパイル
csrfit 回帰 <- mod 回帰$sample(data = dataSet 回帰, chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 回帰 <- rstan::read_stan_csv(csrfit 回帰$output_files()) #stan 形式への変換
```

##### rstan による実行

```
#fit 回帰<-stan(model_code =Poisson_reg, data=dataSet 回帰,#データ
```

```
# pars=par, seed=1134, chains=1, warmup=1000, iter=101000)
#save(fit 回帰, file="./scrT/obje/stan_obje0504")
#load(file="./scrT/obje/stan_obje0504"); #予め stan( ) で作った object をロード
```

```
ext 回帰<-extract(fit 回帰, par); #乱数の取り出し
```

#### 生成量：予測変数を一単位変化させたときの母数の変化率

```
expb<-exp(ext 回帰$b); (5.21) 式
```

##### 表 5.13 ポアソン回帰モデルの母数・生成量等の事後分布の要約

```
gqcal(ext 回帰$a,3)
gqcal(ext 回帰$b,3)
gqcal(ext 回帰$lambda[,7],3)
gqcal(ext 回帰$yaste[,7],3)
gqcal(expb,3)
```

##### 図 5.5 逸脱数のポアソン回帰

```
lambda_i<-apply(ext 回帰$lambda,2,mean)
plot(X[,1],X[,2],type="o",xlab="試行数",ylab="逸脱数",xlim=c(1,15),ylim=c(0,6))
par(new=T);
plot(X[,1],lambda_i,type="l",xlab="",ylab="",lwd=2,xlim=c(1,15),ylim=c(0,6))
```

##### 5.5 対数線形モデル

##### 5.5.1 度数を構造化する

##### 表 5.14 広告 B の宣伝効果を調べる調査の結果  
( $u \leftarrow \text{matrix}(c(29,21,50,145),2,2,\text{byrow}=T)$ )

##### 分割表の分析 01・度数の分解 (2\*2) の stan コード

```
Poisson_Cross01<-'
data {
  int<lower=0> u[2,2]; //基準変数 (計数データ)
}
parameters {
  real mu; //全平均
  real A1; //要因 A の主効果
  real B1; //要因 B の主効果
  real AB11; //交互作用
}
transformed parameters {
  matrix <lower=0>[2,2] lambda; //ラムダ
  lambda[1,1] = exp(mu+ A1 + B1 + AB11); // (5.23) 式, (5.24) 式
  lambda[2,1] = exp(mu- A1 + B1 - AB11);
  lambda[1,2] = exp(mu+ A1 - B1 - AB11);
  lambda[2,2] = exp(mu- A1 - B1 + AB11);
}
model {
  u[1,1] ~ poisson(lambda[1,1]); // (5.25) 式
  u[2,1] ~ poisson(lambda[2,1]);
  u[1,2] ~ poisson(lambda[1,2]);
  u[2,2] ~ poisson(lambda[2,2]);
}
```



```

';

par<-c("mu","A1","B1","AB11","lambda") #母数
dataSet 分割表 01<-list(u=u ) #入力

#####cmdstanr による実行
modfile 分割表 01 <- write_stan_file(Poisson_Cross01) #書き出す一時ファイル
mod 分割表 01 <- cmdstan_model(modfile 分割表 01) #コンパイル
csrfit 分割表 01 <- mod 分割表 01$sample(data = dataSet 分割表 01,chains = 5,
  iter_sampling=20000, iter_warmup=1000, parallel_chains=5, seed=1234)
fit 分割表 01 <- rstan::read_stan_csv(csrfit 分割表 01$output_files()) #n 形式の変換

#####rstan による実行
#fit 分割表 01<-stan(model_code =Poisson_Cross01,data=dataSet 分割表 01,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 分割表 01, file="./scrT/obje/stan_obje0505")
#load(file="./scrT/obje/stan_obje0505"); #予め stan( ) で作った object をロード

ext 分割表 01<-extract(fit 分割表 01, par); #乱数の取り出し

##### 表 5.15 広告効果の分析の母数・生成量の事後分布の要約
gqcal(ext 分割表 01$mu)
gqcal(ext 分割表 01$A1)
gqcal(ext 分割表 01$B1)
gqcal(ext 分割表 01$AB11)

##### 5.5.2 比率を構造化する

##### 表 5.16 タイタニック号の乗員・乗客の犠牲者数
(u<-matrix(c(4,118, 13,154, 106,422, 3,670),4,2,byrow=T)); #犠牲者
(n<-matrix(c(145,180, 106,179, 196,510, 23,862),4,2,byrow=T));#人数

#####分割表の分析 02・比率の分解 (a*b) の stan コード
Poisson_Cross02<-'
functions{
  vector zero_sum_vector(int a, vector m1A){//主効果ゼロ和
    vector[a] muA;
    for(i in 1:(a-1)){ muA[i] = m1A[i];} //a-1 まではソックリ代入
    muA[a] = -sum(m1A); //a 番目に、和の-1 倍代入
    return(muA);
  }
  matrix zero_sum_matrix(int a, int b, matrix m1AB){//交互作用ゼロ和
    vector [a-1] m1a; //行和を入れるベクトル
    vector [b-1] m1b; //列和を入れるベクトル
    matrix [a,b] muAB; //リターンする行列
    for(i in 1:(a-1)){ m1a[i] = 0.0;} //ゼロクリア
    for(j in 1:(b-1)){ m1b[j] = 0.0;} //ゼロクリア
    for(i in 1:(a-1)){ for(j in 1:(b-1)){muAB[i,j] =m1AB[i,j];}}//ソックリ代入
    for(i in 1:(a-1)){ for(j in 1:(b-1)){m1a[i] =m1a[i]+m1AB[i,j];}}//行和作る
    for(j in 1:(b-1)){ for(i in 1:(a-1)){m1b[j] =m1b[j]+m1AB[i,j];}}//列和作る
    for(i in 1:(a-1)){muAB[i,b] = (-1)*m1a[i]; } //第 b 列に行和の-1 倍代入

```

```

    for(j in 1:(b-1)){muAB[a,j] = (-1)*m1b[j]; } //第 a 行に列和の-1 倍代入
    muAB[a,b] = sum(m1a); //第 a 行第 b 列要素に合計代入、注意：マイナスではない
    return(muAB);
  }
}
data {
  int<lower=0> a; //A 水準数
  int<lower=0> b; //B 水準数
  int<lower=0> u[a,b]; //観察度数
  int<lower=0> n[a,b]; //試行数
}
parameters {
  real mu; //全平均
  vector [a-1] m1A; //A 平均 1 つ少ない
  vector [b-1] m1B; //B 平均 1 つ少ない
  matrix [a-1,b-1] m1AB; //交互作用 1 つ少ない
}
transformed parameters {
  matrix <lower=0>[a,b] lambda; //ラムダ
  vector [a] A; //A 主効果
  vector [b] B; //B 主効果
  matrix [a,b] AB; //AB 交互作用
  matrix [a,b] p; //比率
  A =zero_sum_vector(a, m1A); //制約 (5.28) 式
  B =zero_sum_vector(b, m1B); //制約 (5.29) 式
  AB=zero_sum_matrix(a, b, m1AB); //制約 (5.30) 式,(5.31) 式
  for(i in 1:a){
    for(j in 1:b){
      p[i,j] = exp(mu+ A[i] + B[j] + AB[i,j]);//(5.27) 式, 実験計画
      lambda[i,j] = n[i,j]*p[i,j] ; //(5.26) 式, λ の構造
    }
  }
}
model {
  for(i in 1:a){
    for(j in 1:b){
      u[i,j] ~ poisson(lambda[i,j]); //(5.26) 式, ポアソンモデル
    }
  }
}
';

par<-c("mu","A","B","AB","p") #母数
dataSet 分割表 02 <-list(a=nrow(u) ,b=ncol(u), u=u ,n=n ) #入力

#####cmdstanr による実行
modfile 分割表 02 <- write_stan_file(Poisson_Cross02) #書き出す一時ファイル
mod 分割表 02 <- cmdstan_model(modfile 分割表 02) #コンパイル
csrfit 分割表 02 <- mod 分割表 02$sample(data = dataSet 分割表 02,chains = 5,
  iter_sampling=20000, iter_warmup=1000, parallel_chains=5, seed=1234)
fit 分割表 02 <- rstan::read_stan_csv(csrfit 分割表 02$output_files()) #n 形式の変換

```

```
#####rstan による実行
#fit 分割表 02<-stan(model_code =Poisson_Cross02,data=dataset 分割表 02,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 分割表 02, file="./scrT/obje/stan_obje0506")
#load(file="./scrT/obje/stan_obje0506"); #予め stan( ) で作った object をロード

ext 分割表 02<-extract(fit 分割表 02, par); #乱数の取り出し

###表 5.17 タイタニック号事件の犠牲者の分析の母数・生成量の事後分布の要約
print(fit 分割表 02,digits_summary=3,prob=prob)

###表 5.18 水準・交互作用の効果が 0 より大きい (小さい) 確率
for (i in 1:4){ #表 5.18 左
  print(round(mean(ext 分割表 02$A[,i]>0),3))
  print(round(1-mean(ext 分割表 02$A[,i]>0),3))
}
print(round(mean(ext 分割表 02$B[,1]>0),3)) #表 5.18 中
print(round(1-mean(ext 分割表 02$B[,1]>0),3))
for (i in 1:4){
  print(round(mean(ext 分割表 02$AB[,i,1]>0),3)) #表 5.18 右
  print(round(1-mean(ext 分割表 02$AB[,i,1]>0),3))
}

###表 5.19 行の比率が列の比率より大きい phc
phc02(0,cbind(ext 分割表 02$p[,1,],ext 分割表 02$p[,2,],
ext 分割表 02$p[,3,],ext 分割表 02$p[,4,]))

#####実習課題
p_11<-ext 分割表 02$p[,1,1]
p_31<-ext 分割表 02$p[,3,1]
gqcal(p_11)
gqcal(p_31)
phc01(seq(0,1,0.01),p_31-p_11,cc="gtc",byoga="yes",xlab="死亡率のリスク差")
phc01(seq(0.38,0.48,0.01),p_31-p_11,cc="gtc",byoga="no")

phc01(seq(1,30,1),p_31/p_11,cc="gtc",byoga="yes",xlab="死亡率のリスク比")
phc01(seq(5,15,1),p_31/p_11,cc="gtc",byoga="no")
```

## 5.2 以下の関数を自分で復習しよう

dpois, barplot, paste,

## 5.3 ポアソン分布の推定の stan コード

```
u[i] ~ poisson(lambda); u ~ Poisson(λ)
```

```
uaste = poisson_rng(lambda);  $u^{*(t)} \sim \text{Poisson}(\lambda^{(t)})$ 
```

## 5.4 ポアソン回帰の stan コード

```
lambda[i] = exp(a + X[i]*b);  $\lambda_i = f(a + bx_i)$ 
y[i] ~ poisson(lambda[i]);  $u_i \sim \text{Poisson}(\lambda_i)$ 
yaste[i] = poisson_rng(lambda[i]);  $u_i^{*(t)} \sim \text{Poisson}(\lambda_i^{(t)})$ 
```

## 5.5 分割表の分析 01・度数の分解 (2\*2) の stan コード

```
lambda[1,1] = exp(mu+ A1 + B1 + AB11);
lambda[2,1] = exp(mu- A1 + B1 - AB11);
lambda[1,2] = exp(mu+ A1 - B1 - AB11);
lambda[2,2] = exp(mu- A1 - B1 + AB11);
```

$$\lambda_{ij} = \exp(\mu + a_i + b_j + (ab)_{ij}), \quad (i = 1, 2, j = 1, 2)$$

$$a_1 = -a_2, \quad b_1 = -b_2, \quad (ab)_{11} = -(ab)_{12} = -(ab)_{21} = (ab)_{22}$$

## 5.6 分割表の分析 02・比率の分解 (a\*b) の stan コード

「分割表の分析 01」のように実験計画の母数の制約を手動で入れることも可能であるが、モデルが複雑になると大変である。そこで自動的に制約を入れてくれる関数を導入する。

### 5.6.1 functions{ } ブロック

このブロックでは、以下のような自作関数を定義することができる。

### 5.6.2 zero\_sum\_vector: 主効果ゼロ和

```
vector zero_sum_vector(int a, vector m1A)
```

#### 引数

**int a** 主効果の和をゼロにしたい要因の水準数

**vector m1A** 主効果の和をゼロにしたい (サイズ a-1 の) 要因の効果

#### 戻り値

**muA** 和がゼロとなった要因の効果 (サイズは a)

### 5.6.3 zero\_sum\_matrix: 交互作用ゼロ和

```
matrix zero_sum_matrix(int a, int b, matrix m1AB)
```

#### 引数

**int a** 主効果の和をゼロにしたい要因 A の水準数

**int b** 主効果の和をゼロにしたい要因 B の水準数

**matrix m1AB** 和をゼロにしたい (サイズ (a-1) × (b-1) の) 交互作用効果

#### 戻り値

**muAB** 列和・行和がゼロになった (サイズ a × b の) 交互作用効果

### 5.6.4 その他

$$p[i,j] = \exp(\mu + A[i] + B[j] + AB[i,j]);$$

$$p_{ij} = \exp(\mu + a_i + b_j + (ab)_{ij})$$

$$\text{lambda}[i,j] = n[i,j] * p[i,j] ;$$
$$u[i,j] \sim \text{poisson}(\text{lambda}[i,j]);$$

$$u_{ij} \sim \text{Poisson}(\lambda_{ij} = n_{ij} \times p_{ij})$$

### 5.7 自習問題

スクリプトの量が多いので自習問題はなし。

### 5.8 第5章 宿題

「5.4 節 ポアソン回帰」p.67 の「表 5.12 鏡映描写課題における逸脱数」は説明のための人工データであったが、「鏡映描写逸脱数経過.csv」には、実際の鏡映描写実験の「逸脱数」が示されている。

「利」が「利き腕群」5 名、「非」が「非利き腕群」5 名、合計 10 名のデータが、1 行 1 人分の形式で収められている。変数 vi は、第 i 試行を意味し、第 3 試行から第 12 施行までである。

あなたの学籍番号の下 1 桁 \* に相当するプレイヤー s \* のデータを以下のように分析せよ。データは次のように読み込む

```
read.csv("scrT/dat/鏡映描写逸脱数経過.csv", header = T, row.names =1)
```

1. 分析するプレイヤーの番号、生データを示せ。
2. 「試行数」を予測変数、「逸脱数」を基準変数としてポアソン回帰分析をし、母数 a, b の事後分布を求めよ (表 5.13 上段参照、その際、予測変数の値が 3 から 12 であることに注意しなさい)。
3. 第 13 試行の逸脱数の母平均  $\lambda_{13}$  の事後分布の要約統計量 (表 5.13 下段参照) を求めよ。
4. 第 13 試行の逸脱数の事後予測分布の要約統計量 (表 5.13 下段参照) とヒストグラム (教科書にはない) を示しなさい。
5. 予測変数を 1 単位変化させたときの母数の変化率の事後分布の要約統計量 (表 5.13 下段参照) を求め、教科書の表現に従って解釈せよ。

## 6 数種の分布による独立した1要因の推測

### 6.1 スクリプト

#####第6章 数種の分布による独立した1要因の推測

```
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())           #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義
```

#####6.1 対数正規分布

####図 6.1 対数正規分布の確率密度関数

```
par(mfrow=c(1,2))
for (i in c(0.5,1.0,1.8)){
  d1<-function(x){dlnorm(x,0,i)}
  curve(d1,0,3,ylim=c(0,1.2),xlab="",ylab="",cex.axis=2.0)
  par(new=T)
  text(1.3,0.8,"σ=0.5",cex=2)
  text(0.8,0.5,"σ=1.0",cex=2)
  text(0.4,1.1,"σ=1.8",cex=2)
  text(2.3,0.8,"μ=0",cex=2.5);          #1 枚目ここまで
  par(new=F)
  for (i in c(0.0,0.2,0.4,0.6)){
    d1<-function(x){dlnorm(x,i,1)}
    curve(d1,0,4,ylim=c(0,0.7),xlab="",ylab="",cex.axis=2.0)
    par(new=T)
    text(1.0,0.6,"μ=0.0",cex=2)
    text(1.2,0.5,"μ=0.2",cex=2)
    text(1.4,0.4,"μ=0.4",cex=2)
    text(0.8,0.3,"μ=0.6",cex=2)
    text(3.0,0.50,"σ=1.0",cex=2.5)
    par(new=F)
  }
  par(mfrow=c(1,1));          #図 6.1 ここまで
#dev.copy2eps(file="z0501.eps",family="Japan1")
```

#####6.1.1 数値例

### 図 6.2 ある集団の年収の分布

```
hist(rlnorm(10000,15.3,0.5)/10000,xlim=c(0,2000),breaks=30,
     xlab="万円",ylab="",cex.axis=2.0,cex.lab=2.0)
```

### p.77 下部から p.78 上部

```
exp(15.3+0.5*0.5^2);          #平均 (6.3) 式
```

```
exp(15.3);                     #中央値 (6.5) 式
exp(15.3-0.5^2);              #最頻値 (6.6) 式
qlnorm(0.95, 15.3,0.5);       #%点
qlnorm(0.99, 15.3,0.5)
qlnorm(0.999,15.3,0.5)
```

#####6.1.2 対数正規分布の推定

#####表 6.1 職種 I の 30 人の年収 (万円)

```
x1<-c(367, 1611, 263, 405, 754, 415, 837, 341, 396, 342,
      282, 296, 886, 412, 572, 471, 781, 531, 757, 723,
      870, 933, 392, 612, 394, 343, 372, 747, 280, 941)
```

#####対数正規分布の推定の stan コード

```
lognormal01<-'
data {
  int<lower=0> n;          //データ数
  real<lower=0> x[n];      //データ
}
parameters {
  real mu;                //母数 μ (6.1) 式
  real<lower=0> sigma;     //母数 σ (6.1) 式
}
model {
  for(i in 1:n)
    x[i] ~ lognormal(mu,sigma); //対数正規分布 (6.2) 式, (6.8) 式
}
generated quantities{
  real heikin;            //平均値
  real tyuouti;           //中央値
  real saihin;            //最頻値
  real sigma0;            //標準偏差
  heikin = exp(mu+0.5*sigma^2); // (6.3) 式
  tyuouti = exp(mu);       // (6.5) 式
  saihin = exp(mu-sigma^2); // (6.6) 式
  sigma0=sqrt(exp(2*mu+sigma^2)*(exp(sigma^2)-1)); // (6.4) 式
}
';
```

```
par<-c("mu","sigma","heikin","tyuouti","saihin","sigma0") #母数
dataSet 対1 <-list(n=length(x1),x=x1)                       #入力
```

#####cmdstanr による実行

```
modfile 対1 <- write_stan_file(lognormal01)                #書き出す一時ファイル
mod 対1 <- cmdstan_model(modfile 対1)                       #コンパイル
csrfit 対1 <- mod 対1$sample(data = dataSet 対1,chains = 5,iter_sampling = 20000,
                             iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 対1 <- rstan::read_stan_csv(csrfit 対1$output_files()) #stan 形式への変換
```

#####rstan による実行

```
#fit 対1<-stan(model_code =lognormal01,data=dataSet 対1,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 対1, file="./scrT/obje/stan_obje0601")
```

```
#load(file="./scrT/obje/stan_obje0601"); #予め stan( ) で作った object をロード

ext 対 1<-extract(fit 対 1, par);          #乱数の取り出し

##### 表 6.2 職種 I の母数と生成量の事後分布の要約
gqcal(ext 対 1$mu)
gqcal(ext 対 1$sigma)
gqcal(ext 対 1$heikin ,0)
gqcal(ext 対 1$tyuouti,0)
gqcal(ext 対 1$saihin ,0)
gqcal(ext 対 1$sigma0 ,0)

#####6.1.3 2つの対数正規分布の比較
#####表 6.3 職種 II の 30 人の年収(万円)
x2<-c(697, 681, 330, 176, 307, 1668, 641, 1150, 301, 365,
      1033, 373, 836, 367, 265, 217, 941, 278, 504, 299,
      865, 504, 790, 273, 2292, 385, 277, 622, 475,1205)

##### 2つの対数正規分布の推測の stan コード
lognormal02<- '
data {
  int<lower=0> n1;          //データ数 I
  int<lower=0> n2;          //データ数 II
  real<lower=0> x1[n1];     //データ I
  real<lower=0> x2[n2];     //データ II
}
parameters {
  real mu[2];              //母数  $\mu$  1,  $\mu$  2 (6.8) 式
  real<lower=0> sigma[2];   //母数  $\sigma$  1,  $\sigma$  2 (6.8) 式
}
transformed parameters {
}
model {
  for(i in 1:n1) x1[i] ~ lognormal(mu[1],sigma[1]); //対数正規分布
  for(i in 1:n2) x2[i] ~ lognormal(mu[2],sigma[2]); //(6.8) 式
}
generated quantities{
  real heikin[2];          //平均値 2つ
  real tyuouti[2];         //中央値 2つ
  real saihin[2];          //最頻値 2つ
  real sigma0[2];          //標準偏差 2つ
  heikin[1] = exp(mu[1]+0.5*sigma[1]^2);           // (6.3) 式 I
  tyuouti[1]= exp(mu[1]);                           // (6.5) 式 I
  saihin[1] = exp(mu[1]-sigma[1]^2);               // (6.6) 式 I
  sigma0[1]=sqrt(exp(2*mu[1]+sigma[1]^2)*(exp(sigma[1]^2)-1))); // (6.4) 式 I
  heikin[2] = exp(mu[2]+0.5*sigma[2]^2);           // (6.3) 式 II
  tyuouti[2]= exp(mu[2]);                           // (6.5) 式 II
  saihin[2] = exp(mu[2]-sigma[2]^2);               // (6.6) 式 II
  sigma0[2]=sqrt(exp(2*mu[2]+sigma[2]^2)*(exp(sigma[2]^2)-1))); // (6.4) 式 II
}
';
```

```
par<-c("mu","sigma","heikin","tyuouti","saihin","sigma0") #母数
dataSet 対 2 <-list(n1=length(x1),n2=length(x2),x1=x1,x2=x2) #入力

#####cmdstanr による実行
modfile 対 2 <- write_stan_file(lognormal02)          #書き出す一時ファイル
mod 対 2 <- cmdstan_model(modfile 対 2)              #コンパイル
csrfit 対 2 <- mod 対 2$sample(data = dataSet 対 2,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 対 2 <- rstan::read_stan_csv(csrfit 対 2$output_files()) #stan 形式への変換

#####rstan による実行
#fit 対 2<-stan(model_code =lognormal02,data=dataSet 対 2,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit 対 2, file="./scrT/obje/stan_obje0602")
#load(file="./scrT/obje/stan_obje0602"); #予め stan( ) で作った object をロード

ext 対 2<-extract(fit 対 2, par);          #乱数の取り出し

#####表 6.4 職種 II の母数と生成量の事後分布の要約
gqcal(ext 対 2$mu[,2])
gqcal(ext 対 2$sigma[,2])
gqcal(ext 対 2$heikin[,2] ,0)
gqcal(ext 対 2$tyuouti[,2],0)
gqcal(ext 対 2$saihin[,2] ,0)
gqcal(ext 対 2$sigma0[,2] ,0)

#表 6.5 2つの対数正規分布の位置の差の事後分布の要約 平均値 中央値 最頻値の差
gqcal(ext 対 2$heikin[,2]-ext 対 2$heikin[,1])
gqcal(ext 対 2$tyuouti[,2]-ext 対 2$tyuouti[,1])
gqcal(ext 対 2$saihin[,2]-ext 対 2$saihin[,1])

#####表 6.6 平均値, 中央値, 最頻値に差のある phc
sequ66<-c(0,10,20,30)
phc01(sequ66,ext 対 2$heikin[,2] ,ext 対 2$heikin[,1] ,cc="gtc",byoga="no")
phc01(sequ66,ext 対 2$tyuouti[,2],ext 対 2$tyuouti[,1],cc="gtc",byoga="no")
phc01(sequ66,ext 対 2$saihin[,1] ,ext 対 2$saihin[,2] ,cc="gtc",byoga="no")

#####表 6.7 平均値, 中央値, 最頻値が ROPE である phc
sequ67<-c(0,50,100,120)
phc01(sequ67,ext 対 2$heikin[,2] ,ext 対 2$heikin[,1] ,cc="rope",byoga="no")
phc01(sequ67,ext 対 2$tyuouti[,2],ext 対 2$tyuouti[,1],cc="rope",byoga="no")
phc01(sequ67,ext 対 2$saihin[,1] ,ext 対 2$saihin[,2] ,cc="rope",byoga="no")

#####6.2 対数正規分布による 1 要因実験
#####表 6.9 鏡映描写課題の群ごとの 15 試行目のタイムと逸脱数
(鏡映描写 01<-read.csv("./scrT/dat/鏡映描写 01.csv", header = TRUE))

#####図 6.3 鏡映描写課題における群ごとのタイム(秒)
boxplot(鏡映描写 01$time~鏡映描写 01$group,          #図 6.3
  xlab="group", ylab="",cex.axis=2.0,cex.lab=2.0)
#dev.copy2eps(file="z0503.eps",family="Japan1")
```

```
tapply(鏡映描写 01$time, 鏡映描写 01$group,mean); #群ごとの平均タイム p.82
```

```
#####対数正規分布による1要因実験のstanコード
one_factor_lognormal<- '
data {
  int<lower=0> n;          //全データ数
  int<lower=0> J;          //群数
  vector[n] y;            //基準変数
  int<lower=0> k[n];       //分類変数
}
parameters {
  vector[J] mu;           //各群の mu J個
  vector<lower=0>[J] sigma; //各群の sigma J個
}
model {
  for(i in 1:n)
    y[i] ~ lognormal(mu[k[i]], sigma[k[i]]); //(6.14) 式
}
generated quantities{
  vector[J] mu0;          //各群の平均 J個
  vector[J] medi0;        //各群の中央値 J個
  vector[J] mode0;        //各群の最頻値 J個
  vector<lower=0>[J] sigma0; //各群の標準偏差 J個
  for(j in 1:J){
    mu0[j] =exp(mu[j]+(0.5*sigma[j]^2)); // (6.3) 式
    medi0[j] =exp(mu[j]); // (6.5) 式
    mode0[j] =exp(mu[j]-sigma[j]^2); // (6.6) 式
    sigma0[j]=sqrt(exp(2*mu[j]+sigma[j]^2)*(exp(sigma[j]^2)-1)); // (6.4) 式
  }
}
';
par<-c("mu","sigma","mu0","medi0","mode0","sigma0") #母数
dataSetlogn <-list(n=length(鏡映描写 01$time),
  J=max(鏡映描写 01$group),y=鏡映描写 01$time, k=鏡映描写 01$group ) #入力

#####cmdstanr による実行
modfilelogn <- write_stan_file(one_factor_lognormal) #書き出す一時ファイル
modlogn <- cmdstan_model(modfilelogn) #コンパイル
csrfitlogn <- modlogn$sample(data=dataSetlogn,chains=5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitlogn <- rstan::read_stan_csv(csrfitlogn$output_files()) #stan形式への変換

#####rstan による実行
#fitlogn<-stan(model_code =one_factor_lognormal,data=dataSet02,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitlogn, file="./scrT/obje/stan_obje0603")
#load(file="./scrT/obje/stan_obje0603"); #予め stan( ) で作った object をロード

extlogn<-extract(fitlogn, par); #乱数の取り出し

##### 表 6.10 1 要因対数正規分布モデルの母数の事後分布
```

```
print(fitlogn,digits_summary=3,prob=prob)
```

```
##### 図 6.4 鏡映描写課題・群ごとのタイムの分布(対数正規分布)
lognor01<-function(x){dlnorm(x,3.153, 0.685)}
lognor02<-function(x){dlnorm(x,2.981, 0.449)}
lognor03<-function(x){dlnorm(x,2.822, 0.488)}
curve(lognor01,0,80,ylim=c(0,0.055),xlab="",ylab="",lty=1,lwd=2)
par(new=T)
curve(lognor02,0,80,ylim=c(0,0.055),xlab="",ylab="",lty=2,lwd=2)
par(new=T)
curve(lognor03,0,80,ylim=c(0,0.055),xlab="time",
  ylab="確率密度",lty=3,lwd=2)
text(60,0.01,"第1群",cex=1.6)
text(30,0.04,"第2群",cex=1.6)
text( 5,0.05,"第3群",cex=1.6) #図 6.4 ここまで
```

```
#表 6.11 行 j の水準の平均が列 j の水準の平均より大きい確率
phc02(0,extlogn$mu0)
```

```
#休憩群が非利き手群より長く、かつ 非利き手群が利き手群より長い確率
round(mean((extlogn$mu0[,1]>extlogn$mu0[,2])&
  (extlogn$mu0[,2]>extlogn$mu0[,3])),3) # (6.17) 式, (6.18) 式
```

```
#####6.3 ポアソン分布による1要因実験
#####表 6.9 鏡映描写課題の群ごとの15 試行目のタイムと逸脱数
(鏡映描写 01<-read.csv("./scrT/dat/鏡映描写 01.csv", header = TRUE))
```

```
##### 図 6.5 鏡映描写課題における群 逸脱数ごとの人数の棒グラフ
barplot(table(鏡映描写 01$逸脱, 鏡映描写 01$group),
  xlab="群ごとの逸脱数",ylab="人数",beside=T,legend.text=0:5 )
```

```
#群ごとの平均逸脱数 p.84,b1.8
tapply(鏡映描写 01$逸脱, 鏡映描写 01$group,mean)
```

```
#####ポアソン分布による1要因実験のstanコード
one_factor_Poisson<- '
data {
  int<lower=0> n;          //全データ数
  int<lower=0> J;          //群数
  int<lower=0> y[n];       //基準変数
  int<lower=0> k[n];       //分類変数
}
parameters {
  vector<lower=0>[J] lambda; //各群の lambda J個
}
model {
  for(i in 1:n)
    y[i] ~ poisson(lambda[k[i]]); // (6.19) 式
}
';
```

```

par<-c("lambda") #母数
dataSetpo <-list(n=length(鏡映描写 01$逸脱),
  J=max(鏡映描写 01$group),y=鏡映描写 01$逸脱, k=鏡映描写 01$group ) #入力

#####cmdstanr による実行
modfilepo <- write_stan_file(one_factor_Poisson) #書き出す一時ファイル
modpo <- cmdstan_model(modfilepo) #コンパイル
csrfitpo <- modpo$sample(data = dataSetpo,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitpo <- rstan::read_stan_csv(csrfitpo$output_files()) #stan 形式への変換

#####rstan による実行
#fitpo<-stan(model_code =one_factor_Poisson,data=dataSet02,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitpo, file="./scrT/obje/stan_obje0604")
#load(file="./scrT/obje/stan_obje0604"); #予め stan( ) で作った object をロード

extpo<-extract(fitpo, par); #乱数の取り出し

#### 表 6.12 1 要因ポアソン分布モデルの母数の事後分布
print(fitpo,digits_summary=3,prob=prob)

#### 図 6.6 鏡映描写実験・群ごとの逸脱数の分布 (ポアソン分布)
x<-0:5
plot(x,dpois(x,1.100),type="o",ylim=c(0,0.7),xlab="",ylab="",lty=1,lwd=2)
par(new=T)
plot(x,dpois(x,0.666),type="o",ylim=c(0,0.7),xlab="",ylab="",lty=2,lwd=2)
par(new=T)
plot(x,dpois(x,0.401),type="o",ylim=c(0,0.7),
  xlab="逸脱数",ylab="確率",lty=3,lwd=2)
legend(3,0.6,paste("第",1:3,"群"),lty=1:3) #図 6.6 ここまで

##表 6.13 行 j の水準の平均が列 j' の水準の平均より大きい確率
phc02(0,extpo$lambda)

##休憩群が非利き手群より長く、かつ 非利き手群が利き手群より長い確率
round(mean((extpo$lambda[,1]>extpo$lambda[,2])
  &(extpo$lambda[,2]>extpo$lambda[,3])),3) # (6.20) 式

#####6.4 2 項分布による 1 要因実験
####表 6.14 会話促進に関する訓練における群ごとの基本語数と表現数
(会話促進<-read.csv("./scrT/dat/会話促進.csv",header = TRUE))

####図 6.7 会話促進訓練における群ごとの表現率
boxplot(会話促進$表現数/会話促進$基本語数~会話促進$group,
  xlab="group", ylab="", cex.axis=2.0,cex.lab=2.0)
#dev.copy2eps(file="z0507.eps",family="Japan1")

####群ごとの表現率の平均 p.86,b1.7
round(tapply(会話促進$表現数/会話促進$基本語数, 会話促進$group,mean),3)

```

```

#####2 項分布による 1 要因実験の stan コード
one_factor_binom<-'
data {
  int<lower=0> n; //全データ数
  int<lower=0> J; //群数
  int<lower=0> y[n]; //基準変数
  int<lower=0> m[n]; //試行数
  int<lower=0> k[n]; //分類変数
}
parameters {
  vector<lower=0,upper=1>[J] p; //各群の確率 J 個
}
model {
  for(i in 1:n)
    y[i] ~ binomial(m[i],p[k[i]]); //(6.21) 式
}
';

par<-c("p") #母数
dataSetbi <-list(n=length(会話促進$表現数),J=max(会話促進$group),
  y=会話促進$表現数, m=会話促進$基本語数, k=会話促進$group ) #入力

#####cmdstanr による実行
modfilebi <- write_stan_file(one_factor_binom) #書き出す一時ファイル
modbi <- cmdstan_model(modfilebi) #コンパイル
csrfitbi <- modbi$sample(data = dataSetbi,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitbi <- rstan::read_stan_csv(csrfitbi$output_files()) #stan 形式への変換

#####rstan による実行
#fitbi<-stan(model_code =one_factor_binom,data=dataSetbi,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitbi, file="./scrT/obje/stan_obje0605")
#load(file="./scrT/obje/stan_obje0605"); #予め stan( ) で作った object をロード

extbi<-extract(fitbi, par); #乱数の取り出し

#### 表 6.15 1 要因 2 項分布モデルの母数の事後分布
print(fitbi,digits_summary=3,prob=prob)

#### 図 6.8 EAP 推定値による各群の 2 項分布
x<-0:10
plot(x,dbinom(x,10,0.338),type="o",ylim=c(0,0.3),xlab="",ylab="",lty=1,lwd=2)
par(new=T)
plot(x,dbinom(x,10,0.510),type="o",ylim=c(0,0.3),xlab="",ylab="",lty=2,lwd=2)
par(new=T)
plot(x,dbinom(x,10,0.613),type="o",ylim=c(0,0.3),
  xlab="表現数",ylab="確率",lty=3,lwd=2)
legend(8,0.27,paste("第",1:3,"群"),lty=1:3); #図 6.8 ここまで

```

```

###表 6.16 行 j の水準の比率が列 j' の水準の比率より大きい phc
phc02(0,extbi$p)

#連言命題が正しい確率 (仮説 C に対応 教科書にはない)
round(mean((extbi$p[,1]<extbi$p[,2])&(extbi$p[,2]<extbi$p[,3])),3)

#####6. 5 正規分布による 1 要因実験 (変量モデル)
###表 6.17 陸上 100m 種目の選手 6 名の 10 回の成績
(M100 走<-read.csv("./scrT/dat/M100 走.csv", header = TRUE))

####図 6.9 陸上 100m 種目の選手 6 名の 10 回のタイムのボックスプロット
boxplot(M100 走$time~M100 走$選手, xlab="選手番号", ylab="time");#図 6.9
#dev.copy2eps(file="z0501.eps",family='Japan1')

#####正規分布による 1 要因実験 (変量モデル) の stan コード
one_factor_normal<- '
data {
  int<lower=0> n; //全データ数
  int<lower=0> J; //群数
  vector[n] y; //基準変数
  int<lower=0> k[n]; //分類変数
}
parameters {
  vector[J] mu; //各群の平均 J 個
  real<lower=0> sigma; //誤差 SD
  real<lower=0> s_mu; //要因 SD
  real mu0; //全平均
}
model {
  y ~ normal(mu[k], sigma); //(6.24) 式
  mu ~ normal(mu0, s_mu); //(6.25) 式
}
generated quantities{
  real<lower=0,upper=1> eta2; //説明率
  eta2 = (s_mu^2)/((s_mu^2)+(sigma^2)); //(6.27) 式
}
';

par<-c("mu","sigma","s_mu","mu0","eta2") #母数
dataSetnorm <-list(n=length(M100 走$time),
  J=max(M100 走$選手),y=M100 走$time, k=M100 走$選手 ) #入力

#####cmdstanr による実行
modfilenorm <- write_stan_file(one_factor_normal) #書き出す一時ファイル
modnorm <- cmdstan_model(modfilenorm) #コンパイル
csrfitnorm <- modnorm$sample(data=dataSetnorm,chains=5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitnorm <- rstan::read_stan_csv(csrfitnorm$output_files()) #stan 形式への変換

#####rstan による実行
#fitnorm<-stan(model_code =one_factor_normal,data=dataSetnorm,

```

```

# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitnorm, file="./scrT/obje/stan_obje0606")
#load(file="./scrT/obje/stan_obje0606"); #予め stan( ) で作った object をロード

extnorm<-extract(fitnorm, par); #乱数の取り出し

##### 表 6.18 変量モデルの母数の事後分布の要約
print(fitnorm,digits_summary=3,prob=prob)

##### 表 6.19 行の選手の平均が列の選手の平均より大きい確率
phc02(0,extnorm$mu)

#####実習課題
### 1.

#
# 正解は割愛
#

### 2.

#
# 正解は割愛
#

### 3.

#
# 正解は割愛
#

### 4.
c1<-c(0,1,2)
c2<-c(0,1,2)
rengen_01<-matrix(0,length(c1),length(c2));
ii<-0;
for (i in c1){
  ii<-ii+1
  jj<-0
  for (j in c2){
    jj<-jj+1
    rengen_01[ii,jj]<-mean((i < heikin_sa12 )&
      (j < heikin_sa23 ) )
  }
}
rownames(rengen_01)<-c1
colnames(rengen_01)<-c2
round(rengen_01,3)

### 5.
lambda_sa12 <- extpo$lambda[,1]-extpo$lambda[,2]
lambda_sa23 <- extpo$lambda[,2]-extpo$lambda[,3]

```



```
phc01(seq(0,1,0.05),lambda_sa12,cc="gtc",byoga="yes",
      xlab="休憩群と非利き手群の差")
phc01(seq(0,1,0.05),lambda_sa23,cc="gtc",byoga="yes",
      xlab="非利き手群と利き手群の差")
phc01(seq(0,0.3,0.05),lambda_sa12,cc="gtc",byoga="no")
phc01(seq(0,0.2,0.05),lambda_sa23,cc="gtc",byoga="no")
```

```
### 6.
c1<-c(0,0.1,0.2)
c2<-c(0,0.1,0.2)
rengen_02<-matrix(0,length(c1),length(c2));
ii<-0;
for (i in c1){
  ii<-ii+1
  jj<-0
  for (j in c2){
    jj<-jj+1
    rengen_02[ii,jj]<-mean((i < lambda_sa12 )&
                          (j < lambda_sa23 ))
  }
}
rownames(rengen_02)<-c1
colnames(rengen_02)<-c2
round(rengen_02,3)
```

```
### 7.
hiritu_sa21 <- extbi$p[,2]-extbi$p[,1]
hiritu_sa32 <- extbi$p[,3]-extbi$p[,2]
phc01(seq(0,0.3,0.01),hiritu_sa21,cc="gtc",byoga="yes",
      xlab="2 か月と 1 か月の差")
phc01(seq(0,0.2,0.01),hiritu_sa32,cc="gtc",byoga="yes",
      xlab="3 か月と 2 か月の差")
phc01(seq(0,0.15,0.03),hiritu_sa21,cc="gtc",byoga="no")
phc01(seq(0,0.1,0.02),hiritu_sa32,cc="gtc",byoga="no")
```

```
### 8.
eta2<-extnorm$eta2
phc01(seq(0.1,0.8,0.05),eta2,cc="gtc",byoga="yes",
      xlab="説明率")
phc01(seq(0,3,0.02),extnorm$mu[,1]-extnorm$mu[,2],cc="gtc",byoga="yes",
      xlab="選手 1 と選手 2 の差")
phc01(seq(0,0.15,0.03),eta2,cc="gtc",byoga="no")
phc01(seq(0,1,0.2),extnorm$mu[,1]-extnorm$mu[,2],cc="gtc",byoga="no")
```

```
#####鏡映描写実験データ
#タイム
休 time<-c(40.03,35.04,32.38,23.25,24.45,
           30.66,32.26,21.18,25.53,37.66,
           22.51,33.46,14.87,23.52,26.63,
```

```
           35.27,54.38,17.59,39.75,29.63,
           36.72,26.77,24.67,61.77,31.28,
           27.83,27.78,33.15)
非 time<-c(21.39,13.89,24.76,24.20,17.77,
           24.90,24.15,13.56,21.86,17.58,
           34.03,20.41,19.09,27.90,29.91,
           20.88,26.44,21.29,10.71,21.92,
           17.09,31.98,31.18,18.37,16.53,
           27.40,26.03)
利 time<-c(08.63,12.40,15.11,23.68,16.28,
           29.56,25.40,12.53,19.00,21.59,
           21.93,23.03,17.09,11.37,23.84,
           22.03,18.98,12.40,25.77,12.62,
           13.78,10.43,24.49,34.28,13.74,
           18.22,18.56,21.22)
```

```
#逸脱回数
休 freq<-c(1,1,0,2,1,0,2,2,2,2,0,4,2,0,1,2,2,1,6,5,1,2,3,1,0,1,2,2)
非 freq<-c(2,0,0,0,1,2,0,3,0,1,2,1,0,0,1,0,1,2,0,0,2,1,1,0,2,4,1)
利 freq<-c(0,1,0,1,1,0,1,1,0,1,0,0,1,2,0,1,0,0,0,0,0,0,0,0,0,0,1,0)
```

## 6.2 以下の関数を自分で復習しよう

dlnorm, qlnorm, tapply,

## 6.3 複数ある母数の番号を、分類変数のベクトルの値で指示する

対数正規分布による 1 要因実験の事後分布で (6.14) 式には、

$$f(\mu_1, \mu_2, \mu_3, \sigma_1, \sigma_2, \sigma_3 | \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ \propto \prod_{j=1}^3 \left[ \prod_{i=1}^{n_j} \log\_normal(x_{ij} | \mu_j, \sigma_j) \right] f(\mu_j) f(\sigma_j) \quad (6.14)$$

のように群を表現する添え字  $j$  と、群内の観測対象を表現する添え字  $i$  のふたつがある。

しかし対応する stan コードには

```
model {
  for(i in 1:n)
    y[i] ~ lognormal(mu[k[i]], sigma[k[i]]); //(6.14) 式
}
```

のように添え字は  $i$  のみで、群を表現する添え字も、群内を表現する添え字もない。この理由を説明する。

$k[i]$  は、1,2,3 のどれかの値をとり、第  $i$  番目の観測対象がどの群に属しているのかを表現している。 $\mu_j$  は  $\text{mu}[j]$  であり、 $\text{mu}[k[i]]$  は、 $\text{mu}[1], \text{mu}[2], \text{mu}[3]$  のどれかを表現している。ゆえに添え字は2つは必要なく、 $i$  が1から  $n$  まで動けばよい。 $\text{sigma}[k[i]]$  も同様である。

母数ベクトル [ 分類変数ベクトル [ 添え字 ] ]

という形式のこのテクニックは、今後、頻繁に多用される。モデル式の添え字が3つに増えても、スクリプトには添え字が1つだけ、ということもあるので、しっかりと理解していただきたい。

## 6.4 phc02: phc の行列表示

```
phc02(c=0, ext, cc="gtc", digits=3)
```

### 引数

**c** スカラー 基準点

**ext** 数値行列、行に乱数、列に母数をもつ乱数行列

**cc** "gtc" の場合は 行の母数一列の母数  $> c$  であり、  
"rope" の場合は  $\text{abs}(\text{行の母数一列の母数}) < c$  であり、  
それ以外は ("ltc" が想定されるが) 行の母数一列の母数  $< c$

**dedits** 整数、出力される小数点以下の桁数

### 戻り値

**av** phc の行列

## 6.5 自習問題

- 6.1.2 項「年収の中央値は  $c$  円より大きい」の phc 曲線と phc テーブルを作成しなさい。
- 6.1.3 項「職種 II のほうが職種 I より年収の最頻値が  $c$  円高い」の phc 曲線と phc テーブルを作成しなさい。
- 6.2 節、(6.15) 式と (6.16) 式の phc テーブルを作成しなさい。

## 6.6 第6章 宿題

「6.3 節 ポアソン分布による1要因実験」p.84の「図6.5 鏡映描写課題における群×逸脱数ごとの人数の棒グラフ」は説明のための人工データであったが、'鏡映描写3群逸脱数.R'には、実際の鏡映描写実験の「逸脱数」が示されている。データは次のように読み込む

```
source("scrT/dat/鏡映描写3群逸脱数.R")
```

「利」が「利き腕群」、「非」が「非利き腕群」、「休」が「休息群」のデータベクトルであり、第13試行から第15施行までを区別せず収めてある。以下の手順で分析せよ。

- "鏡映描写 01.csv" を参考に、入力すべきデータ成形して、ポアソン分布による1要因3水準実験の分析をする。
- 表6.12を参考にして、3つのポアソン分布の母数の事後分布の数値要約を示せ。
- 図6.6を参考にして、3つのポアソン分布の確率関数を折れ線で描きなさい。
- 上巻 p.137 の仮説 A に相当する  $\text{phc}(c < \lambda \text{休憩} - \lambda \text{非利き手})$  の曲線とテーブルを描き、解釈せよ。
- 上巻 p.137 の仮説 B に相当する  $\text{phc}(c < \lambda \text{非利き手} - \lambda \text{利き手})$  の曲線とテーブルを描き、解釈せよ。
- 第I巻 p.137 の仮説 C に相当する、第II巻 (6.20) 式の phc テーブルを作成せよ。 $c_1=(0.1,0.2,0.3)$ ,  $c_2=(0.1,0.2,0.3)$  の組み合わせで  $3 \times 3$  のテーブルを描き、解釈せよ。

## 7 共分散分析/傾向スコア

### 7.1 スクリプト

```
#####第7章 共分散分析/傾向スコア
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())           #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
library(pROC)
library(Matching)
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

##### 第7章で利用する関数ここから
#2 群の2変量データの散布図をpchを変えて描く関数
plot2g<-function(y,x,a,ba=5,xlab="pre test"){
  plot(x,y,pch=ifelse(a-1,1,16),xlab=xlab,ylab="post test",
       cex.lab=1.5,cex.axis=1.5,xlim=c(min(x)-ba,max(x)+ba),
       ylim=c(min(y)-ba,max(y)+ba))
}
#2 群の散布図と2本の傾き共通の回帰直線を描く関数
plot2gequ<-function(y,x,a,ext,ba=5,xlab="pre test"){
  plot(x,y,pch=ifelse(a-1,1,16),xlab=xlab,ylab="post test",
       cex.lab=1.5,cex.axis=1.5,xlim=c(min(x)-ba,max(x)+ba),
       ylim=c(min(y)-ba,max(y)+ba))
  abline(mean(ext$a[,1]),mean(ext$b[,1]))
  abline(mean(ext$a[,2]),mean(ext$b[,2]))
}
#標本分散
bunsan<-function(x){mean((x-mean(x))^2)}

#散布図に傾きの異なった2本の回帰直線を描く関数
plot2gdef<-function(y,x,a,ext,ba=3){
  plot(x,y,pch=ifelse(a-1,1,16),xlab="pre test",ylab="post test",
       cex.lab=1.5,cex.axis=1.5,xlim=c(min(x)-ba,max(x)+ba),
       ylim=c(min(y)-ba,max(y)+ba))
  abline(mean(ext$a[,1]),mean(ext$b[,1]))
  abline(mean(ext$a[,2]),mean(ext$b[,2]))
}
#傾きの異なった2本の回帰直線の予測値の差のEAPと95%確信区間を描く関数
plot2gdefdef<-function(x,ext){
  leng=30
  xlab<-seq(min(x),max(x),length=leng)
  yhat_sa<-matrix(0,nrow(ext$a),leng)
  yhat_sa_HL<-matrix(0,leng,2)
  for (i in (1:leng)){
```

```
    yhat_sa[i]<-ext$a[,1]-ext$a[,2]+(ext$b[,1]-ext$b[,2])*xlab[i]}
  for (i in (1:leng)){
    yhat_sa_HL[i,]<-quantile(yhat_sa[i],prob=c(0.025,0.975))}
  plot(xlab,colMeans(yhat_sa),ylim=c(min(yhat_sa_HL),max(yhat_sa_HL)),
       type="l",ylab="",xlab="",lwd=2.0)
  abline(a=0,b=0,lwd=2.0)
  par(new=T)
  plot(xlab,yhat_sa_HL[,1],ylim=c(min(yhat_sa_HL),max(yhat_sa_HL)),
       type="l",ylab="",xlab="",lty=2,lwd=2.0)
  par(new=T)
  plot(xlab,yhat_sa_HL[,2],ylim=c(min(yhat_sa_HL),max(yhat_sa_HL)),
       type="l",ylab="post の予測値の差",xlab="pre 値",lty=2,lwd=2.0)
}
```

#####第7章で利用する関数ここまで

#### ##### 7.2 傾きが共通した共分散分析

##### ##### 7.2.1 「データ1」の分析

```
##### 表7.1 「データ1」の平均値
学習01<-read.csv("./scrT/dat/学習01.csv", header = TRUE)
round(tapply(学習01$post, 学習01$subject,mean),1)
round(tapply(学習01$pre , 学習01$subject,mean),1)
```

```
##### 図7.1 「データ1」の散布図
plot2g(y=学習01$post,x=学習01$pre,a=学習01$subject,ba=3)
#dev.copy2eps(file="z0701.eps",family='Japan1')
```

#####共分散分析(傾きが共通のモデル)のstanコード

```
ANCOVAequ<-'
data {
  int<lower=0> n;           //データ数
  vector[n] y;             //基準変数
  vector[n] x;             //予測変数
  int<lower=0,upper=2> k[n]; //分類変数
}
parameters {
  vector[2] a;             //切片
  real b;                  //回帰係数
  real<lower=0> sigma;      //誤差標準偏差
}
transformed parameters {
  vector[n] yhat;          //予測値 yhat
  for(i in 1:n)
    yhat[i] = a[k[i]] + x[i]*b; // (7.4) 式, (7.5) 式
}
model {
  y ~ normal(yhat, sigma); // (7.1) 式, (7.2) 式, (7.3) 式
}
';

par<-c("a","b","sigma"); #母数
```

```

dataSetequ01<-list(n=nrow(学習 01),y=学習 01[,2],x=学習 01[,1],k=学習 01[,3]);#入力

#####cmdstanr による実行
modfileequ01 <- write_stan_file(ANCOVAequ)          #書き出す一時ファイル
modequ01 <- cmdstan_model(modfileequ01)             #コンパイル
csrfitequ01<-modequ01$sample(data=datasetequ01,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitequ01 <- rstan::read_stan_csv(csrfitequ01$output_files()) #stan 形式への変換

#####rstan による実行
#fitequ01<-stan(model_code =ANCOVAequ,pars=par,data=datasetequ01,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitequ01, file="./scrT/obje/stan_obje0701")
#load(file="./scrT/obje/stan_obje0701"); #予め stan( ) で作った object をロード

extequ01<-extract(fitequ01, par);                  #乱数の取り出し

##### 表 7.2 「データ 1」の傾きが共通したモデルの母数の事後分布の要約
gqcal(extequ01$a[,1],2)
gqcal(extequ01$a[,2],2)
gqcal(extequ01$b      ,2)
gqcal(extequ01$sigma,2)

##### 図 7.2 「データ 1」の回帰直線 (傾き共通)
plot2gequ(y=学習 01$post,x=学習 01$pre,
  a=学習 01$subject,ext=extequ01,ba=3);
#dev.copy2eps(file="z0702.eps",family='Japan1')

a_sa01<-extequ01$a[,1]-extequ01$a[,2];              #切片の差の事後分布
gqcal(a_sa01,1);                                    #p.96,1.6

##### 図 7.3 「データ 1」の切片の差の事後分布
hist(a_sa01,breaks=100,ylab="",xlab="切片の差",main="",cex.lab=2,cex.axis=1.5)
#dev.copy2eps(file="z0703.eps",family='Japan1')

##### 7.2.2 「データ 2」の分析
##### 表 7.3 「データ 2」の平均値
学習 02<-read.csv("./scrT/dat/学習 02.csv", header = TRUE)
round(tapply(学習 02$post, 学習 02$subject,mean),1)
round(tapply(学習 02$pre, 学習 02$subject,mean),1)

##### 図 7.4 「データ 2」の散布図
plot2g(y=学習 02$post,x=学習 02$pre,a=学習 02$subject,ba=3)
#dev.copy2eps(file="z0704.eps",family='Japan1')

par<-c("a","b","sigma");                          #母数
dataSetequ02<-list(n=nrow(学習 02),y=学習 02[,2],x=学習 02[,1],k=学習 02[,3]);#入力

#####cmdstanr による実行
### コンパイル結果は (傾きが共通のモデル)'modequ01' を利用する
csrfitequ02<-modequ01$sample(data=datasetequ02,chains=5,iter_sampling=20000,

```

```

  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitequ02 <- rstan::read_stan_csv(csrfitequ02$output_files()) #stan 形式への変換

#####rstan による実行
#fitequ02<-stan(model_code =ANCOVAequ,pars=par,
# data=list,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitequ02, file="./scrT/obje/stan_obje0702")
#load(file="./scrT/obje/stan_obje0702"); #予め stan( ) で作った object をロード

extequ02<-extract(fitequ02, par);                  #乱数の取り出し

##### 表 7.4 「データ 2」の傾きが共通したモデルの母数の事後分布の要約
gqcal(extequ02$a[,1],2)
gqcal(extequ02$a[,2],2)
gqcal(extequ02$b      ,2)
gqcal(extequ02$sigma,2)

##### 図 7.5 「データ 2」の回帰直線 (傾き共通)
plot2gequ(y=学習 02$post,x=学習 02$pre,a=学習 02$subject,ext=extequ02,ba=3)
#dev.copy2eps(file="z0705.eps",family='Japan1')

a_sa02<-extequ02$a[,1]-extequ02$a[,2];              #切片の差の事後分布
gqcal(a_sa02,1);                                    #p.97,b1.7

##### 図 7.6 「データ 2」の切片の差の事後分布
hist(a_sa02,breaks=100,ylab="",xlab="切片の差",
  main="",cex.lab=2,cex.axis=1.5)
#dev.copy2eps(file="z0706.eps",family='Japan1')

mean(abs(a_sa02)<1.0);                              #ROPE であるという phc
mean(abs(a_sa02)<1.5);                              #p.97,b1.4

##### 7. 3 傾きが異なる共分散分析

##### 7.3.1 「データ 3」の分析
##### 表 7.5 「データ 3」の平均値
学習 03<-read.csv("./scrT/dat/学習 03.csv", header = TRUE)
round(tapply(学習 03$post, 学習 01$subject,mean),1)
round(tapply(学習 03$pre , 学習 01$subject,mean),1)

##### 図 7.7 「データ 3」の散布図
plot2g(y=学習 03$post,x=学習 03$pre,a=学習 02$subject,ba=3);#図 7.7
#dev.copy2eps(file="z0707.eps",family='Japan1')

#####共分散分析 (傾きが異なるモデル) の stan コード
ANCOVAdef<- '
//共分散分析 (傾きが異なるモデル)
data {
  int<lower=0> n;          //データ数
  vector[n] y;            //基準変数
  vector[n] x;            //予測変数

```

```

int<lower=0,upper=2> k[n];          //分類変数
}
parameters {
  vector[2]  a;                    //切片
  vector[2]  b;                    //回帰係数
  real<lower=0> sigma;              //誤差標準偏差
}
transformed parameters {
  vector[n]  yhat;                 //予測値 yhat
  for(i in 1:n)
    yhat[i] = a[k[i]] + x[i]*b[k[i]]; //(7.7) 式,(7.8) 式
}
model {
  y ~ normal(yhat, sigma);          //(7.9) 式
}
';

par<-c("a","b","sigma")
dataSetdef03 <-list(n=nrow(学習 03),y=学習 03[,2],x=学習 03[,1],k=学習 03[,3])

#####cmdstanr による実行
modfiledef03 <- write_stan_file(ANCOVAdef)          #書き出す一時ファイル
moddef03 <- cmdstan_model(modfiledef03)             #コンパイル
csrfitdef03<-moddef03$sample(data=dataSetdef03,chains = 5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitdef03 <- rstan::read_stan_csv(csrfitdef03$output_files()) #stan 形式への変換

#####rstan による実行
#fitdef03<-stan(model_code =ANCOVAdef,pars=par,data=dataSetdef03,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitdef03, file="./scrT/obje/stan_obje0703")
#load(file="./scrT/obje/stan_obje0703"); #予め stan( ) で作った object をロード

extdef03<-extract(fitdef03, par);          #乱数の取り出し

##### 表 7.6 「データ 3」の傾きが異なるモデルによる母数の事後分布の要約
gqcal(extdef03$a[,1],2)
gqcal(extdef03$a[,2],2)
gqcal(extdef03$b[,1],2)
gqcal(extdef03$b[,2],2)
gqcal(extdef03$sigma,2)

##### 図 7.8 「データ 3」の回帰直線 (傾き異なる)
plot2gdef(y=学習 03$post,x=学習 03$pre,
  a=学習 03$subject,ext=extdef03,ba=3)
#dev.copy2eps(file="z0708.eps",family='Japan1')

##### 図 7.9 「データ 3」の予測値の差の点推定値と 95%予測区間
plot2gdefdef(x=学習 03[,1],extdef03);          #図 7.9
#dev.copy2eps(file="z0709.eps",family='Japan1')

#####7. 3. 2 「データ 1」と「データ 2」の再分析

```

```

#「データ 1」の傾きが異なるモデルでの再分析
#####cmdstanr による実行
### コンパイル結果は (傾きが異なるモデル)'moddef03' データは'dataSetequ01'
csrfitdef01<-moddef03$sample(data=dataSetequ01,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitdef01 <- rstan::read_stan_csv(csrfitdef01$output_files()) #stan 形式への変換

#####rstan による実行
#fitdef01<-stan(model_code =ANCOVAdef,pars=par,data=dataSetequ01,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitdef01, file="./scrT/obje/stan_obje0704")
#load(file="./scrT/obje/stan_obje0704"); #予め stan( ) で作った object をロード

extdef01<-extract(fitdef01, par);          #乱数の取り出し

#### 「データ 1」の傾きが異なるモデルによる母数の事後分布の要約 (教科書にない)
gqcal(extdef01$a[,1],2)
gqcal(extdef01$a[,2],2)
gqcal(extdef01$b[,1],2)
gqcal(extdef01$b[,2],2)
gqcal(extdef01$sigma,2)

##### 図 7.10 「データ 1」の回帰直線 (傾き異なる)
plot2gdef(y=学習 01$post,x=学習 01$pre,a=学習 01$subject,ext=extdef01)
#dev.copy2eps(file="z0710.eps",family='Japan1')

##### 図 7.11 「データ 1」の予測値の差の点推定値と 95%予測区間
plot2gdefdef(x=学習 01[,1],extdef01); abline(h=4.8)
#dev.copy2eps(file="z0711.eps",family='Japan1')

#「データ 2」の傾きが異なるモデルでの再分析
#####cmdstanr による実行
### コンパイル結果は (傾きが異なるモデル)'moddef03' データは'dataSetequ02'
csrfitdef02<-moddef03$sample(data=dataSetequ02,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitdef02 <- rstan::read_stan_csv(csrfitdef02$output_files()) #stan 形式への変換

#####rstan による実行
#fitdef02<-stan(model_code =ANCOVAdef,pars=par,data=dataSetequ02,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitdef02, file="./scrT/obje/stan_obje0705")
#load(file="./scrT/obje/stan_obje0705"); #予め stan( ) で作った object をロード

extdef02<-extract(fitdef02, par);          #乱数の取り出し

##### 図 7.12 「データ 2」の回帰直線 (傾き異なる)
plot2gdef(y=学習 02$post,x=学習 02$pre,a=学習 02$subject,ext=extdef02)
#dev.copy2eps(file="z0712.eps",family='Japan1')

```

```
##### 図 7.13 「データ 2」の予測値の差の点推定値と 95%予測区間
plot2gdefdef(x=学習 02[,1],extdef02)
#dev.copy2eps(file="z0713.eps",family='Japan1')
```

```
#####7. 3. 3 傾きの差の事後分布
##### 表 7.7 傾きの差の事後分布の要約
b_sa01<-extdef01$b[,1]-extdef01$b[,2]
gqcal(b_sa01)
b_sa02<-extdef02$b[,1]-extdef02$b[,2]
gqcal(b_sa02)
b_sa03<-extdef03$b[,1]-extdef03$b[,2]
gqcal(b_sa03)
```

```
##### 7.4 傾向スコア
##### 7.4.2 データの説明
##### 表 7.8 学習塾の既存のクラスを利用した新教授法の効果の検証実験データ
学習 04<-read.csv("./scrT/dat/propen.csv", header = TRUE)
head(学習 04)
tail(学習 04)
```

```
##### 変数ごとの度数 (群 p.103, b1.4 以外は、教科書にはない)
table(学習 04$群)
table(学習 04$高校)
table(学習 04$収入)
table(学習 04$母教育)
table(学習 04$父教育)
```

```
##### 表 7.9 実験群と対照群の平均値
round(tapply(学習 04$成績, 学習 04$群,mean),1)
round(tapply(学習 04$高校, 学習 04$群,mean),1)
round(tapply(学習 04$収入, 学習 04$群,mean),1)
round(tapply(学習 04$母教育, 学習 04$群,mean),1)
round(tapply(学習 04$父教育, 学習 04$群,mean),1)
```

```
#####傾向スコアによる分析の stan コード
propensity01<-'
data {
  int<lower=0> n; //データ数
  int<lower=0> p; //予測変数の
  vector[n] y; //基準変数
  matrix[n,p] x; //予測変数
  int<lower=1,upper=2> k[n]; //分類変数 1 実験群 2 対照群
}
transformed data {
  int<lower=0,upper=1> kk[n]; //分類変数
  for(i in 1:n){
    kk[i]=k[i]*(-1)+2;
  }
}
```

```
parameters {
  vector[2] a; //切片
  real b; //回帰係数
  real alpha; //ロジスティック回帰切片
  vector[p] beta; //ロジスティック回帰係数
  real<lower=0> sigma; //誤差標準偏差
}
transformed parameters {
  vector[n] yhat; //yhat
  vector<lower=0,upper=1>[n] propensity; //傾向スコア
  for(i in 1:n){
    propensity[i] = inv_logit(alpha + x[i,]*beta);
    yhat[i] = a[k[i]] + propensity[i]*b;
  }
}
model {
  for(i in 1:n){
    kk[i] ~ bernoulli(propensity[i]); //ベルヌイ分布モデル
    y[i] ~ normal(yhat[i], sigma); //正規分布モデル
  }
}
generated quantities{
}
';
```

```
par<-c("a","b","sigma","alpha","beta","propensity") #母数
dataSetpropen01 <-list(n=nrow(学習 04),p=4,y=学習 04[,2],
  x=学習 04[,c(3:6)],k=学習 04[,1]) #入力
```

```
#####cmdstanr による実行
modfilepropen01 <- write_stan_file(propensity01) #書き出す一時ファイル
modpropen01 <- cmdstan_model(modfilepropen01) #コンパイル
csrfitpropen01 <- modpropen01$sample(data = dataSetpropen01,chains = 5,
  iter_sampling=20000,iter_warmup=1000,parallel_chains= 5,seed=1234) #MCMC
fitpropen01 <- rstan::read_stan_csv(csrfitpropen01$output_files()) #stan へ変換
```

```
#####rstan による実行
#fitpropen01<-stan(model_code =propensity01,pars=par,data=dataSetpropen01,
# seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitpropen01, file="./scrT/obje/stan_obje0706")
#load(file="./scrT/obje/stan_obje0706"); #予め stan( ) で作った object をロード
```

```
extpropen01<-extract(fitpropen01, par); #乱数の取り出し
```

```
##### 表 7.10 傾向スコアを共変量とした共分散分析の母数の事後分布の要約
gqcal(extpropen01$a[,1],2)
gqcal(extpropen01$a[,2],2)
gqcal(extpropen01$b[,2])
gqcal(extpropen01$sigma,2)
gqcal(extpropen01$alpha,2)
gqcal(extpropen01$beta[,1],2)
gqcal(extpropen01$beta[,2],2)
```

```
gqcal(extpropen01$beta[,3],2)
gqcal(extpropen01$beta[,4],2)
gqcal(extpropen01$propensity[,1],2)
gqcal(extpropen01$propensity[,2],2)
gqcal(extpropen01$propensity[,3],2)
gqcal(extpropen01$propensity[,4],2)
```

```
##### 図 7.14 群別の傾向スコアのヒストグラム
propen01<-apply(extpropen01$propensity,2,mean);      #傾向スコアの EAP
階級<-seq(0,1,0.1)
barplot(table(学習 04$群,cut(propen01, 階級,include.lowest = T,right = F)),
  xlab="propensity score", axisnames = F,cex.lab=2.0,cex.axis=1.5)
#dev.copy2eps(file="z0714.eps",family='Japan1')
```

```
##### 図 7.15 ROC 曲線
(ROC1<-roc(学習 04$群 ~ propen01,ci=T))
plot(ROC1,cex.lab=2.0,cex.axis=1.5)
#dev.copy2eps(file="z0715.eps",family='Japan1')
```

```
##### 図 7.16 傾向スコアを共変量とした共分散分析
plot2gequ(y=学習 04$成績,x=propen01,a=学習 04$群,
  ext=extpropen01,ba=0,xlab="propensity score")
#dev.copy2eps(file="z0716.eps",family='Japan1')
```

```
##### 図 7.17 切片の差の事後分布
a_sa04<-extpropen01$a[,1]-extpropen01$a[,2]
gqcal(a_sa04,1);
hist(a_sa04,br=100,cex.lab=2.0,cex.axis=1.5,xlab="a1 - a2",ylab="",main="")
#dev.copy2eps(file="z0717.eps",family='Japan1')
```

```
##### 「切片の差が ROPE(1.0) である」の phc p,108,1.3
mean(abs(a_sa04)<1)
```

```
#以下実習はしない
set.seed(1234)
match<-Match(Tr= 学習 04$群*-1+2, X=propen01, M=1, caliper=0.2, replace=F)
match
```

## 7.2 本章で利用する関数

### 7.2.1 plot2g: 2 群の 2 変量データの散布図を pch を変えて描く

```
plot2g(y,x,a,ba=5,xlab="pre test")
```

## 引数

y 基準変数

x 予測変数 (補助変数)

a 分類変数、内容は 1 または 2

ba 最大値・最小値の外側にどれだけ縦軸・横軸を伸ばすか

### 7.2.2 plot2gequ: 2 群の散布図と 2 本の傾き共通の回帰直線を描く

```
plot2gequ(y,x,a,ext,ba=5,xlab="pre test")
```

## 引数

y 基準変数

x 予測変数 (補助変数)

a 分類変数、内容は 1 または 2

ext extract( ) によって取り出した乱数のオブジェクト

ba 最大値・最小値の外側にどれだけ縦軸・横軸を伸ばすか

### 7.2.3 plot2gdef: 散布図に傾きの異なった 2 本の回帰直線を描く

```
plot2gdef(y,x,a,ext,ba=3)
```

## 引数

**y** 基準変数

**x** 予測変数（補助変数）

**a** 分類変数、内容は 1 または 2

**ext** `extract( )` によって取り出した乱数のオブジェクト

**ba** 最大値・最小値の外側にどれだけ縦軸・横軸を伸ばすか

#### 7.2.4 `plot2gdef`: 傾きの異なった 2 本の回帰直線の予測値の差の EAP と 95 % 確信区間を描く

```
plot2gdefdef(x,ext)
```

### 引数

**x** 予測変数（補助変数）

**ext** `extract( )` によって取り出した乱数のオブジェクト

### 7.3 自習問題

1. 図 7.12 によって、「データ 2」の傾きが異なるモデルによる 2 本の回帰直線 はほぼ同じであることが目視で分かった。念のために、表 7.6 に相当する事後分布の要約を示しなさい。
2. 7.3.3 傾きの差の事後分布における表 7.7 において、傾きの差の事後分布の要約が示された。分布と 0 との位置関係を目視で確認するために、それら 3 つのヒストグラムを描きなさい。

### 7.4 第 7 章 宿題

教科書 p.110, 7.8 節 実習課題 をしなさい。



## 8 さらに進んだ実験計画

### 8.1 スクリプト

```
#####第8章 さらに進んだ実験計画
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))            #wd 戻す
library(rstan)           #パッケージ rstan の呼び出し
library(cmdstanr)        #パッケージ cmdstanr の呼び出し
library(posterior)       #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

#####8.1 対応のある 1 要因の推測

##### 8.1.1 触 2 点閾の感覚実験

##### 表 8.1 触 2 点閾の感覚実験データ
(触2点閾<-read.csv("./scrT/dat/触2点閾.csv", header = TRUE))

##### 図 8.1 触 2 点閾の測定値 (部位ごと mm)
boxplot(触2点閾$長さ~触2点閾$部位,xlab="部位",ylab="",cex.axis=2,cex.lab=2)
#dev.copy2eps(file="z0601.eps",family='Japan1')

##### 部位ごとの触 2 点閾の測定値の平均・sd p.111,b1.2
tapply(触2点閾$長さ, 触2点閾$部位,mean)
tapply(触2点閾$長さ, 触2点閾$部位,sd)

##### 図 8.2 触 2 点閾の測定値 (被験者ごと mm)
boxplot(触2点閾$長さ~触2点閾$被験者,xlab="被験者",cex.axis=1.4,cex.lab=1.5)
#dev.copy2eps(file="z0602.eps",family='Japan1')

##### 被験者ごとの触 2 点閾の測定値の平均・sd p.112, 下部、具体的な数値はなし
tapply(触2点閾$長さ, 触2点閾$被験者,mean)
tapply(触2点閾$長さ, 触2点閾$被験者,sd)

#####対応のある 1 要因の推測の stan コード
RBlockD<- '
data {
  int<lower=0> n;          //全データ数
  int<lower=0> J;          //群数
  int<lower=0> K;          //ブロック数
  vector[n] y;            //基準変数
  int<lower=0> j[n];       //分類変数 A
  int<lower=0> k[n];       //ブロック変数 B
}
parameters {
  vector[J] mu;           //各群の効果
  vector[K] s;            //ブロックの効果
```

```
real<lower=0> s_e;        //誤差 SD
real<lower=0> s_s;        //ブロック SD
}
model {
  y ~ normal(mu[j]+s[k], s_e); // (8.1) 式, (8.2) 式
  s ~ normal(0, s_s);         // (8.3) 式
}
generated quantities{
  real tmu;                  //全平均
  real s_mu;                 //要因 SD
  tmu = mean(mu);            //μ j の平均
  s_mu =sqrt(variance(mu)*(J-1)/J); //μ j の s d
}
';
par<-c("mu","s_e","s_s","tmu","s_mu","s"); #母数
dataSetran <-list(n=length(触2点閾$長さ),J=max(触2点閾$部位), #入力
K=max(触2点閾$被験者),y=触2点閾$長さ,j=触2点閾$部位,k=触2点閾$被験者)

#####cmdstanr による実行
modfileran <- write_stan_file(RBlockD) #書き出す一時ファイル
modran <- cmdstan_model(modfileran)    #コンパイル
csrfitrان <- modran$sample(data = dataSetran,chains = 5,iter_sampling = 20000,
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitran <- rstan::read_stan_csv(csrfitrان$output_files()) #stan 形式への変換

#####rstan による実行
#fitran<-stan(model_code =RBlockD,data=dataSetran,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitran, file="./scrT/obje/stan_obje0801")
#load(file="./scrT/obje/stan_obje0801"); #予め stan( ) で作った object をロード

extran<-extract(fitran, par)

##### 表 8.2 乱塊計画の母数の事後分布
eta2 = (extran$s_mu^2)/((extran$s_mu^2)+(extran$s_e^2));#説明率 (8.5) 式
gqcal(extran$mu[,1])
gqcal(extran$mu[,2])
gqcal(extran$mu[,3])
print('-----')
gqcal(extran$s_e )
gqcal(extran$s_s )
print('-----')
gqcal(extran$tmu )
gqcal(extran$s_mu )
print('-----')
gqcal(extran$s[,4] )
print('-----')
gqcal(eta2)

##### 群間に差がある確率 (意味があるための必要条件の確認) p.114,b1.2
mean(extran$mu[,1]>extran$mu[,2])
mean(extran$mu[,2]>extran$mu[,3])
```

```

mean(extran$mu[,1]>extran$mu[,3])

##### 図 8.3 平均値の差に関する phc 曲線
par(mfrow=c(2,1)); #図 8.3
phc01(seq01=seq(0,1.2,0.05),a=extran$mu[,1],b=extran$mu[,2],
      xlab="c<math>\mu</math> 1-

```

```

s ~ normal(0, s_s); // (8.8) 式
}
';

par<-c("mu","s_e","s_s") #母数
dataSetnest <-list(n=length(潜在学習$time), J=max(潜在学習$条件), #入力
                  K=max(潜在学習$ネズミ),y=潜在学習$time,j=潜在学習$条件,k=潜在学習$ネズミ)

##### cmdstanr による実行
modfilenest <- write_stan_file(nestedD) #書き出す一時ファイル
modnest <- cmdstan_model(modfilenest) #コンパイル
csrfitnest <- modnest$sample(data=DataSetnest,chains=5,iter_sampling = 20000,
                             iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitnest <- rstan::read_stan_csv(csrfitnest$output_files()) #stan 形式への変換

##### rstan による実行
#fitnest<-stan(model_code =nestedD,data=DataSetnest,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitnest, file="./scrT/obje/stan_obje0802")
#load(file="./scrT/obje/stan_obje0802"); #予め stan( ) で作った object をロード

extnest<-extract(fitnest, par); #乱数の取り出し

##### 表 8.4 枝分かれ計画の母数と生成量の事後分布の要約 (前半)
gqcal(extnest$mu[,1])
gqcal(extnest$mu[,2])
gqcal(extnest$mu[,3])
gqcal(extnest$s_e )
gqcal(extnest$s_s )

##### 表 8.4 (後半) 平均値の差の事後分布
def32 = extnest$mu[,3]-extnest$mu[,2]; # (8.10) 式
def13 = extnest$mu[,1]-extnest$mu[,3]; # (8.11) 式
gqcal(def32);
gqcal(def13); #表 8.4 ここまで

##### 8. 3 対応ある 2 要因の推測
##### 8. 3. 1 説得的メッセージの比較調査

##### 表 8.5 説得的メッセージの比較調査データ
(便所標語<-read.csv("./scrT/dat/便所標語.csv", header = TRUE))

##### 図 8.6 セルごとの箱ひげ図
boxplot(便所標語$評定~便所標語$標語+便所標語$katu,xlab="標語/活字",ylab="評定")
#dev.copy2eps(file="z0606.eps",family='Japan1')

##### 図 8.7 被験者ごとの箱ひげ図
boxplot(便所標語$評定~便所標語$被験者,xlab="被験者", ylab="評定")
#dev.copy2eps(file="z0607.eps",family='Japan1')

```

```
#####対応ある 2 要因の推測の stan コード
RBFDF<-
//乱塊要因計画・被験者内 2 要因 実習 1 2
functions{
  vector zero_sum_vector(int a, vector m1A){//主効果ゼロ和
    vector[a] muA;
    for(i in 1:(a-1)){ muA[i] = m1A[i];} //a-1 まではソックリ代入
    muA[a] = -sum(m1A); //a 番目に、和の-1 倍代入
    return(muA);
  }
  matrix zero_sum_matrix(int a, int b, matrix m1AB){//交互作用ゼロ和
    vector [a-1] m1a; //行和を入れるベクトル
    vector [b-1] m1b; //列和を入れるベクトル
    matrix [a,b] muAB; //リターンする行列
    for(i in 1:(a-1)){ m1a[i] = 0.0;} //ゼロクリア
    for(j in 1:(b-1)){ m1b[j] = 0.0;} //ゼロクリア
    for(i in 1:(a-1)){ for(j in 1:(b-1)){muAB[i,j] =m1AB[i,j];}}//ソックリ代入
    for(i in 1:(a-1)){ for(j in 1:(b-1)){m1a[i] =m1a[i]+m1AB[i,j];}}//行和作る
    for(j in 1:(b-1)){ for(i in 1:(a-1)){m1b[j] =m1b[j]+m1AB[i,j];}}//列和作る
    for(i in 1:(a-1)){muAB[i,b] = (-1)*m1a[i]; } //第 b 列に行和の-1 倍代入
    for(j in 1:(b-1)){muAB[a,j] = (-1)*m1b[j]; } //第 a 行に列和の-1 倍代入
    muAB[a,b] = sum(m1a); //第 a 行第 b 列要素に合計代入、注意：マイナスではない
    return(muAB);
  }
}
data {
  int<lower=0> n; //全データ数
  int<lower=0> a; //A 水準数
  int<lower=0> b; //B 水準数
  int<lower=0> s; //ブロック数
  vector[n] y; //基準変数
  int<lower=0> A[n]; //分類変数 A
  int<lower=0> B[n]; //分類変数 B
  int<lower=0> S[n]; //ブロック変数 s
}
parameters {
  real mu; //全平均
  vector [a-1] m1A; //A 平均 1 つ少ない
  vector [b-1] m1B; //B 平均 1 つ少ない
  vector [s-1] m1S; //S 平均 1 つ少ない
  matrix [a-1,b-1] m1AB; //交互作用 1 つ少ない
  matrix [a-1,s-1] m1AS; //交互作用 AS1 つ少ない
  matrix [b-1,s-1] m1BS; //交互作用 BS1 つ少ない
  real<lower=0> s_E; //E 標準偏差
  real<lower=0> s_S; //S 標準偏差
  real<lower=0> s_AS; //AS 標準偏差
  real<lower=0> s_BS; //BS 標準偏差
}
transformed parameters {
  vector [a] muA; //A 平均
  vector [b] muB; //B 平均
  matrix [a,b] muAB; //交互作用
```

```
muA =zero_sum_vector(a, m1A); //p.121(8.12) 式 2 行下、1 つ目
muB =zero_sum_vector(b, m1B); //2 つ目
muAB=zero_sum_matrix(a, b, m1AB); //3, 4 つ目
}
model {
  vector [s] muS; //S 平均
  matrix [a,s] muAS; //交互作用 AS
  matrix [b,s] muBS; //交互作用 BS
  m1S ~ normal(0,s_S); //p.121(8.12) 式 1 行下、2 つ目
  muS =zero_sum_vector(s, m1S); //p.125(8.13) 式 2 行下、1 つ目
  for(j in 1:a-1){ for(l in 1:s-1){m1AS[j,l] ~ normal(0,s_AS);}}// 3 つ目
  for(k in 1:b-1){ for(l in 1:s-1){m1BS[k,l] ~ normal(0,s_BS);}}// 4 つ目
  muAS=zero_sum_matrix(a, s, m1AS); //as の和を 0 にする
  muBS=zero_sum_matrix(b, s, m1BS); //bs の和を 0 にする
  for(i in 1:n){
    y[i] ~ normal(mu+muA[A[i]]+muB[B[i]]+muS[S[i]]+
      muAB[A[i],B[i]]+muAS[A[i],S[i]]+muBS[B[i],S[i]],s_E);} //(8.12) 式
  }
  generated quantities{
    real s_a; //要因 A の SD
    real s_b; //要因 B の SD
    real s_ab; //交互作用 AB の SD
    s_a =sqrt(variance(muA)*(a-1)/a); //要因 A の sd
    s_b =sqrt(variance(muB)*(b-1)/b); //要因 B の sd
    s_ab =sqrt(variance(muAB)*((a*b)-1)/(a*b)); //交互作用 AB の sd
  }
};

par<-c("mu", "muA", "muB", "muAB", "s_S", "s_AS",
  "s_BS", "s_E", "s_a", "s_b", "s_ab") #母数
dataSetRBFDF <-list(n=length(便所標語$評定), a=max(便所標語$標語),
  b=max(便所標語$活字), s=max(便所標語$被験者), y=便所標語$評定,
  A=便所標語$標語, B=便所標語$活字, S=便所標語$被験者) #入力

#####cmdstanr による実行
modfileRBFDF <- write_stan_file(RBFDF) #書き出す一時ファイル
modRBFDF <- cmdstan_model(modfileRBFDF) #コンパイル
csrfitRBFDF <- modRBFDF$sample(data=dataSetRBFDF,chains=5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitRBFDF <- rstan::read_stan_csv(csrfitRBFDF$output_files()) #stan 形式への変換

#####rstan による実行
#fitRBFDF<-stan(model_code =RBFDF,data=dataSetRBFDF,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitRBFDF, file="./scrT/obje/stan_obje0803")
#load(file="./scrT/obje/stan_obje0803"); #予め stan( ) で作った object をロード

extRBFDF<-extract(fitRBFDF, par); #乱数の取り出し

##### 表 8.6 乱塊要因計画の母数の事後分布
gqcal(extRBFDF$mu )
gqcal(extRBFDF$muA[,1] )
```

```

gqcal(extRBFd$muA[,2] )
gqcal(extRBFd$muA[,3] )
gqcal(extRBFd$muB[,1] )
gqcal(extRBFd$muAB[,1,1])
gqcal(extRBFd$muAB[,2,1])
gqcal(extRBFd$muAB[,3,1])
gqcal(extRBFd$s_S )
gqcal(extRBFd$s_AS )
gqcal(extRBFd$s_BS )
gqcal(extRBFd$s_E )
print('-----')
gqcal(extRBFd$s_a )
gqcal(extRBFd$s_b )
gqcal(extRBFd$s_ab )

##### 8. 3. 4 PHC 「種類」の水準間に差がある確率
phc02(0,extRBFd$muA)

# 「フロント」の水準間に差がある確率
mean(extRBFd$muB[,2]-extRBFd$muB[,1]>0)

# 連言命題が正しい確率 「罰」 < 「直截」 かつ 「直截」 < 「公共心」
round(mean((extRBFd$muA[,1]<extRBFd$muA[,2])&(extRBFd$muA[,2]<extRBFd$muA[,3])),3)

##### 図 8.8 水準の効果に c 以上の差のある確率の曲線
par(mfrow=c(3,1))
phc01(seq01=seq(0.5,1.4,0.05),a=extRBFd$muA[,2],b=extRBFd$muA[,1],
      xlab="c<a2-a1",cex.lab=2.5)
phc01(seq01=seq(1,2,0.05),a=extRBFd$muA[,3],b=extRBFd$muA[,2],
      xlab="c<a3-a2",cex.lab=2.5)
phc01(seq01=seq(1,2,0.05),a=extRBFd$muB[,2],b=extRBFd$muB[,1],
      xlab="c<b2-b1",cex.lab=2.5)
par(mfrow=c(1,1))
#dev.copy2eps(file="z0608.eps",family='Japan1')

##### 実質的に差があると明言できる確率 p.123,1.3 以降このページの最後まで
phc01(seq01=0.8,a=extRBFd$muA[,2],b=extRBFd$muA[,1],byoga="no")
phc01(seq01=1.3,a=extRBFd$muA[,3],b=extRBFd$muA[,2],byoga="no")
phc01(seq01=1.5,a=extRBFd$muB[,2],b=extRBFd$muB[,1],byoga="no")

##### 8. 4 混合した 2 要因の推測
##### 8. 4. 1 健康診断の追跡調査

##### 表 8.7 健康診断の追跡調査
(うつ追跡<-read.csv("./scrT/dat/うつ追跡.csv", header = TRUE))

##### 図 8.9 うつ病検査の追跡調査 乱数を加え散布図を見やすくしている
ma<-57;mi<-48
a1<-うつ追跡 [ 1: 50,1]+runif(50)
a2<-うつ追跡 [ 51:100,1]+runif(50)

```

```

b1<-うつ追跡 [101:150,1]+runif(50)
b2<-うつ追跡 [151:200,1]+runif(50)
plot(a1,a2,pch=21,xlim=c(mi,ma),ylim=c(mi,ma),xlab="pre.test",ylab="post.test")
abline(0,1)
par(new=T)
plot(b1,b2,pch=19,xlim=c(mi,ma),ylim=c(mi,ma),xlab="",ylab="")
legend(49,56,c("対照群","介入群"),pch=c(21,19)); #図 8.9 ここまで

##### 混合した 2 要因の推測の stan コード
SPD<-'
functions{
  vector zero_sum_vector(int a, vector m1A){//主効果ゼロ和
    vector[a] muA;
    for(i in 1:(a-1)){ muA[i] = m1A[i];} //a-1 まではソックリ代入
    muA[a] = -sum(m1A); //a 番目に、和の-1 倍代入
    return(muA);
  }
  matrix zero_sum_matrix(int a, int b, matrix m1AB){//交互作用ゼロ和
    vector [a-1] m1a; //行和を入れるベクトル
    vector [b-1] m1b; //列和を入れるベクトル
    matrix [a,b] muAB; //リターンする行列
    for(i in 1:(a-1)){ m1a[i] = 0.0;} //ゼロクリア
    for(j in 1:(b-1)){ m1b[j] = 0.0;} //ゼロクリア
    for(i in 1:(a-1)){ for(j in 1:(b-1)){muAB[i,j] =m1AB[i,j];}}//ソックリ代入
    for(i in 1:(a-1)){ for(j in 1:(b-1)){m1a[i] =m1a[i]+m1AB[i,j];}}//行和作る
    for(j in 1:(b-1)){ for(i in 1:(a-1)){m1b[j] =m1b[j]+m1AB[i,j];}}//列和作る
    for(i in 1:(a-1)){muAB[i,b] = (-1)*m1a[i]; } //第 b 列に行和の-1 倍代入
    for(j in 1:(b-1)){muAB[a,j] = (-1)*m1b[j]; } //第 a 行に列和の-1 倍代入
    muAB[a,b] = sum(m1a); //第 a 行第 b 列要素に合計代入、注意：マイナスではない
    return(muAB);
  }
}
data {
  int<lower=0> n; //全データ数
  int<lower=0> a; //A 水準数 (標本間要因)
  int<lower=0> b; //B 水準数 (標本内要因)
  int<lower=0> s; //ブロック数
  vector[n] y; //基準変数
  int<lower=0> A[n]; //分類変数 A (標本間要因)
  int<lower=0> B[n]; //分類変数 B (標本内要因)
  int<lower=0> S[n]; //ブロック変数 s
}
parameters {
  real mu; //全平均
  vector [a-1] m1A; //A 平均 1 つ少ない
  vector [b-1] m1B; //B 平均 1 つ少ない
  vector [s*a-1] m1S; //S 平均 1 つ少ない
  matrix [a-1,b-1] m1AB; //交互作用 1 つ少ない
  real<lower=0> s_E; //E 標準偏差
  real<lower=0> s_S; //S 標準偏差
}

```

```

transformed parameters {
  vector      [a] muA;          //A 平均
  vector      [b] muB;          //B 平均
  matrix      [a,b] muAB;       //交互作用
  muA =zero_sum_vector(a, m1A); //p.125(8.13) 式 2 行下、1 つ目
  muB =zero_sum_vector(b, m1B); //2 つ目
  muAB=zero_sum_matrix(a, b, m1AB); //3, 4 つ目
}
model {
  vector      [s*a] muS;        //S 平均
  m1S ~ normal(0,s_S);          //p.125(8.13) 式 1 行下、2 つ目
  muS =zero_sum_vector(s*a, m1S); //p.125(8.13) 式 2 行下、1 つ目
  for(i in 1:n){
    y[i] ~ normal(mu+muA[A[i]]+muB[B[i]]+muS[S[i]]+
      muAB[A[i],B[i]],s_E);}    //(8.13) 式
}
generated quantities{
  real s_a;                      //要因 A の SD
  real s_b;                      //要因 B の SD
  real s_ab;                     //交互作用 AB の SD
  s_a =sqrt(variance(muA)*(a-1)/a); //要因 A の sd
  s_b =sqrt(variance(muB)*(b-1)/b); //要因 B の sd
  s_ab =sqrt(variance(muAB)*((a*b)-1)/(a*b)); //交互作用 AB の sd
}
';
par<-c("mu","muA","muB","muAB","s_S","s_E","s_a","s_b","s_ab") #母数
dataSetSPD <-list(n=length(うつ追跡$score),a=max(うつ追跡$群),
  b=max(うつ追跡$時期),s=max(うつ追跡$学生),y=うつ追跡$score,
  A=うつ追跡$群, B=うつ追跡$時期, S=うつ追跡$学生) #入力

#####cmdstanr による実行
modfileSPD <- write_stan_file(SPD) #書き出す一時ファイル
modSPD <- cmdstan_model(modfileSPD) #コンパイル
csrfitSPD <- modSPD$sample(data = dataSetSPD,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitSPD <- rstan::read_stan_csv(csrfitSPD$output_files()) #stan 形式への変換

#####rstan による実行
#fitSPD<-stan(model_code =SPD,data=dataSetSPD,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fitSPD, file="./scrT/obje/stan_obje0804")
#load(file="./scrT/obje/stan_obje0804"); #予め stan( ) で作った object をロード

extSPD<-extract(fitSPD, par); #乱数の取り出し

#####表 8.8 混合計画の母数の事後分布の要約
gqcal(extSPD$mu )
gqcal(extSPD$muA[,1] )
gqcal(extSPD$muB[,1] )
gqcal(extSPD$muAB[,1,1])
gqcal(extSPD$muAB[,1,2])
gqcal(extSPD$muAB[,2,1])

```

```

gqcal(extSPD$muAB[,2,2])
gqcal(extSPD$s_S )
gqcal(extSPD$s_E )
print('-----')
gqcal(extSPD$s_a )
gqcal(extSPD$s_b )
gqcal(extSPD$s_ab )

```

#####生成量 (j 番目の群の k 番目の時期のセル jk の効果) の事後分布 (8.15) 式

```

ce11<-extSPD$muA[,1]+extSPD$muB[,1]+extSPD$muAB[,1,1]
ce12<-extSPD$muA[,1]+extSPD$muB[,2]+extSPD$muAB[,1,2]
ce21<-extSPD$muA[,2]+extSPD$muB[,1]+extSPD$muAB[,2,1]
ce22<-extSPD$muA[,2]+extSPD$muB[,2]+extSPD$muAB[,2,2]

```

##### 表 8.9 群内の時期差の事後分布

```

gqcal(ce11-ce12); # (8.16) 式
gqcal(ce21-ce22);

```

##### 群内の時期差の事後分布のヒストグラム (教科書にはない)

```

hist(ce11-ce12,main="",ylab="")
hist(ce21-ce22,main="",ylab="")

```

##### 実習課題

```

#1
#def32
phc01(seq01=seq(-3,6,0.05),a=def32,cc="gtc",byoga="yes",xlab="d32",cex.lab=2.5)
phc01(seq01=seq(-3,6,0.5 ),a=def32,cc="gtc",byoga="no")

```

```

#def32,ROPE
phc01(seq01=seq(0,6,0.05),a=def32,cc="rope",byoga="yes",xlab="|d32|<c",
  cex.lab=2.5)
phc01(seq01=seq(0,6,0.5 ),a=def32,cc="rope",byoga="no")

```

```

#def13
phc01(seq01=seq(17,25,0.05),a=def13,cc="gtc",byoga="yes",xlab="d32",
  cex.lab=2.5)
phc01(seq01=seq(17,25,0.5 ),a=def13,cc="gtc",byoga="no")

```

#2

```

#
# 正解は割愛
#

```

## 8.2 自習問題

- 8.2 節、ネストした実験、p.118, (8.10) 式,(8.11) 式の群間差のヒストグラムを描きなさい。可能であれば、共通の横軸に描いた平均値の差の 2 つの事

後分布のヒストグラムを描きなさい。

2. 8.4 節、混合実験、p.126, 表 8.9 と (8.16) 式に登場するセル  $jk$  の効果に関して、 $\text{phc}(c < ce_{11} - ce_{12})$  と ROPE の  $\text{phc}(|ce_{11} - ce_{12}| < c)$  と  $\text{phc}(c < ce_{21} - ce_{22})$  の曲線とテーブルを作成し、差を考察しなさい。

### 8.3 第 8 章 宿題

「8. 3 節 対応ある 2 要因の推測」は自習してください。

「8. 2 節 ネストした 1 要因の推測」p.115 の分析を行う。 ”鏡映描写 3 群 time.R” は、実際の鏡映描写実験の time である。データは次のように読み込む  
`source("scrT/dat/鏡映描写 3 群 time.R")`

「利」が「利き腕群」、「非」が「非利き腕群」、「休」が「休息群」のデータ行列である。3 列あり、変数 13, 14, 15 は何番目の試行かを意味する。行は被験者である。このデータには、固定効果の実験要因 (3 水準) と変量効果のブロック要因 (被験者 107 水準) がある。「利き手群」の 1 – 36 人目の被験者と「非利き手群」の 37 – 72 人目の被験者と「休憩群」の 73 – 107 人目の被験者は、それぞれに異なった学生である。

ゆえにブロック要因は実験要因にネストしている。(8.6) 式の添字 ( $j=1, \dots, 3$ ) は 3 つの実験水準、( $k=1, \dots, 107$ ) は 107 人の被験者、( $i=1, \dots, 3$ ) は 3 回の試行を意味する。

十分学習した後なので、実力は安定したと考え、3 回の試行は単なる繰り返しとして扱う。しかし安定した後も被験者間の実力差はあるものとみなし、ブロック要因とした。以下の手順で分析せよ。

1. 分析できるように “潜在学習.csv” の形式に成形せよ。  
表 8.4 の  $\mu$  に相当する表を作成し解釈せよ。
2. 表 8.4 の d に相当する表を作成し解釈せよ。
3.  $\text{phc}(c < \mu_{\text{休}} - \mu_{\text{非}})$  のテーブルを作り、仮説 A を考察せよ。
4.  $\text{phc}(c < \mu_{\text{非}} - \mu_{\text{利}})$  のテーブルを作り、仮説 B を考察せよ。
5. 仮説 C に相当する連言命題が成立する確率を求めよ。  
 $c_1=(1, 2, 3)$  秒,  $c_2=(1, 2, 3)$  秒の組み合わせで  $3 \times 3$  のテーブルを描き、解釈せよ。

## 9 階層線形モデル

### 9.1 スクリプト

```
##### 第9章 階層線形モデル
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))        #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))            #wd 戻す
library(rstan)           #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義
```

##### 9.1 切片と回帰係数に分布を仮定したモデル

##### 9.1.1 知覚された長さに関する実験

##### 表 9.1 知覚された長さに関する実験  
(知覚長 11<-read.csv("./scrT/dat/知覚長 11.csv", header = TRUE))

##### 図 9.1 知覚された長さの実験 (2 名分)

```
X<-知覚長 11[!(知覚長 11$学生==17)|(知覚長 11$学生==1)], ]
plot(X$実測,X$目測,pch=X$学生,xlab="実測",ylab="目測")
  legend(60,160,c("学生 1","学生 17"),pch=c(1,17),cex=1.5)
#dev.copy2eps(file="z0806.eps",family='Japan1')
```

##### 図 9.2 知覚された長さの実験 (25 名分)

```
plot(知覚長 11$実測, 知覚長 11$目測,pch=知覚長 11$学生, #図 9.2
     xlab="実測",ylab="目測")
```

##### 切片と回帰係数に分布を仮定したモデルの stan コード  
HLM01<-'

```
data {
  int<lower=0> n;          //全データ数
  int<lower=0> J;          //群数
  vector[n] y;            //基準変数
  vector[n] x;            //予測変数
  int<lower=0> k[n];       //分類変数
}
parameters {
  vector[J] a;            //切片
  vector[J] b;            //回帰係数
  real<lower=0> sigma;     //誤差標準偏差
  real a0;               //切片平均
  real b0;               //回帰係数平均
  real<lower=0> s_a;      //切片 SD
  real<lower=0> s_b;      //回帰係数 SD
}
transformed parameters {
  vector[n] yhat;         //予測値
```

```
for(i in 1:n)
  yhat[i] = a[k[i]] + x[i]*b[k[i]]; //(9.1 式)
}
model {
  y ~ normal(yhat, sigma);          //(9.2 式)
  a ~ normal(a0, s_a);              //(9.3 式)
  b ~ normal(b0, s_b);              //(9.4 式)
}
';
```

```
par<-c("sigma","a0","b0","s_a","s_b","a","b") #母数
dataSet81 <-list(n=length(知覚長 11$目測), J=max(知覚長 11$学生),
  y=知覚長 11$目測, x=知覚長 11$実測, k=知覚長 11$学生) #入力データ
```

```
#####cmdstanr による実行
modfile81 <- write_stan_file(HLM01)          #書き出す一時ファイル
mod81 <- cmdstan_model(modfile81)            #コンパイル
csrfit81 <- mod81$sample(data = dataSet81,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit81 <- rstan::read_stan_csv(csrfit81$output_files()) #stan 形式への変換
```

```
#####rstan による実行
#fit81<-stan(model_code =HLM01,data=dataSet81,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit81, file="./scrT/obje/stan_obje0901")
#load(file="./scrT/obje/stan_obje0901"); #予め stan( ) で作った object をロード
```

```
ext81<-extract(fit81, par);                #乱数の取り出し
```

##### 表 9.2 切片と回帰係数が分布をもつモデルの母数の事後分布の要約

```
gqcal(ext81$sigma)
gqcal(ext81$a0 )
gqcal(ext81$b0 )
gqcal(ext81$s_a )
gqcal(ext81$s_b )
```

```
##### 25 本の切片・回帰係数の EAP 推定値 (作図のためであり、教科書にはない)
av<-apply(ext81$a,2,mean)
bv<-apply(ext81$b,2,mean)
round(rbind(av,bv),2)
```

##### 図 9.3 25 本の単回帰直線

```
plot(知覚長 11$実測, 知覚長 11$目測,pch=知覚長 11$学生,
     xlab="実測",ylab="目測",type="n")
  for (i in 1:25) {abline(av[i],bv[i],lwd=1.8)}
```

##### 図 9.4 回帰係数の標準偏差の事後分布

```
hist(ext81$s_b,breaks=100,main="",ylab="",xlab="回帰係数の標準偏差")
```

##### 図 9.5 切片と回帰係数の散布図

```
plot(av,bv,pch=16,xlab="切片",ylab="回帰係数")
```

```
cor(av,bv);
```

切片と傾きの相関係数 p.131,1.3

##### 9.2 切片に分布を仮定したモデル

##### 9.2.1 高等学校別の中学の成績と高校の偏差値

##### 表 9.3 高等学校別の中学の成績と高校の偏差値  
(学力変化<-read.csv("./scrT/dat/学力変化.csv", header = TRUE))

##### 図 9.6 高校 2 と 9 の中・高校時の偏差値

```
X<-学力変化[!(学力変化$学校==2)|(学力変化$学校==9)], ]
plot(X$中学,X$高校,pch=X$学校,xlab="中学3年偏差値",
      ylab="高校1年偏差値", xlim=c(25,75),ylim=c(25,75))
legend(30,70,c("高校2","高校9"),pch=c(2,9))
```

##### 図 9.7 10 校の中学時と高校時の偏差値

```
plot(学力変化$中学, 学力変化$高校,pch=学力変化$学校,xlab="中学3年偏差値",
      ylab="高校1年偏差値", xlim=c(25,75),ylim=c(25,75))
```

##### まずは切片と回帰係数に分布を仮定したモデル

```
par<-c("sigma","a0","b0","s_a","s_b","a","b") #母数
dataSet821 <-list(n=length(学力変化$高校),J=max(学力変化$学校),
                  y=学力変化$高校,x=学力変化$中学, k=学力変化$学校)#入力
```

##### cmdstanr による実行

#コンパイル結果は 'mod81' を利用する

```
csrfit821 <- mod81$sample(data = dataSet821,chains = 5,iter_sampling = 20000,
                           iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit821 <- rstan::read_stan_csv(csrfit821$output_files()) #stan 形式への変換
```

##### rstan による実行

```
#fit821<-stan(model_code =HLM01,data=dataSet821,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit821, file="./scrT/obje/stan_obje0902")
#load(file="./scrT/obje/stan_obje0902"); #予め stan( ) で作った object をロード
```

```
ext821<-extract(fit821, par); #乱数の取り出し
```

##### 表 9.4 切片と回帰係数に分布のあるモデルの母数

##### の事後分布の要約 (偏差値データ)

```
gqcal(ext821$sigma)
gqcal(ext821$a0 )
gqcal(ext821$b0 )
gqcal(ext821$s_a )
gqcal(ext821$s_b )
```

##### 切片・回帰係数の EAP 推定値 (作図のためであり、教科書にはない)

```
av<-apply(ext821$a,2,mean)
bv<-apply(ext821$b,2,mean)
round(rbind(av,bv),2)
```

##### 図 9.8 高校別の 10 本の単回帰直線

```
plot(学力変化$中学, 学力変化$高校,pch=学力変化$学校,xlab="中学3年偏差値",
      ylab="高校1年偏差値", xlim=c(25,75),ylim=c(25,75),type="n")
for (i in 1:10) {abline(av[i],bv[i],lwd=1.8)}
```

##### 図 9.9 回帰係数の標準偏差の事後分布

```
hist(ext821$s_b,breaks=100,main="",ylab="",xlab="回帰係数の標準偏差")
```

##### 切片に分布を仮定したモデルの stan コード

```
HLM02<-'
data {
  int<lower=0> n; //全データ数
  int<lower=0> J; //群数
  vector[n] y; //基準変数
  vector[n] x; //予測変数
  int<lower=0> k[n]; //分類変数
}
parameters {
  vector[J] a; //切片
  real b; //回帰係数
  real<lower=0> sigma; //誤差標準偏差
  real a0; //切片平均
  real<lower=0> s_a; //切片 SD
}
transformed parameters {
  vector[n] yhat; //予測値
  for(i in 1:n)
    yhat[i] = a[k[i]] + x[i]*b; //(9.5 式),(9.8 式)
}
model {
  y ~ normal(yhat, sigma); //(9.6 式)
  a ~ normal(a0, s_a); //(9.7 式)
}
';
```

```
par<-c("sigma","a0","b","s_a","a") #母数
dataSet822 <-list(n=length(学力変化$高校),J=max(学力変化$学校),
                  y=学力変化$高校,x=学力変化$中学, k=学力変化$学校) #入力
```

##### cmdstanr による実行

```
modfile822 <- write_stan_file(HLM02) #書き出す一時ファイル
mod822 <- cmdstan_model(modfile822) #コンパイル
csrfit822 <- mod822$sample(data = dataSet822,chains = 5,iter_sampling = 20000,
                           iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit822 <- rstan::read_stan_csv(csrfit822$output_files()) #stan 形式への変換
```

##### rstan による実行

```
#fit822<-stan(model_code =HLM02,data=dataSet822,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit822, file="./scrT/obje/stan_obje0903")
#load(file="./scrT/obje/stan_obje0903"); #予め stan( ) で作った object をロード
```

```
ext822<-extract(fit822, par); #乱数の取り出し
```



##### 表 9.5 切片に分布があるモデルの母数の事後分布の要約

```
gqcal(ext822$sigma )
gqcal(ext822$a0    )
gqcal(ext822$b     )
gqcal(ext822$s_a   )
print('-----')
gqcal(ext822$a[,1] )
gqcal(ext822$a[,2] )
gqcal(ext822$a[,3] )
gqcal(ext822$a[,4] )
gqcal(ext822$a[,5] )
gqcal(ext822$a[,6] )
gqcal(ext822$a[,7] )
gqcal(ext822$a[,8] )
gqcal(ext822$a[,9] )
gqcal(ext822$a[,10])
```

##### 表 9.6 切片に関して高等学校 i 行が j 列より大きい確率

```
phc02(0,ext822$a,digits=2)
```

##### 図 9.10 高等学校 8 と 7 の切片の差が c 以上である確率

```
phc01(seq(2,8,0.2),ext822$a[,8],ext822$a[,7],xlab="高等学校 8 と 7 の切片の差")
```

##### 9. 3 切片と傾きに相関のあるモデル

##### 9. 3. 1 減量開始から 4 週間の体重変化

##### 表 9.7 減量開始から 4 週間の体重変化

```
(減量経過<-read.csv("./scrT/dat/減量経過.csv", header = TRUE))
```

##### 図 9.11 4 週間の体重変化 (4 人分)

```
Y<-減量経過[減量経過$女性==14,]
plot(Y$週数,Y$体重,pch=Y$女性,xlab="減量開始からの週数",
      ylab="体重",ylim=c(46,54),type="b")
for (i in 15:17){
  Y<-減量経過[減量経過$女性==i,]
  par(new=T)
  plot(Y$週数,Y$体重,pch=Y$女性,xlab="",ylab="",
        ylim=c(46,54),type="b")
}
legend(3.2,53.5,c("女性 14","女性 15","女性 16","女性 17"),pch=c(14:17))
#dev.copy2eps(file="z0811.eps",family='Japan1')
```

##### 図 9.12 4 週間の体重変化 (全員)

```
Y<-減量経過[減量経過$女性==1,]
plot(Y$週数,Y$体重,xlab="減量開始からの週数",
      ylab="体重",ylim=c(46,54),type="l")
for (i in 2:50){
  Y<-減量経過[減量経過$女性==i,]
  par(new=T)
  plot(Y$週数,Y$体重,xlab="",ylab="",ylim=c(46,54),type="l")
}
```

```
par<-c("sigma","a0","b0","s_a","s_b","a","b") #母数
dataSet91 <-list(n=length(減量経過$体重), J=max(減量経過$女性),
                y=減量経過$体重, x=減量経過$週数, k=減量経過$女性) #入力
```

#####cmdstanr による実行

#コンパイル結果は 'mod81' を利用する

```
csrfit91 <- mod81$sample(data = dataSet91,chains = 5,iter_sampling = 20000,
                        iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit91 <- rstan::read_stan_csv(csrfit91$output_files()) #stan 形式への変換
```

#####rstan による実行

```
#fit91<-stan(model_code =HLM01,data=dataSet91,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit91, file="./scrT/obje/stan_obje0904")
#load(file="./scrT/obje/stan_obje0904"); #予め stan( ) で作った object をロード
```

```
ext91<-extract(fit91, par); #乱数の取り出し
```

##### 表 9.8 切片と回帰係数に分布を仮定したモデルの事後分布の要約

```
gqcal(ext91$sigma)
gqcal(ext91$a0    )
gqcal(ext91$b0    )
gqcal(ext91$s_a   )
gqcal(ext91$s_b   )
```

##### 切片・回帰係数の EAP 推定値

```
av<-apply(ext91$a,2,mean)
bv<-apply(ext91$b,2,mean)
round(rbind(av,bv),2)
```

##### 図 9.13 切片と回帰係数の散布図

```
plot(av,bv,pch=16,xlab="切片",ylab="回帰係数"); abline(h=0.0)
```

```
cor(av,bv); #切片と傾きの相関係数 p.137,1.6
```

#####切片と傾きに相関のあるモデルの stan コード

```
HLM03<-'
data {
  int<lower=0> n; //全データ数
  int<lower=0> J; //群数
  vector[n] y; //基準変数
  vector[n] x; //予測変数
  int<lower=0> k[n]; //分類変数
}
parameters {
  matrix[J,2] ab; //切片・回帰係数
  real<lower=0> sigma; //誤差標準偏差
  real a0; //切片平均
  real b0; //回帰係数平均
  real<lower=0> s_a; //切片 SD
```

```

    real<lower=0> s_b;                //回帰係数 SD
    real<lower=-1,upper=1> rho;      //切片と回帰係数の相関
}
transformed parameters {
    vector[n] yhat;                  //予測値
    vector[2] mu;                    //切片・回帰係数平均
    cov_matrix[2] Sigma;             //切片・回帰係数の分散共分散行列
    mu[1] = a0;                      //切片の平均を  $\mu$  の第 1 要素に代入
    mu[2] = b0;                      //回帰係数の平均を  $\mu$  の第 2 要素に代入
    Sigma[1,1] = pow(s_a,2);          //切片の分散
    Sigma[2,2] = pow(s_b,2);          //回帰係数の分散
    Sigma[1,2] = s_a*s_b*rho;         //切片と回帰係数の共分散
    Sigma[2,1] = Sigma[1,2];          //同上
    for(i in 1:n)
        yhat[i] = ab[k[i],1] + x[i]*ab[k[i],2]; // (9.10) 式 予測値の構造
}
model {
    //y は yhat の周りで正規分布に従う
    y ~ normal(yhat, sigma);          // (9.11) 式 ベクトル表記で for 文なし
    for(j in 1:J)                    //切片と回帰係数は 2 変量正規分布に従う
        ab[j,]~multi_normal(mu,Sigma); // (9.12) 式
}
';

par<-c("sigma","a0","b0","s_a","s_b","rho","ab") #母数
dataSet92 <-list(n=length(減量経過$体重), J=max(減量経過$女性),
               y=減量経過$体重, x=減量経過$週数, k=減量経過$女性 ) #入力

#####cmdstanr による実行
modfile92 <- write_stan_file(HLM03) #書き出す一時ファイル
mod92 <- cmdstan_model(modfile92) #コンパイル
csrfit92 <- mod92$sample(data = dataSet92,chains = 5,iter_sampling = 20000,
                        iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit92 <- rstan::read_stan_csv(csrfit92$output_files()) #stan 形式への変換

#####rstan による実行
#fit92<-stan(model_code =HLM03,data=dataSet92,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit92, file="./scrT/obje/stan_obje0905")
#load(file="./scrT/obje/stan_obje0905"); #予め stan( ) で作った object をロード

ext92<-extract(fit92, par); #乱数の取り出し

##### 表 9.9 切片と回帰係数に相関のあるモデルの母数の事後分布の要約
gqcal(ext92$sigma)
gqcal(ext92$a0 )
gqcal(ext92$b0 )
gqcal(ext92$s_a )
gqcal(ext92$s_b )
gqcal(ext92$rho )

##### 切片・回帰係数の EAP 推定値
av<-apply(ext92$ab[, ,1],2,mean)

```

```

bv<-apply(ext92$ab[, ,2],2,mean)
round(rbind(av,bv),2)

```

```

#### 図 9.14 50 本の単回帰直線
plot(減量経過$週数, 減量経過$体重,pch=減量経過$女性,
     xlab="週数",ylab="体重",type="n")
for (i in 1:50) {abline(av[i],bv[i],lwd=1.8)}

```

```

##### 図 9.15 切片と回帰係数の相関の事後分布
hist(ext92$rho,breaks=100,main="",ylab="",xlab="切片と回帰係数の相関")

```

```

cor(av,bv); #切片と傾きの相関係数（教科書にはなし）

```

```

##### 9. 4 レベル 2 の変数があるモデル
##### 9. 4. 1 勤続年数と年収の分析

```

```

#####表 9.10 企業別の勤続年数と年収
(勤続年収<-read.csv("./scrT/dat/勤続年収.csv", header = TRUE))

```

```

#####表 9.11 企業の資本金の規模
(資本金 <-read.csv("./scrT/dat/資本金.csv", header = TRUE))

```

```

##### 3 つの企業の勤続年数と年収の散布図（教科書にはない）
X<-勤続年収 [(勤続年収$企業==3)|(勤続年収$企業==15)|
              (勤続年収$企業==21)], ]
plot(X$勤続年,X$年収,pch=X$企業,xlab="入社からの年数",ylab="年収")
legend(2.5,850,c("企業 3","企業 15","企業 21"),pch=c(3,15,21))

```

```

#####図 9.16 勤続年数と年収の分析
plot(勤続年収$勤続年, 勤続年収$年収,pch=勤続年収$企業,
     xlab="入社からの年数",ylab="年収"); #図 9.16

```

```

#####レベル 2 の変数があるモデルの stan コード
HLM04<- '
data {
    int<lower=0> n;                //全データ数
    int<lower=0> J;                //群数
    vector[n] y;                  //基準変数
    vector[n] x;                  //予測変数
    vector[J] z;                  //レベル 2 の変数
    int<lower=0> k[n];             //分類変数
}
parameters {
    vector[J] a;                  //切片
    vector[J] b;                  //回帰係数
    real<lower=0> sigmae;          //誤差標準偏差
    real a0;                      //切片の切片
    real b0;                      //回帰係数の切片
    real ca;                      //切片の係数
    real cb;                      //回帰係数の係数

```

```

    real<lower=0> sigmaa;          //切片 SD
    real<lower=0> sigmab;         //回帰係数 SD
  }
  transformed parameters {
    vector[n] yhat;              //予測値
    for(i in 1:n)
      yhat[i] = a[k[i]] + x[i]*b[k[i]]; // (9.14) 式
  }
  model {
    y ~ normal(yhat, sigmae);     // (9.15) 式
    a ~ normal(a0+ z*ca, sigmaa); // (9.16) 式
    b ~ normal(b0+ z*cb, sigmab); // (9.17) 式
  }
';

par<-c("sigmae","a0","b0","ca","cb","sigmaa","sigmab","a","b")#母数
dataSet93 <-list(n=length(勤続年収$年収),J=max(勤続年収$企業),
  y=勤続年収$年収, x=勤続年収$勤続年, z=資本金$区分, k=勤続年収$企業 )#入力

#####cmdstanr による実行
modfile93 <- write_stan_file(HLM04)          #書き出す一時ファイル
mod93 <- cmdstan_model(modfile93)             #コンパイル
csrfit93 <- mod93$sample(data = dataSet93,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit93 <- rstan::read_stan_csv(csrfit93$output_files()) #stan 形式への変換

#####rstan による実行
#fit93<-stan(model_code =HLM04,data=dataSet93,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit93, file="./scrT/obje/stan_obje0906")
#load(file="./scrT/obje/stan_obje0906"); #予め stan( ) で作った object をロード

ext93<-extract(fit93, par);                  #乱数の取り出し

##### 表 9.12 レベル 2 の変数があるモデルの母数の事後分布
gqcal(ext93$sigmae)
gqcal(ext93$a0)
gqcal(ext93$b0)
gqcal(ext93$ca)
gqcal(ext93$cb)
gqcal(ext93$sigmaa)
gqcal(ext93$sigmab)

#### 切片・回帰係数の EAP 推定値
av<-apply(ext93$a,2,mean)
bv<-apply(ext93$b,2,mean)
round(rbind(av,bv),2)

#####図 9.17 レベル 1 の 25 本の回帰直線
plot(勤続年収$勤続年, 勤続年収$年収,xlab="勤続年",ylab="年収",type="n")
for (i in 1:25) {abline(av[i],bv[i],lwd=1.8,lty=(資本金 [i,2]+3))}
legend(2.5,900,paste("区分",-2:2,sep=""),lty=1:5)

```

```

#####図 9.18 レベル 2 の変数による切片に対する回帰直線
plot(資本金 [,2],av,ylab="レベル 1 の切片",xlab="資本金区分",pch=16)
abline(a=mean(ext93$a0),b=mean(ext93$ca))

```

```

#####図 9.19 レベル 2 の変数による回帰係数に対する回帰直線
plot(資本金 [,2],bv,ylab="レベル 1 の回帰係数",xlab="資本金区分",pch=16)
abline(a=mean(ext93$b0),b=mean(ext93$cb))

```

```
cor(av,bv);          #切片と傾きの相関係数(教科書にはなし)
```

##### 9.5 傾きが共通でレベル 2 の質的変数があるモデル

#####表 9.13 特別英語教育を実施しているか否かに関するレベル 2 の変数  
(学力変化<-read.csv("./scrT/dat/学力変化.csv", header = TRUE))  
特別英語<-c(0,0,1,1,0,0,1,1,1,1)

```

#####傾きが共通でレベル 2 の質的変数があるモデルの stan コード
HLM05<- '
data {
  int<lower=0> n;          //全データ数
  int<lower=0> J;          //群数
  vector[n] y;            //基準変数
  vector[n] x;            //予測変数
  vector[J] z;            //レベル 2 の変数
  int<lower=0> k[n];       //分類変数
}
parameters {
  vector[J] a;            //切片
  real b;                 //回帰係数
  real<lower=0> sigmae;    //誤差標準偏差
  real a0;                //切片平均
  real ca;                //切片の係数
  real<lower=0> sigmaa;    //切片 SD
}
transformed parameters {
  vector[n] yhat;         //予測値
  for(i in 1:n)
    yhat[i] = a[k[i]] + x[i]*b; // (9.18) 式
}
model {
  y ~ normal(yhat, sigmae); // (9.19) 式
  a ~ normal(a0+ z*ca, sigmaa); // (9.20) 式, (9.21) 式
}
';

```

```

par<-c("sigmae","a0","ca","sigmaa","b","a")          #母数
dataSet94 <-list(n=length(学力変化$高校),J=max(学力変化$学校),
  y=学力変化$高校,x=学力変化$中学, z=特別英語, k=学力変化$学校)#入力

```

```
#####cmdstanr による実行
modfile94 <- write_stan_file(HLM05)          #書き出す一時ファイル
mod94 <- cmdstan_model(modfile94)            #コンパイル
csrfit94 <- mod94$sample(data = dataSet94,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit94 <- rstan::read_stan_csv(csrfit94$output_files()) #stan 形式への変換

#####rstan による実行
#fit94<-stan(model_code =HLM05,data=dataSet94,
# pars=par, seed=1234, chains=5, warmup=1000, iter=21000)
#save(fit94, file="./scrT/obje/stan_obje0907")
#load(file="./scrT/obje/stan_obje0907"); #予め stan( ) で作った object をロード

ext94<-extract(fit94, par);                  #乱数の取り出し

### 表 9.14 傾きが共通でレベル 2 の質的変数があるモデルの母数の事後分布の要約
gqcal(ext94$sigmae)
gqcal(ext94$a0)
gqcal(ext94$ca)
gqcal(ext94$sigmaa)
gqcal(ext94$b)

#### 切片・回帰係数の EAP 推定値
av<-apply(ext94$a,2,mean)
bv<-mean(ext94$b)
round(rbind(av,bv),2)

##### 図 9.20 特別教育をしているか否かによる回帰直線の相違
plot(学力変化$中学, 学力変化$高校,xlab="中学 3 年偏差値",
  ylab="高校 1 年偏差値", xlim=c(25,75),ylim=c(25,75),type="n")
for (i in 1:10) {
  abline(av[i],bv,lwd=1.9,lty=((特別英語 [i])*(-1)+2) )
}
legend(30,70,c("特別教育あり","特別教育なし"),lty=c(1,2))

mean(ext94$ca>0); #特別教育の効果がある確率 p.143,1.3

##### 図 9.21 特別教育をしている高校の切片の平均の差が c 以上である確率
phc01(seq(0,10,0.5),ext94$ca,0,xlab="切片の差")

##### 実習課題

#
# 正解は割愛
#
```

## 9.2 以下の関数を自分で復習しよう

rbind

## 9.3 自習問題

教科書 第 9 章 p.144, 9.7 節 実践課題

ただし高校 2 と高校 5 は除くものとする。

## 9.4 第 9 章 宿題

第 1 回の授業の宿題で、p.15, 1.6 節 実習課題 をした。この 2 変量データを 4 人の友達からもらいなさい。ただしもらった人からは 2 人までしかデータをあげてはいけません。以下のような階層線形モデルで分析する

- 基準変数：目測
- 予測変数：実測（第 1 章とは逆なので注意）
- レベル 1 の観測対象：パスタ or 鉛筆 10 本
- レベル 2 の観測対象：被験者 5 人

9.2 節 「切片に分布を仮定したモデル」で分析する。

（できる人は傾きを 1.0 に固定してみよう）

- ”知覚長 11.csv” や ”学力変化.csv” に合わせてデータを前処理する。  
図 9.2 知覚された長さの実験、に準じて 5 種類の記号で生データの散布図を示せ。
- 「表 9.5 切片に分布があるモデルの母数の事後分布の要約」に相当する表を作れ。a1 ～ a5 は氏名を書くこと。
- 「図 9.8 高校別の 10 本の単回帰直線」に準じ 5 本の（この場合は平行な）回帰直線を描き、氏名を描き込め。
- 「表 9.6 切片に関して高等学校  $i$  行が  $j$  列より大きい確率」に準じて  $5 \times 5$  の phc 行列を作表せよ。実質的に差がある必要条件を満たすペアを 1 つ挙げよ。
- 図 9.10 に準じて、その 2 名の切片の差が  $c$  以上である確率の曲線を描き、解釈せよ。

## 10 間接質問法

### 10.1 スクリプト

```
#####第 10 章 間接質問法
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd()) #working directory(wd) の確認
(setwd('./scrT')) #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd)) #wd 戻す
library(rstan) #パッケージ rstan の呼び出し
library(cmdstanr) #パッケージ cmdstanr の呼び出し
library(posterior) #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

#####本章で利用する関数
#MAP 推定値を与える関数
MAP<-function(ext){density(ext)$x[which.max(density(ext)$y)]}

##### 10. 1 混合分布
##### 図 10.1 大人と子供の身長ヒストグラム
(身長<-read.csv("../scrT/dat/身長.csv", header = TRUE))
hist(身長$x,breaks = 12,col=8,main="",xlab="身長 (cm)",ylab="人数");#図 10.1

#####混合分布モデルの stan コード
mix01<-'
data {
    //入力部
    int<lower=0> n; //データ数
    real x[n]; //データ
    real<lower=0,upper=1> p; //混合比率
}
parameters {
    //母数定義部
    ordered[2] mu; //平均
    real<lower=0> sigma[2]; //標準偏差
}
model {
    //モデル記述部
    for(i in 1:n)
        target += log_sum_exp( // (10.9) 式
            log(p) + normal_lpdf(x[i] | mu[1], sigma[1]),
            log(1-p) + normal_lpdf(x[i] | mu[2], sigma[2])
        );
}
';

par<-c("mu","sigma") #母数
dataSetmix <-list(n=length(身長$x),x=身長$x,p=0.4) #入力

#####cmdstanr による実行
modfilemix <- write_stan_file(mix01) #書き出す一時ファイル
modmix <- cmdstan_model(modfilemix) #コンパイル
csrfitmix <- modmix$sample(data = dataSetmix,chains = 5,iter_sampling = 20000,
```

```
init = function(){list(mu=c(138,172),sigma=c(5.8,6.8))},
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitmix <- rstan::read_stan_csv(csrfitmix$output_files()) #stan 形式への変換

#####rstan による実行
#fitmix<-stan(model_code =mix01,data=dataSetmix,
# pars=par,seed=1234,chains=1,warmup=400,iter=2400)
#save(fitmix,file="./scrT/obje/stan_obje1001")
#load(file="./scrT/obje/stan_obje1001"); #予め stan( ) で作った object をロード

extmix<-extract(fitmix, par); #乱数の取り出し

##### 表 10.1 混合正規モデルの事後分布の要約
print(fitmix,digits_summary=3, probs =prob)

##### 10. 2 ランダム回答法
yes<-1168; no<-3072;
x<-c(rep(1,yes),rep(0,no)); #危険ドラッグの経験比率
2.0*(yes/(yes+no))-0.5 ; #簡便解 (10.11) 式

##### ランダム回答法の stan コード
random01<-'
//ランダム回答法
data {
    //入力部
    int<lower=0> n; //データ数
    int<lower=0> x[n]; //データ
    real<lower=0,upper=1> q; //マスク比率
}
parameters {
    //母数定義部
    real<lower=0,upper=1> p; //知りたい比率
}
model {
    //モデル記述部
    for(i in 1:n)
        target += log_sum_exp( // (10.12) 式の対数の指数の和の対数
            log(0.5)+bernoulli_lpmf(x[i] | p),
            log(0.5)+bernoulli_lpmf(x[i] | q)
        );
}
';

par<-c("p") #母数
dataSetrand <-list(n=length(x),x=x,q=0.5) #入力

#####cmdstanr による実行
modfilerand <- write_stan_file(random01) #書き出す一時ファイル
modrand <- cmdstan_model(modfilerand) #コンパイル
csrfitrand <- modrand$sample(data=dataSetrand,chains=5,iter_sampling=20000,
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitrand <- rstan::read_stan_csv(csrfitrand$output_files()) #stan 形式への変換

#####rstan による実行
```

```
#fitrand<-stan(model_code =random01,data=dataSetrand,
# pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fitrand,file="./scrT/obje/stan_obje1002")
#load(file="./scrT/obje/stan_obje1002"); #予め stan( ) で作った object をロード
```

```
extrand<-extract(fitrand, par); #乱数の取り出し
```

```
##### 表 10.2 危険ドラッグの経験比率の事後分布の要約
print(fitrand,digits_summary=3, probs =prob)
```

```
##### 図 10.2 危険ドラッグの経験比率の事後分布
hist(extrand$p,breaks=50,main="",xlab="比率",ylab="",xlim=c(0,0.1),
cex.lab=2.0,cex.axis=1.5)
#dev.copy2eps(file="z0902.eps",family='Japan1')
```

```
MAP(extrand$p); #MAP 推定値 p.150,1.1
```

```
##### 図 10.3 p(p > c) という研究仮説が正しい確率
phc01(seq(0,0.10,0.005),extrand$p,0,xlab="比率 > c")
#dev.copy2eps(file="z0903.eps",family='Japan1')
```

### ##### 10. 3 Aggregated Response 法 ##### 10. 3. 1 異性と性の行為の経験人数

```
x<-c(
-3,-2, 1, 3,-9, 4,-1,-3,-5, 2, 0,-7,-1,-4,-3,10, 2, 6, 2,-2,-1,-6,-6,-1, 8,-6,
-6, 1, 6, 1, 1, 5, 6, 1, 2, 9, 3, 3,-2, 5, 6, 5, 0, 7,-3,-1,-1, 6, 6, 4, 4,-8,
5, 2, 7, 7, 5,-1, 6,10, 9, 2,-8,-6,-8,-7,-2, 7, 4, 6,-5, 0,-1,14, 0, 5, 9,-2,
7, 4,-2, 2,-8,-3,11,-5,-1,-5, 3, 6,-5,-4, 4, 4, 2,10,-3, 1, 3, 5,-6,-3,-1, 2,
-9, 2,-4, 3,-1, 1,-6, 7, 4,-5, 0, 7, 4,-7,-4,-3, 2, 2,-3, 6,-4,-6,-3, 0, 5, 3,
6,-2, 0,10, 9, 1, 8,10, 3, 4, 5, 4, 9,-5, 8,-1, 1,-7, 2, 9,-2, 5,-7,-1, 4,
1, 3, 1, 2, 0,-4, 1,-5, 3,-6,-4,-4, 2,-1,-3,-2,-8,-3,-4, 1, 0, 1, 0, 7,-4, 7,
10, 4, 9,11, 7,10,-1, 5, 5, 0, 9,-3,-7,-6, 9, 3, 0, 7, 2, 7, 0, 3,-1,-3,-8, 0,
0,10, 7, 1,-2,-3,-9,-7,-4, 6,-2,11,13, 8, 8,-4, 0,-1,-8, 7,10, 8, 3,10, 4,-6,
6, 7,-5, 4,-8, 7,-7,-3, 2,-8, 3, 3,-2, 2,-3,-4,-5,-2, 7,-3, 2,-1,-7,-1,-6,-6,
10,11,-5, 6, 7, 5,-6, 8,-4,-1, 4,-2,-1,-6, 6,-5,11, 5, 1, 3, 0,-2,-7,-1,-6, 9,
-6,-7,-6, 4, 3,10, 4, 3,-1, 5, 7, 6,-4, 3, 0,12,11, 7, 8, 4,-2, 1,-6, 6,-1,-1,
5, 4, 6,11,-7, 4,-1, 1,-2, 2,-6,-3,-7, 2,-5, 4, 2,-3)
```

```
##### 図 10.4 回答数字のヒストグラム
hist(x,main="",xlab="回答した数字",ylab="",
xlim=c(-9.5,14.5),cex.lab=2.0,cex.axis=1.5,breaks=24)
#dev.copy2eps(file="z0904.eps",family='Japan1')
```

```
mean(x); #平均 簡便解 (10.16) 式
max(x);min(x); #最大・最小
```

### ##### Aggregated Response 法の stan コード

```
AR01<-'data {
//Aggregated Response 法
int<lower=0> n; //データ数
int<lower=0> m; //マスク数
```

```
int x[n]; //データ
int mask[m]; //マスク要素
real pro; //一様分布の確率
}
parameters {
real<lower=0> lambda; //母数定義部
//ポアソン分布母数
}
model {
//モデル記述部
//そのデータからは存在するマスク数
//
//可能性のあるマスクの数の逆数
real j;
int jj;
real pr;
vector [m] lp;
for(i in 1:n){
j = 0.0;
for(k in 1:m) if (0 <= x[i]-mask[k]){ j =j +1.0;}
pr = 1.0/j;
jj=0;
for(k in 1:m){ //k はマスク
if (0 <= x[i]-mask[k]) { //可能性のあるマスクだけ計算
jj=jj+1;
lp[jj] = log(pr)+poisson_lpmf(x[i]-mask[k]|lambda);/(10.24) 式
}
}
target += log_sum_exp( lp[1:jj] );
}
}
';
```

```
par<-c("lambda") #母数
mask<-c(-9:0,0:9); m<-length(mask) #マスク
dataSetAR <-list(n=length(x),m=m,x=x,mask=mask,pro=1/m) #入力

#####cmdstanr による実行
modfileAR <- write_stan_file(AR01) #書き出す一時ファイル
modAR <- cmdstan_model(modfileAR) #コンパイル
csrfitAR <- modAR$sample(data = dataSetAR,chains = 5,iter_sampling = 20000,
iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitAR <- rstan::read_stan_csv(csrfitAR$output_files()) #stan 形式への変換
```

```
#####rstan による実行
#fitAR<-stan(model_code =AR01,data=dataSetAR,
# pars=par,seed=1234,chains=5,warmup=1000,iter=21000,init=mean(x))
#save(fitAR,file="./scrT/obje/stan_obje1003")
#load(file="./scrT/obje/stan_obje1003"); #予め stan( ) で作った object をロード
```

```
extAR<-extract(fitAR, par); #乱数の取り出し
```

```
##### 表 10.3 λ の事後分布の要約
print(fitAR,digits_summary=3, probs =prob)
```

```
##### 図 10.5 ポアソン分布の母数の事後分布
hist(extAR$lambda,breaks=100,main="",xlab="λ",ylab="",cex.lab=2,cex.axis=1.5)
```

```
#dev.copy2eps(file="z0905.eps",family='Japan1')

##### 図 10.6  $p(\lambda > c)$  という研究仮説が正しい確率
phc01(seq(0.5,2.5,0.1),extAR$lambda,0,xlab="平均  $\lambda > x$ ")
#dev.copy2eps(file="z0906.eps",family='Japan1')

##### 図 10.7 経験人数の事後予測分布
T<-length(extAR$lambda);pre<-numeric(T);          #格納の箱
for(i in 1:T){ pre[i]<-rpois(1,extAR$lambda[i])} #事後予測分布
barplot(table(pre),axes=F, xlab="経験人数",cex.lab=2.0,cex.axis=2.0)
#dev.copy2eps(file="z0907.eps",family='Japan1')

##### 表 10.4 経験人数の確率と累積確率
round(table(pre)/T,3)          #確率
round(cumsum(table(pre)/T),3) #累積確率
summary(pre)                   #平均・中央値・最大値
round(sd(pre),3)               #標準偏差

##### 10.4 アイテムカウント法
##### 表 10.5 短リストと長リストに対する該当数による度数分布
a<-c(107,133,160,80,75,110, 96)
b<-c( 98,142,166,75,82,104,135, 0)
K<-6

##### アイテムカウント法の stan コード
itemcount02<-'
data {
    int<lower=0> K;          //入力部
    int<lower=0> a[K+1];     //短リストのカテゴリ数
    int<lower=0> b[K+2];     //短リストの反応数
    int<lower=0> b[K+2];     //長リストの反応数
}
transformed data {
    //データ変換部
}
parameters {
    simplex[K+1]    theta;   //母数定義部
    simplex[K+1]    pi;      //短リストの確率
    real<lower=0,upper=1.0> p; //p による単リストの条件付き確率
    //キー項目にあてはまる確率
}
transformed parameters {
    //母数変換部
    vector[K+1]    pi1;     //短リストが k でマスクが 1 となる同時確率
    simplex[K+2]    theas;   //長リストの母比率
    real temp;
    pi1=pi*p;
    theas[1]=theta[1]-pi1[1]; // (10.38) 式
    for (j in 2:K+1){
        theas[j]=theta[j]-pi1[j]+pi1[j-1]; // (10.39) 式
    }
    temp=0;
    for(j in 1:K+1){ temp=temp+theas[j];}
    theas[K+2]=1-temp;
}
model {
    //モデル記述部
```

```
a ~ multinomial(theta);          //短リストの多項分布
b ~ multinomial(theas);          //長リストの多項分布
}
';

par<-c("p","theta","theas")     #母数
dataSet 違法 <-list(K=K,a=a,b=b) #入力

##### cmdstanr による実行
modfile 違法 <- write_stan_file(itemcount02)          #書き出す一時ファイル
mod 違法 <- cmdstan_model(modfile 違法)              #コンパイル
csrfit 違法 <- mod 違法$sample(data=dataSet 違法,chains=5,iter_sampling=20000,
    iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit 違法 <- rstan::read_stan_csv(csrfit 違法$output_files()) #stan 形式への変換

##### rstan による実行
#fit 違法<-stan(model_code =itemcount02,data=dataSet 違法,
# pars=par,seed=1234,chains=5,warmup=1000,iter=21000)
#save(fit 違法,file="./scrT/obje/stan_obje1004")
#load(file="./scrT/obje/stan_obje1004"); #予め stan( ) で作った object をロード

ext 違法<-extract(fit 違法, par);          #乱数の取り出し

##### 表 10.6 p の事後分布の要約
print(fit 違法,digits_summary=3, probs =prob)
```

## 10.2 MAP: MAP 推定値を与える

MAP(ext)

## 指数

ext 事後分布あるいは事後予測分布に従う乱数

## 10.3 混合分布モデル

```
for(i in 1:n)
    target += log_sum_exp(          // (10.9) 式
        log(p) +normal_lpdf(x[i]|mu[1],sigma[1]),
        log(1-p)+normal_lpdf(x[i]|mu[2],sigma[2])
    )
```

);

この部分には幾つかの新出の関数がある。

- `target +=` は、「`target =target +対数尤度`」の意味である。尤度の場合は総積となるが、対数尤度の場合は総和となる。
- `log_sum_exp( )` は、指数 (`exp`) の和 (`sum`) の対数を計算する関数である。
- `normal_lpdf( )` は、正規分布の確率密度の対数変換値を返す関数である。
- 後の節で登場する `bernoulli_lpmf( )` は、ベルヌイ分布の確率関数の対数変換値を返す関数である。
- 後の節で登場する `poisson_lpmf( )` は、ポアソン分布の確率関数の対数変換値を返す関数である。

## 10.4 自習問題

10.1 節の混合分布モデルでは、入門ということで、大人と子供の混合率は既知 ( $p = 0.4$ ) であると仮定して母数の推定を求めた。しかし実践場面では、ライフスタイルや消費志向など、潜在したクラスの混合率は未知であることが多い。

初期値を 0.4 として、混合率  $p$  も同時に推定する stan スクリプトを書きなさい。

- 以下のスクリプトを R の新規エディタにコピーして利用する。
- 添え字は 02 をつけている。
- stan 側で書き換えるのは、//入力部、//母数定義部、である。
- R 側で書き換えるのは、、#母数、#入力、#初期値、である。

```
#####混合分布モデルの stan コード
mix02<-"
data {                                     //入力部

}

parameters {                             //母数定義部

}

model {                                   //モデル記述部
  for(i in 1:n)
    target += log_sum_exp(               //(10.9) 式
      log(p) +normal_lpdf(x[i]|mu[1],sigma[1]),
      log(1-p)+normal_lpdf(x[i]|mu[2],sigma[2])
    );
}
";

par02<-c(                                )      #母数
dataSetmix02 <-list(                      )      #入力

#####cmdstanr による実行
modfilemix02 <- write_stan_file(mix02)        #書き出す一時ファイル
modmix02 <- cmdstan_model(modfilemix02)        #コンパイル
csrfitmix02 <- modmix02$sample(data=dataSetmix02,chains=5,iter_sampling=2000,
  init = function(){list(                    )},#初期値
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitmix02 <- rstan::read_stan_csv(csrfitmix02$output_files())#stan への変換

extmix02<-extract(fitmix02, par02);           #乱数の取り出し

##### 表 10.1 相当の 混合正規モデルの事後分布の要約
print(fitmix02,digits_summary=3, probs =prob)
```

## 10.5 第 10 章 宿題

「たのしいベイズモデリング：事例で拓く研究のフロンティア, 北大書房」(拙編著、心理事務学読にあります)に、早稲田大学文学部 (心理統計学・心理学概論) の学生から収集した実際のデータが分析されている。

- 第 1 巻 第 1 章 大学生は 18 禁映像をどれくらい見ているか——2 要因配置の



AR 法――

- 第 2 巻 第 2 章 正直に回答しづらい違法行為の経験率の推定――アイテムカウント法――
- 第 2 巻 第 3 章 本当のこと, 教えてもらいます!――ランダム回答法――

以上 3 つの章を読み、データ収集法・結果に対する感想・論評を、それぞれ 350 字以上ー 500 字以内で述べよ。

## 11 項目反応理論

### 11.1 スクリプト

```
##### 第 11 章 項目反応理論
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())           #working directory(wd) の確認
(setwd('./scrT'))         #wd の移動
source('../myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))             #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
library(psych)
library(irtoys)
library(ltm)
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義

#####11. 1 2 値項目の項目特性曲線
##### 11. 1. 3 項目困難度
##### 図 11.1 項目困難度の役割
(ip<-matrix(cbind(1.7,-2:2,0),5,3))
plot(irf(ip),main="")
text(-2.5,0.5,"b= - 2",cex=1.5)
text(-1.5,0.5,"b= - 1",cex=1.5)
text(-0.5,0.5,"b= 0",cex=1.5)
text( 0.5,0.5,"b= 1",cex=1.5)
text( 1.5,0.5,"b= 2",cex=1.5)
abline(v=0)
abline(v=1)
#dev.copy2eps(file="z1001.eps",family='Japan1');

round(irf(ip,x=0.0)[[2]],3) #p.165,b1.3  $\theta=0.0$  の正答率
round(irf(ip,x=1.0)[[2]],3) #p.165, 1.2  $\theta=1.0$  の正答率

##### 11. 1. 4 項目識別力
##### 図 11.2 項目識別力の役割
(ip<-matrix(cbind(1.7*c(0.5,1,2),0,0),3,3))
plot(irf(ip),main="")
text(-2.5,0.2,"a=0.5",cex=1.5)
text(-1.4,0.17,"a=1.0",cex=1.5)
text(-0.4,0.05,"a=2.0",cex=1.5)
abline(v=-2)
abline(v=1)
#dev.copy2eps(file="z1002.eps",family='Japan1')

round(irf(ip[c(1,3),],x=-2)[[2]],3) #p.164,b1.5  $\theta=-2.0$  の正答率
round(irf(ip[c(1,3),],x= 1)[[2]],3) #p.165,b1.2  $\theta= 1.0$  の正答率

##### 表 11.1 指導性・リーダーシップ尺度の項目 (2 値)
a<-c(0.607,0.854,0.637,1.789,0.985,0.446)
```

```
b<-c(-0.947,-0.047,-0.250,-0.744,-0.289,-0.253)
(ip<-matrix(cbind(1.7*a,b,0),6,3))
```

##### 図 11.3 指導性・リーダーシップの尺度の項目の ICC

```
plot(irf(ip),main="")
text(-0.4,0.9, 4,cex=2); #識別力が高い
text( 2,0.8, 6,cex=2); #識別力が低い
text(-1.5,0.45,1,cex=2); #困難度低い
text(-0.1,0.4, 2,cex=2); #困難度高い
#dev.copy2eps(file="abcd.eps",family='Japan1')
```

#####指導性・リーダーシップテストの分析

```
miki<-c(
"私の意見は仲間に反映されることが多い",
"人を引っ張っていく力がある",
"集団の中ではリーダーシップを発揮する",
"仲間に明確なアドバイスを与えることができる",
"自分は仲間の中でまとめ役である",
"対立している仲間の仲裁が得意である")
I<-length(miki);u<-numeric(I)
#for (i in 1:I){ #自分でやってみる場合のデータ収集
# u[i]<-menu(c("いいえ","はい"), graphics = F,title = miki[i])}
```

u<-c(2, 1, 2, 1, 1, 1); #例題データ (11.11) 式

#####多値項目の正規累積項目反応モデルの stan コード

```
irt<-'
data {
  int<lower=2> K; //反応カテゴリ数
  int<lower=1> p; //項目数
  int<lower=1,upper=K> u[p]; //試験結果 (1 から K まで)
  real a[p]; //識別力
  matrix [p,K-1] b; //困難度
  real mu; //平均 (事前分布)
  real <lower=0> sigma; //標準偏差 (事前分布)
}
parameters {
  real theta; //被験者母数
}
model {
  vector[K] pro; //反応確率
  theta ~ normal(mu,sigma); //尺度値事前分布,(11.9) 式右辺後半
  for(j in 1:p){
    pro[1]=1-Phi(a[j]*(theta-b[j,1])); // (11.14) 式,(11.22) 式,(11.6) 式
    for (k in 2:(K-1)){
      pro[k]=Phi(a[j]*(theta-b[j,k-1]))
        -Phi(a[j]*(theta-b[j,k])); // (11.15) 式,(11.22-25) 式
    }
    pro[K]=Phi(a[j]*(theta-b[j,K-1])); // (11.16) 式,(11.26) 式,(11.6) 式
    u[j] ~ categorical(pro); //尤度,(11.8) 式
  }
}
```

```

generated quantities{          //生成量
  real hensati;                //偏差値
  hensati=theta*10+50;
}
';

par<-c("theta","hensati")      #母数
a<-c(0.607,0.854,0.637,1.789,0.985,0.446) #識別力
b<-c(-0.947,-0.047,-0.250,-0.744,-0.289,-0.253) #困難度
dataSetirt02 <-list(K=2,p=I, u=u,a=a,b=matrix(b,I,1),mu=0,sigma=1)

#####cmdstanr による実行
modfileirt02 <- write_stan_file(irt)          #書き出す一時ファイル
modirt02 <- cmdstan_model(modfileirt02)        #コンパイル
csrfitirt02<-modirt02$sample(data=DataSetirt02,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitirt02 <- rstan::read_stan_csv(csrfitirt02$output_files()) #stan 形式への変換

#####rstan による実行
#fitirt02<-stan(model_code =irt,data=DataSetirt02,
# pars=par,seed=1234,chains=5,warmup=200,iter=4200)
#save(fitirt02,file="./scrT/obje/stan_obje1101")
#load(file="./scrT/obje/stan_obje1101"); #予め stan( ) で作った object をロード

irt02<-extract(fitirt02, par);                #乱数の取り出し

##### 表 11.2 尺度値  $\theta$  と偏差値の事後分布の要約統計量
print(fitirt02,digits_summary=3, probs =prob)

#####11.2 3 値項目の段階反応モデル

##### 図 11.4 ICC と IRCCC の関係
p1<-function(theta){pnorm(theta-(-1))}
p2<-function(theta){pnorm(theta-1)}
plot(p1,-3,3,ylim=c(0,1),xlab="",ylab="",lwd=2)
par(new=T)
plot(p2,-3,3,ylim=c(0,1),xlab="",ylab="",lwd=2)
text(-0.7,0.9,"f(u=1|  $\theta=0$ )",cex=2);
text(0.0,0.5,"f(u=2|  $\theta=0$ )",cex=2);
text(0.7,0.1,"f(u=3|  $\theta=0$ )",cex=2);
arrows(0,0,0,0.15,code=3,angle=15,lwd=1.8)
arrows(0,0.17,0,0.83,code=3,angle=15,lwd=1.8)
arrows(0,0.87,0,1,code=3,angle=15,lwd=1.8)
text(-1.6,0.4,"p1( $\theta$ )",cex=2);
text( 1.6,0.6,"p2( $\theta$ )",cex=2);          #図 11.4 ここまで
#dev.copy2eps(file="z1004.eps",family='Japan1')

##### 11.2.1 社会的外向性テスト
miki<-c(
"1 話し好きである",

```

```

"2 人と広く付き合うほうだ",
"3 無口である",
"4 自分はわりと人気者だ",
"5 生き生きしていると人に言われる",
"6 陽気である",
"7 初対面のひとには自分のほうから話しかける",
"8 よく人から相談を持ちかけられる",
"9 話題には事欠かないほうだ",
"10 誰とでも気さくに話せる")
I<-length(miki);u<-numeric(I)
#for (i in 1:I){                      #自分でやってみる場合のデータ収集
#  u[i]<-menu(c("あてはまらない","どちらともいえない",
#              "あてはまる"), graphics = F,title = miki[i])}
#u[3]<- (-1)*u[3]+4;    #逆転項目処理
u<-c(3,3,2,1,1,3,3,1,3,3)          #例題データ (11.17) 式

##### 表 11.3 社会的外向性尺度の項目 (3 値)
a<-c(1.241,1.096,0.856,0.735,0.761,1.331,0.710,0.528,0.883,1.461); #識別力
b<-matrix(
  c(-1.283, 0.086,
    -1.060, 0.199,
    -1.333,-0.021,
    -0.974, 1.989,
    -0.395, 1.379,
    -1.388, 0.146,
    -1.015, 0.768,
    -1.291, 1.221,
    -0.585, 1.180,
    -0.979, 0.306),10,2,T);          #困難度
p1<-function(theta){pnorm(aa*(theta-b1))} ; # (11.12) 式
p2<-function(theta){pnorm(aa*(theta-b2))} ; # (11.13) 式
q1<-function(theta){1-p1(theta)} ;        # (11.14) 式
p21<-function(theta){p1(theta)-p2(theta)} ; # (11.15) 式

##### 図 11.5 困難度レベルの違い p.170
##### 図 11.6 識別力の違い p.170
##### 図 11.7 困難度の差の違い p.171
par(mfrow=c(3,3))
for(i in c(1:9)){
  aa<-a[i];b1<-b[i,1];b2<-b[i,2]
  plot(q1,-3,3,ylim=c(0,1),xlab="",ylab="")
  par(new=T)
  plot(p21,-3,3,ylim=c(0,1),xlab="",ylab="")
  par(new=T)
  plot(p2,-3,3,ylim=c(0,1),xlab=miki[i],ylab="",cex.lab=1.5)
}
par(mfrow=c(1,1))
#dev.copy2eps(file="z13043.eps",family='Japan1')

par<-c("theta","hensati")          #母数
dataSetirt03 <-list(K=3,p=I, u=u,a=a,b=b,mu=0,sigma=1) #入力

```

```
#####cmdstanr による実行
#コンパイルは 'modirt02' を利用する
csrfitirt03<-modirt02$sample(data=dataSetirt03,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitirt03 <- rstan::read_stan_csv(csrfitirt03$output_files()) #stan 形式への変換
```

```
#####rstan による実行
#fitirt03<-stan(model_code =irt, data=,
# pars=par,seed=1234,chains=5,warmup=200,iter=4200)
#save(fitirt03,file="./scrT/obje/stan_obje1102")
#load(file="./scrT/obje/stan_obje1102"); #予め stan( ) で作った object をロード
```

```
irt03<-extract(fitirt03, par); #乱数の取り出し
```

```
#表 11.4 尺度値  $\theta$  と偏差値の事後分布の要約統計量
print(fitirt03,digits_summary=3, probs =prob)
```

#### ##### 11.3 5 値項目の段階反応モデル

```
##### 11.3.2 共感性テスト
miki<-c(
"人の話をじっくり聞くことが得意である",
"人から悩みを相談されることが多い",
"涙もろく、もらい泣きをするほうである",
"他人の立場にたって共感することができる",
"悲しんでいる相手の話を聞くと、同様の悲しさを感じてしまう",
"仲間の関心事には積極的に興味を示す")
I<-length(miki);u<-numeric(I)
for (i in 1:I){ #自分でやってみる場合のデータ収集
# u[i]<-menu(c("あてはまらない","ややあてはまらない","どちらともいえない",
# "ややあてはまる","あてはまる"), graphics = F,title = miki[i])}
```

```
u<-c(2, 3, 3, 2, 5, 4) #例題データ (11.28) 式
```

```
##### 表 11.5 共感性尺度の項目 (5 値)
a<-c(1.974,1.665,1.407,1.543,1.008,1.770) #識別力
b<-matrix(c(
-2.164, -0.499, 0.149, 1.426,
-2.452, -0.517, 0.598, 2.265,
-2.434, -0.968, -0.345, 0.945,
-2.043, -0.402, 0.634, 2.103,
-4.150, -1.679, -0.348, 1.863,
-2.208, -0.727, 0.278, 1.644),6,4,T) #困難度
```

```
###各種関数の定義 p1 (11.18) 式, p2 (11.19) 式, q1 (11.22) 式は定義済
p3<-function(theta){pnorm(aa*(theta-b3))}; # (11.20) 式
p4<-function(theta){pnorm(aa*(theta-b4))}; # (11.21) 式
p21<-function(theta){p1(theta)-p2(theta)}; # (11.23) 式
p32<-function(theta){p2(theta)-p3(theta)}; # (11.24) 式
p43<-function(theta){p3(theta)-p4(theta)}; # (11.25) 式
```

```
##### 図 11.8 項目識別力の役割
par(mfrow=c(3,2))
for(i in 1:6){
aa<-a[i];b1<-b[i,1];b2<-b[i,2];b3<-b[i,3];b4<-b[i,4]
plot(q1,-3,3,ylim=c(0,1),xlab="",ylab="")
par(new=T)
plot(p21,-3,3,ylim=c(0,1),xlab="",ylab="")
par(new=T)
plot(p32,-3,3,ylim=c(0,1),xlab="",ylab="")
par(new=T)
plot(p43,-3,3,ylim=c(0,1),xlab="",ylab="")
par(new=T)
plot(p4,-3,3,ylim=c(0,1),xlab=miki[i],ylab="",cex.lab=1.5)
}
par(mfrow=c(1,1))
#dev.copy2eps(file="z1306.eps",family='Japan1')
```

```
par<-c("theta","hensati") #母数
dataSetirt05 <-list(K=5,p=I, u=u,a=a,b=b,mu=0,sigma=1) #入力
```

```
#####cmdstanr による実行
#コンパイルは 'modirt02' を利用する
csrfitirt05<-modirt02$sample(data=dataSetirt05,chains=5,iter_sampling=20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fitirt05 <- rstan::read_stan_csv(csrfitirt05$output_files()) #stan 形式への変換
```

```
#####rstan による実行
#fitirt05<-stan(model_code =irt,data=dataSetirt05,
# pars=par,seed=1234,chains=5,warmup=200,iter=4200)
#save(fitirt05,file="./scrT/obje/stan_obje1103")
#load(file="./scrT/obje/stan_obje1103"); #予め stan( ) で作った object をロード
```

```
irt05<-extract(fitirt05, par); #乱数の取り出し
```

```
##### 表 11.6 尺度値  $\theta$  と偏差値の事後分布の要約統計量
print(fitirt05,digits_summary=3, probs =prob)
```

#### ##### 11.4 2 値項目の 3 母数項目特性曲線

```
##### 図 11.9 3 母数モデルの ICC
(ip<-matrix(cbind(1.7,0,c(0.2,0.25,0.33, 0.5)),4,3))
plot(irf(ip),main="")
text(-3.5,0.55,"c= 0.5",cex=1.5)
text(-3.5,0.40,"c= 0.33",cex=1.5)
text(-3.5,0.29,"c= 0.25",cex=1.5)
text(-3.5,0.16,"c= 0.2",cex=1.5)
#dev.copy2eps(file="z1009.eps",family='Japan1')
```

```
##### 図 11.10 3 母数モデルの 10 項目の ICC
a<-c(1.55,1.82,1.58,1.81,1.76,1.95,1.53,1.66,1.75,1.55); #識別力
b<-c(0.23,0.33,0.00,0.07,-0.98,0.61,-0.26,0.06,0.17,-0.54);#困難度
```

```

c<-c(0.15,0.00,0.09,0.15,0.18,0.04,0.11,0.10,0.13,0.14); #当て推量母数
ip<-cbind(a,b,c)
plot(irf(ip),main="")
#dev.copy2eps(file="z1010.eps",family='Japan1')

u<-c(1,1,1,1,2,1,2,2,1,2) #例題データ p.176,b1.2

#####2 値項目の 3 母数項目特性曲線の stan コード
irt_guess<-'
data {
  int<lower=1> p; //項目数
  int<lower=1,upper=2> u[p]; //試験結果
  real a[p]; //識別力
  real b[p]; //困難度
  real c[p]; //当て推量
  real mu; //平均 (事前分布)
  real <lower=0> sigma; //標準偏差 (事前分布)
}
parameters {
  real theta; //被験者母数
}
model {
  vector[2] pro; //反応確率
  theta ~ normal(mu,sigma); //尺度値事前分布
  for(j in 1:p){
    pro[2]=c[j]+(1-c[j])*Phi(a[j]*(theta-b[j])); //(11.29) 式
    pro[1]=1-pro[2];
    u[j] ~ categorical(pro); //尤度
  }
}
generated quantities{
  real hensati; //生成量
  hensati=theta*10+50; //偏差値
}
';

par<-c("theta","hensati")
dataSetirt_guess <-list(p=length(u), u=u,a=a,b=b,c=c,mu=0,sigma=1)

#####cmdstanr による実行
modfileirt_guess <- write_stan_file(irt_guess) #書き出す一時ファイル
modirt_guess <- cmdstan_model(modfileirt_guess) #コンパイル
csrfirt_guess<-modirt_guess$sample(data=datasetirt_guess,chains=5,
  iter_sampling = 20000,iter_warmup=1000,parallel_chains=5,seed=1234) #MCMC
fitirt_guess<-rstan::read_stan_csv(csrfirt_guess$output_files())#stan 形へ変換

#####rstan による実行
#fitirt_guess<-stan(model_code =irt_guess,data=datasetirt_guess,
# pars=par,seed=1234,chains=5,warmup=200,iter=4200)
#save(fitirt_guess,file="./scrT/obje/stan_obje1104")
#load(file="./scrT/obje/stan_obje1104"); #予め stan( ) で作った object をロード

```

```

irt_guess<-extract(fitirt_guess, par); #乱数の取り出し

##### 表 11.7 3 母数モデルによる尺度値 と偏差値の事後分布の要約
print(fitirt_guess,digits_summary=3, probs =prob)

```

```

#####実習課題
#自分でデータを作ったら以下はやらない
#irt02<-extract(fitirt02, par)
#irt03<-extract(fitirt03, par)
#irt05<-extract(fitirt05, par)

#####phc 曲線
par(mfrow=c(3,1));
phc01(seq(30,50,0.5),irt02$hensati,cc="gtc",byoga="yes",xlab="指導性・リーダーシップ
尺度 (2 値)")
phc01(seq(45,65,0.5),irt03$hensati,cc="gtc",byoga="yes",xlab="社会的外向性尺度 (3
値)")
phc01(seq(40,55,0.5),irt05$hensati,cc="gtc",byoga="yes",xlab="共感性尺度の項目 (5
値)")
par(mfrow=c(1,1))

#####phc テーブル
phc01(seq(30,35,1),irt02$hensati,cc="gtc",byoga="no")
phc01(seq(45,50,1),irt03$hensati,cc="gtc",byoga="no")
phc01(seq(43,48,1),irt05$hensati,cc="gtc",byoga="no")

```

## 11.2 自習問題

1. 表 11.1 の質問に回答し、表 11.2 を作成し、解釈せよ。  
解釈とは、この場合、点推定値と自覚している性格との異同、確信区間によるその確実度である。
2. 表 11.3 の質問に回答し、表 11.4 を作成し、解釈せよ。

## 11.3 第 11 章 宿題

私の共感性を評価する。まず表 11.5 の共感性尺度の項目 (5 値) に回答しなさい。

1. 自身の尺度値を推定し、出力例の表 11.6 に従って、 $\theta$  と偏差値の事後分布の要約統計量を報告しなさい。

2. 推定された偏差値と、自信が思う自分の共感性性格を比較し、一致していたか否かを中心に、テスト結果を論じなさい（150 字程）。
3. 自分の偏差値の  $\text{phc}(c \mid \text{偏差値})$  の曲線を描きなさい。
4.  $\text{phc}(c < \text{偏差値})$  と  $\text{phc}(c > \text{偏差値})$  のテーブルを作成しなさい。
5. 2 つのテーブルを、解釈しなさい。

## 12 予測変数を直交化した重回帰分析

### 12.1 スクリプト

```
##### 第 12 章 予測変数を直交化した重回帰分析
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())                #working directory の確認
(setwd('./scrT'))
source('./myfunc/myfunc.R')     #自作関数の読み込み
(setwd(n_wd))
#install.packages("conjoint")
#install.packages("DoE.base")
library("conjoint")
library("DoE.base")

##### パッケージ conjoint による "data.frame" による直交表の作成
#384 種類の全てのプロファイルの作成
(omiaai_every<-expand.grid(
  喫煙=c("禁煙","喫煙有"),
  年齢=c("29 歳迄","制限無"),
  職業=c("公務員","限定無"),
  年収=c("500 万以上","限定無"),
  学歴=c("大学卒","高校卒"),
  結婚=c("初婚","限定無"),
  お酒=c("お酒無","お酒有"),
  場所=c("公民館","ホテル","客船上")
))
#384 種類から 44 プロファイルで直交表を作る。
#それでも多すぎる大変だから、パッケージ conjoint は使わない
(omiaai_con<-caFactorialDesign(data=omiaai_every,type="orthogonal"))
nrow(omiaai_con)

##### 直交表は、パッケージ DoE.base によって作る
#乱数を使用しているので、seed を指定して再現性を確保する
#24 プロファイルで優れている、クラスは"design"
omiaai_design<-oa.design(factor.names=list(
  喫煙=c("禁煙","喫煙有"),
  年齢=c("29 歳迄","制限無"),
  職業=c("公務員","限定無"),
  年収=c("500 万以上","限定無"),
  学歴=c("大学卒","高校卒"),
  結婚=c("初婚","限定無"),
  お酒=c("お酒無","お酒有"),
  場所=c("公民館","ホテル","客船上")
),seed=1234)

#DoE.base で作った直交表をパッケージ"conjoint"で分析する。
#そのためには"data.frame"に変換する必要がある。水準は数字で表現される
(omiaai_doe<-caEncodedDesign(omiaai_design))

#バージョンが変わると同じ seed に対して、"oa.design"は同一の直交表を
```

#作らないので"omiaai\_doe"は保存しておいた方がよい。  
#omiaai\_design の方は分析には使わない、事実、以下に読みこむものとは違う

##### 表 12.1 「直交お見合いデータ」に使用する直交表 (水準)  
(omiaai\_design<-read.csv(file="scrT/dat/お見合い水準表.csv",header=T))

##### 表 12.2 「直交お見合いデータ」に使用する直交表 (ダミー変数) の<<別表現>>  
#数字で水準を表現する (分析にはこちらを使用する)  
(omiaai\_doe<-read.csv(file="scrT/dat/お見合い直交表.csv",header=T))

```
#水準レベルの作成      p.181,b1.7
omiaai_lab<-c("禁煙","喫煙有",
              "29 歳迄","制限無",
              "公務員","限定無",
              "500 万以上","限定無",
              "大学卒","高校卒",
              "初婚","限定無",
              "お酒無","お酒有",
              "公民館","ホテル","客船上")
```

##### 表 12.3 「直交お見合いデータ」  
(omiaai\_data<-read.csv(file="scrT/dat/お見合い.csv"))

##### 表 12.4 「直交お見合いデータ」の部分効用値 中の「全体」  
Conjo<-Conjoint(y=omiaai\_data,x=omiaai\_doe,z=omiaai\_lab)

##### 図 12.2 部分効用値の図示 (の左図) 全体の描画  
source("scrT/myfunc もうひとつの重回帰分析.R")  
omiaai\_util<-caUtilities(y=omiaai\_data,x=omiaai\_doe,z=omiaai\_lab)  
plot.Conjoint(utility=omiaai\_util,label=omiaai\_lab)  
text(0.4,2,"全体",cex=1.7)

##### 回答者ごとの効用値 (全員分、教科書にはない)  
(sub\_util <- caPartUtilities(y=omiaai\_data,x=omiaai\_doe,z=omiaai\_lab))

##### 図 12.2 部分効用値の図示 (の右図) 回答者 8 の描画  
plot.Conjoint(utility=sub\_util[8,],label=omiaai\_lab)  
text(0.4,2,"被験者 8",cex=1.7)  
#dev.copy2eps(file="z1202.eps",family='Japan1')

##### 表 12.4 「直交お見合いデータ」の部分効用値 中の s1 から s8  
t(sub\_util)[,1:8]

##### 表 12.5 「直交お見合いデータ」の相対重要度  
subimport<-sub.imp(data=omiaai\_data,design=omiaai\_doe,label=omiaai\_lab)  
import<-round(colMeans(subimport),1); #「全体」の相対重要度の計算  
cbind(import,round(t(subimport)[,1:8],1)); #「全体」と回答者を同時に表示  
round(apply(cbind(import,t(subimport)[,1:8]),2,sum),0);#和が 100 になる確認

##### 表 12.6 直交表に登場したプロファイルの効用値  
#総効用値の出力 (1 つ 1 つ計算できるが、一度に全部は出ない)

```
conjyo.sim(data=omiaai_data,design=omiaai_doe,lab=omiaai_lab,num=3)

#回答者ごとの総効用値      (これは教科書にはない)
#omiaai_data[i,] に回答者番号 i を入れる。この例は 1 番目の回答者
conjyo.sim(data=omiaai_data[1,],design=omiaai_doe,label=omiaai_lab,num=3)
```

## 12.2 自習問題

「心理学演習 4」にて案出した、「基準変数に影響を与えるであろう 7 つの要因とその 2 水準」に関して 関数 `oa.design()` によって、 $8 \times 7$  の直交表を作りなさい。

## 12.3 第 12 章 宿題

授業内レポート、自習時間に作った直交表を使い、以下を報告せよ。

1. 基準変数名：変数の内容と、その変数を予測することによって生じる利便を説明せよ。
2. その基準変数に影響を与えるであろう 7 つの要因とその 2 水準をあげよ。
3. 表 12.1 に相当する  $8 \times 7$  の（自習時間に作った）直交表を示しなさい。
4. 例を参考にして、8 つの質問を作り、アンケート調査をするための調査票を作成しなさい。



## 13 質的研究における飽和率・寡占度

### 13.1 スクリプト

##### 第13章 質的研究における飽和率・寡占度

```
# Working directory が 'toyoda' であることの確認
(n_wd<-getwd())          #working directory(wd) の確認
(setwd('./scrT'))        #wd の移動
source('./myfunc/myfunc.R') #自作関数の読み込み
(setwd(n_wd))            #wd 戻す
library(rstan)            #パッケージ rstan の呼び出し
library(cmdstanr)         #パッケージ cmdstanr の呼び出し
library(posterior)        #パッケージ posterior の呼び出し
prob<-c(0.025, 0.05, 0.5, 0.95, 0.975) #確率点の定義
```

##### 13.1 質的知見収集の特徴

###表 13.1 大学入試方法の改善に関する進路指導担当教員からの自由記述意見の度数

```
N<- 91;          #対象数
x<- 1:N;         #対象名ベクトル
f1<-
c(282,266,265,237,189,163,152,137,135,131,129,99,98,94,93,93,80,78,73,72,58,
58,57,56,52,51,45,44,40,39,39,35,32,29,28,28,27,25,23,23,22,20,20,19,
17,16,15,14,14,14,13,11,11,7 ,7 ,6)
f2<-
c(6,6,6,5,5,5,5,4,4,4,4,4,3,3,3,3,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1)
f<-c(f1,f2)
names(f)<-x
cumsum(f)/sum(f);    #累積確率  p.196, 第1パラグラフ
```

#####図 13.1 自由記述意見のヒストグラムの外枠

```
plot(f,type="l",ylim=c(0,290),xlab="順位",ylab="",
,cex.axis=1.8, cex.lab=2.0,lwd=2.5)
lines(c(1,91),c(0,0),lwd=2.5)
lines(c(1,1),c(0,f[1]),lwd=2.5)
lines(c(54,54),c(0,f[54]),lwd=2.5)
lines(c(74,74),c(0,f[74]),lwd=5.5)
text(25,170,"7.5%の時間, 59%の知見",cex=2.0)
text(25,150,"累積確率 97.1%", cex=2.0)
text(65,75,"22.5%の時間, 81%の知見",cex=2.0)
text(65,55,"累積確率 99.4%", cex=2.0)
arrows(65,45,60,f[60],lwd=2.0)
arrows(25,140,16,f[16],lwd=2.0);    #図 13.1 ここまで
#dev.copy2eps(file="z1301.eps",family='Japan1')
```

#####第13章で利用する関数ここから

#13.2.1 ジップ分布 確率関数 (13.1) 式、(13.2) 式

```
zipf_pmf<-function(r,s,N,Q=0){
  if (Q==1) {zeta<-myzeta(s,1000,Q)}
  if (Q==0) {zeta<-myzeta(s,N,Q)}
  (1/(r^s))/zeta
}
```

}  
#13.2.3 ゼータ関数の近似 (13.6) 式

```
myzeta<-function(s,m,Q){
  zeta<-0
  for (n in 1:m){zeta <- zeta + (1/(n^s))}
  if (Q==1) { zeta<- zeta -((m^(1-s))/(1-s))}
  return(zeta)
}
```

#13.2.4 飽和率 zipf 分布の累積分布関数 (13.9) 式

```
zipf_cdf<-function(r,s,N,Q=0){
  pro<-0
  for (i in 1:r){
    pro<-pro+zipf_pmf(i,s,N,Q)
  }
  return(pro)
}
```

#####第13章で利用する関数ここまで

#####図 13.2 ジップ分布の形状と寡占度 s

```
N<- 8;          #対象数
x<- 1:N;        #対象名ベクトル
p<- numeric(N); #zipf 分布の確率分布
names(p)<-x
for (s in seq(0.6,1.2,0.2)){
  for (i in 1:N) {p[i]<-zipf_pmf(r=i,s=s,N=N, Q=0)}
}
plot(p,type="l",ylim=c(0,0.5),xlab="順位",ylab="確率",
,cex.axis=1.8, cex.lab=1.3,lwd=2.0)
par(new=T)
}
text(1.5,0.42,"s=1.2",cex=1.4)
text(1.0,0.38,"s=1.0",cex=1.4)
text(1.0,0.28,"s=0.8",cex=1.4)
text(1.5,0.18,"s=0.6",cex=1.4)
par(new=F);    #図 13.2 ここまで
#dev.copy2eps(file="z1302.eps",family='Japan1')
```

#####13.2.3 ゼータ関数の近似

#####小数 6 桁までの正確な値

```
(pi^2)/6;          #(13.7) 式
(pi^4)/90;         #(13.8) 式
```

#####近似式の精度

```
myzeta(2,400,1); myzeta(2,4000,1);    #p.199,1.2
myzeta(4, 30,1); myzeta(4, 40,1);     #p.199,1.6
```

##### ジップ分布と、独立サンプルの一致の確認 (教科書にはない)

#状況の設定

```
n<- 1800;          #被検者数
a<- 5;             #非復元抽出数
N<- 20;            #対象数
```

```

s<- 1.5;          #zipf 分布の母数
bp<- 0.05;        #途中で想起をやめる確率、最低一つは想起する
x<- 1:N;          #対象名ベクトル
p<- numeric(N);   #zipf 分布の確率分布
pro<-c(0.025, 0.5, 0.975); #確率点の定義
par<-c("s","prob"); #s は zipf の母数、prob は被覆率

##### zipf 分布の確率分布表の作成
names(p)<-x
for (i in 1:N) {p[i]<-zipf_pmf(r=i,s=s,N=N, Q=0)}
maxy<-ceiling(p[i]*10)/10
plot(p,type="b",ylim=c(0,maxy),xlab="")
print(p);

#a × n 個の zipf 分布に従う乱数を発生
#(グラフが一致し正しく発生されていることを確認)
Gv<-sample(x,a*n,prob=p, replace =T)
Gf<-table(Gv);      #非復元抽出データの度数表
Gs<-Gf/(n*a);       #復元抽出データの相対度数
plot(Gs, ylim=c(0,maxy),lty=1,lwd=5,ylab="",xlab="",main="")
par(new=T)
plot(p, ylim=c(0,maxy),type="l",xaxt="n",lwd=2,ylab="",xlab="",main="")
##### ジップ分布と、独立サンプルの一致の確認 (ここまで)

13.3.1 2 番目以降に抽出される知見

##### 図 13.3 ジップ分布と、そこから非復元抽出した分布の食い違い
#a 個非復元抽出数したときの n 人分のデータを作る。確率 bp で想起止める。
F<-matrix(0,n,a)
for (i in 1:n){
  y<-numeric(0)
  repeat{
    t1<-sample(x,1,prob=p, replace =T)
    if (is.element(t1,y)==0) {y<-c(y,t1)}
    if (length(y)==a) break
    if (sample(c(0,1),1,prob=c(bp,1-bp))==0) break
  }
  F[i,1:(length(y))]<-y;      #復元抽出データ
}
head(F,n=50)

#zipf 分布と、そこから a 個非復元抽出した場合の相対度数が食い違うことを確認
Fv<-as.vector(ifelse(F==0,NA,F))
Ff<-table(Fv);      #非復元抽出データの度数表
Fs<-Ff/(n*a);       #非復元抽出データの相対度数
plot(Fs, ylim=c(0,maxy),lty=1,lwd=5,ylab="",xlab="",main="")
par(new=T)
plot(p, ylim=c(0,maxy),type="l",xaxt="n",lwd=2,ylab="",xlab="",main="")
#dev.copy2eps(file="zipf01.eps",family='Japan1');      #図 13.3 ここまで

##### 13.4 インタビュー調査の寡占度・飽和率・遭遇率

```

```

##### 表 13.2 就職活動全体を通じての心得・不安の入力データと人数
(F<-read.csv(file="scrT/dat/順位不安.csv",header=T))
N<-max(F)
n<-nrow(F)
a<-ncol(F)

##### a 個非復元抽出する場合の zipf 分布の推定の stan コード
zipfa<-’
functions{
  real zeta(real s, int m, int Q){
    real ii;
    real zet;
    ii=0;
    zet=0;
    for(i in 1:m) {
      ii=ii+1;
      zet=zet+inv(pow(ii,s));          //与えられた項数までの和
    }
    if (Q==1) zet=zet-((pow(m,1-s))/(1-s)); //無限の場合は近似式で補正
    return(zet);
  }
}
data {
  int<lower=0>      Q;      //無限モデルなら 1, 有限モデルなら 0
  int<lower=0>      N;      //観察された対象の数 (13.9) 式の r *
  int<lower=0>      n;      //被験者数
  int<lower=0>      a;      //非復元抽出数の最大値
  real<lower=-1>    r[n,a]; //ランクデータ 欠測値はゼロ 無効は-1 とする
}
transformed data {
  real lb;          //s の下限値
  lb=0.1;
  if (Q==1) lb=1.001; //発散より少しだけ大きい値を指定する
}
parameters {
  real<lower=lb>    s;      //母数 (寡占度)
}
model {
  real zip;          //復元抽出での r[i,j] の確率
  real temp;         //抽出済の対象の確率の和
  real zet;          //ゼータ関数の値
  if (Q==1) {zet=zeta(s,1000,Q);} //対象数が無限の場合は項数 1000 で近似
  if (Q==0) {zet=zeta(s,N,Q);}   //対象数が有限の場合の zeta
  for(i in 1:n) {          // i 番目の人の処理開始
    temp = 0;
    for(j in 1:a) {        // i 番目の人の j 番目の回答の処理開始
      if (r[i,j]>0) {      // 欠測や無効でないものだけ処理
        zip = inv(pow(r[i,j],s))/zet;
        target += log(zip)-log(1-temp); // (13.19) 式
        temp = temp + zip;
      }
    }
  }
}

```

```

}
}
generated quantities{
  real      prob; //飽和率
  real      Lmprob;//遭遇率
  real      zet;
  if (Q==1) {zet=zeta(s,1000,Q);} //対象数が無限の場合は項数 1000 で近似
  if (Q==0) {zet=zeta(s,N,Q);} //対象数が有限の場合の zeta
  prob=0;
  for(i in 1:N) prob=prob+inv(pow(i,s)); //(13.9) 式, (13.1) 式, (13.2) 式
  prob=prob/zet;
  Lmprob=1-prob;
}
';

par<-c("s","prob","Lmprob"); #s は寡占度、prob は飽和率、Lmprob は遭遇率
dataSet02 <-list(Q=1,N=N,n=n,a=a,r=F)

#####cmdstanr による実行
modfile02 <- write_stan_file(zipfa)          #書き出す一時ファイル
mod02 <- cmdstan_model(modfile02)           #コンパイル
csrfit02 <- mod02$sample(data = dataSet02,chains = 5,iter_sampling = 20000,
  iter_warmup = 1000,parallel_chains = 5,seed=1234) #MCMC
fit02 <- rstan::read_stan_csv(csrfit02$output_files()) #stan 形式への変換

#####rstan による実行
#fit02<-stan(model_code =zipfa,data=dataSet02,
# pars=par,seed=1234,chains=4,warmup=200,iter=10200)
#save(fit02, file="./scrT/obje/stan_obje1301")
#load(file="./scrT/obje/stan_obje0301"); #予め stan( ) で作った object をロード

extlec02<-extract(fit02, par);              #乱数の取り出し

##### 表 13.3 寡占度・飽和率・遭遇率の事後分布の要約
print(fit02,digits_summary=3, probs =pro)

##### 図 13.4 寡占度の事後分布
hist(extlec02$s, breaks=100,cex.axis=2.0,cex.lab=2.0,
  ylab="", xlab="寡占度", main="");
#dev.copy2eps(file="z1304.eps",family='Japan1')#図 13.4

##### 図 13.5 飽和率の phc 曲線
phc01(seq(1.30,1.45,0.01),extlec02$s,cc="ltc",byoga="yes",xlab="寡占度")
phc01(seq(0.75,0.90,0.01),extlec02$prob,cc="gtc",byoga="yes",xlab="飽和率")
#dev.copy2eps(file="z1305.eps",family='Japan1')

##### 表 13.4 飽和率の phc テーブル
phc01(c(0.76,0.78,0.80,0.82),extlec02$prob,cc="gtc",byoga="no",xlab="飽和率")

#実習課題
F<-read.csv(file="scrT/dat/順位 ES.csv",header=T)

```

```

N<-max(F)
n<-nrow(F)
a<-ncol(F)
par<-c("s","prob","Lmprob"); #s は寡占度、prob は飽和率、Lmprob は遭遇率
#a 個非復元抽出する場合の zipf 分布の推定 (r が 順位 data 行列)
fit02<-stan(model_code =zipfa,data=list(Q=1,N=N,n=n,a=a,r=F),
  pars=par,seed=1234,chains=4,warmup=200,iter=10200)
print(fit02,digits_summary=4, probs =pro)
extlec02<-extract(fit02, par)
hist(extlec02$s, breaks=200,cex.axis=2.0);
gqcal(extlec02$s,4)

```

## 13.2 R の自作関数

### 13.2.1 zipf\_pmf: ジップ分布 確率関数

```
zipf_pmf(r,s,N,Q=0)
```

## 引数

r 順位 (スカラー)

s 母数

N 対象の総数 (無限モデルの場合は無視される)

Q 論理値、確率分布の N が無限なら 1, 有限なら 0

## 戻り値

ジップ分布で r の対象が観察される確率

### 13.2.2 myzeta: ゼータ関数の (近似) 値

```
myzeta(s,m,Q)
```

## 引数

**s** 母数

**m** 和をとる項数

**Q** 論理値、確率分布の  $N$  が無限なら 1, 有限なら 0

## 戻り値

ゼータ関数の値、有限の場合は正確な値、無限の場合は近似値

### 13.2.3 zipf\_cdf: 飽和率 zipf 分布の累積分布関数

```
zipf_cdf(r,s,N,Q=0)
```

## 引数

**r** 順位 (スカラー)

**s** 母数

**N** 対象の総数 (無限モデルの場合は無視される)

**Q** 論理値、確率分布の  $N$  が無限なら 1, 有限なら 0

## 戻り値

$r$  までの累積確率

## 13.3 stan の自作関数

### 13.3.1 zeta: ゼータ関数の (近似) 値

```
real zeta(real s, int m, int Q)
```

## 引数

**s** 母数

**m** 対象の総数 (無限モデルの場合は, 近似のための項数)

**Q** 論理値、確率分布の  $N$  が無限なら 1, 有限なら 0

## 戻り値

ゼータ関数の (近似) 値

## 13.4 自習問題

今回はなし

## 13.5 第 13 章 宿題

教科書 p.205, 13.7 節 実習課題 をしなさい。