

分布を求める】

- (A) スクリプト Stan-Ymet-Xnom2grp-MrobustHet-Example.R と Stan-Ymet-Xnom2grp-MrobustHet.R の関数を修正し、事前分布を図示してみよう。詳細については、セクション 14.3.4, p. 420 を参照してほしい。変更点を説明し、グラフ出力を示してみよう。Stan に収束の問題が起るか否かを検討してみよう。
- (B) 事前分布からのサンプルを得るために JAGS のプログラムを修正してみよう。JAGS における事前分布からのサンプリング方法については、セクション 8.5, p. 214 を見返そう。
- (C) 上記 2 点をもとに、効果量と 2 群の差についての事前分布はよい具合に「曖昧」かどうかを議論しよう。

第17章

1つの量的変数で量的変数を予測する

Contents

17.1.	単純な線形回帰	486
17.2.	ロバスト線形回帰	487
17.2.1.	JAGS によるロバスト線形回帰	491
17.2.1.1.	MCMC サンプリングのためにデータを標準化する	492
17.2.2.	Stan によるロバスト線形回帰	495
17.2.2.1.	漠然とした事前分布の定数	496
17.2.3.	Stan か JAGS か	497
17.2.4.	事後分布の解釈	498
17.3.	個人と集団の階層回帰モデル	499
17.3.1.	JAGS によるモデルの実装方法	501
17.3.2.	事後分布：縮小と予測	504
17.4.	線形 2 次式とデータの重み付け	505
17.4.1.	結果と解釈	509
17.4.2.	モデルのさらなる拡張	510
17.5.	モデル拡張の手続きとリスク	511
17.5.1.	事後予測チェック	511
17.5.2.	JAGS あるいは Stan によるモデル拡張の手続き	512
17.5.3.	パラメータの追加に伴うリスク	513
17.6.	エクササイズ	516

The agri-bank's threatnin' to revoke my lease
(借金取りがやってくる)

If my field's production don't rapid increase.

(畑の生産増えない限り、お前の貸付取り消すと)

Oh Lord how I wish I could divine the trend,

(ああ神様よ、占いたまえ)

Will my furrows deepen? and Will my line end?

(深まる筋を辿った先に、どんな未来があるのかを)¹⁾

本章では、身長から体重を予測したり、体重から血圧を予測したり、また教育年数から収入を予測するといったことを試みたい。身長や体重はいずれも量的変数である。したがって、ここでは予測変数と被予測変数がどちらも量的変数である場合の分析を取り扱って

1) 本章では（他の話の中でも特に）線形回帰について論じている。線形回帰を使って分析する人は、将来に向けた傾向を探査したいと考えているものだ。冒頭の詩では、深まる畠筋 (furrows), あるいは額に深まる皺筋 (furrows) に、未来への希望や憂いが表現されている。この「深まる筋を辿った先」は単純な線形回帰を意味しているが、家系のつながりも意味している。

いく。

まず、被予測変数 y と予測変数 x の関係を単純な線形モデルで記述してみよう。予測値とデータの残差に対して正規分布を仮定した、いわゆる「単純な回帰分析」の話である。このモデルを3つの方法で一般化する。1つ目は、正規分布の代わりに、前章で説明した t 分布を用いた外れ値を含むノイズ分布を与える。このモデルについては、JAGS と Stan の両方でコードを実装する。2つ目は、被予測変数と予測変数の間に直線ではない関係（例えば2次式）を仮定したモデルを紹介する。3つ目は、個々のトレンドと、それらを踏まえた集團レベルのトレンドを推定する階層モデルについて議論を展開させる。

なお、本章を読むにあたっては、第15章の一般化線形モデル(GLM)で紹介した、1つの量的予測変数を用いた線形モデルの知識がいくつか必要となる（表15.1、p.443の3列目を参照）。また、モデルのリンク関数に関する知識は、表15.2(p.450)の1行目を参照してほしい。そして、本章で扱う被予測変数と予測変数の組み合わせが、他の組み合わせとどのような関係にあるかについては、表15.3(p.453)を参照してほしい。

17.1. 単純な線形回帰

図17.1にあるのは、単純な線形回帰モデルをもとに生成したサンプルデータである。まず x を任意に生成する。その x について、予測される平均値 y を $\mu = \beta_0 + \beta_1 x$ で計算する。それから、データ y についての確率変数を、平均 μ 、標準偏差 σ の正規分布から生成する。切片 (β_0) と傾き (β_1) の解釈については、図15.1(p.435)を見てほしい。

具体的な話の前に、今回取り扱う線形モデルによって明らかになるのは y が x に依存する程度だけだ、という点は強調しておきたい。言い換えれば、このモデルを用いたところで x の性質や確率分布について何かがわかるということはない、ということである。事実、図17.1の左側に示された x の値は一様分布から、右側の x は二峰分布から生成したものだが、背景となっている線形モデルは同一となっている。これは、 x を生成した確率分布の違いをモデルで区別できていないということを意味している。

また、このモデルでは、 x の各値における y の値の分散が同一であるという分散の等質性（等分散性）が仮定されている。この仮定は、図17.1の左側に容易に見てとることができる。しかし、等分散性の仮定をこのように視認できるのは、 x が一様に分布している場合だけであり、実際に x が一様分布でない場合は、等分散性を確認するのは簡単ではない。その一例が図17.1の右側である。このデータを見る限り、 x の各値における y の分散は等質とは思えない。 $x = 2.5$ における y のばらつきは、 $x = 7.5$ のときよりも大きいように見えるだろう。しかし、実は左右の図のシミュレーションデータは、生成する段階で y に同一の分散（同一の標準偏差 σ ）を設定していた。したがって、見た目にはわかりにくいが、実は右図のデータにも等分散性の仮定が成立しているのである。見た目にわかりにくい理由として、データ数が多いポイントでは正規分布の裾のほうから y の外れた

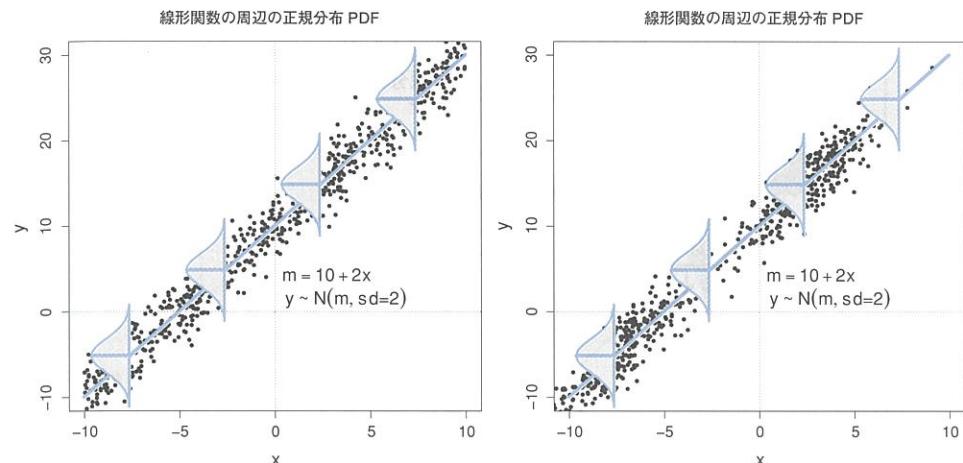


図17.1 サンプルデータと回帰直線（左側は図15.9、p.451と同じもの）。左側では、 x の各値における y の値に正規性と分散の等質性が仮定されている。なお、変数 x の分布については制約をおいていない（一様分布）。右側は変数 x の分布が二峰性の場合を示す。なお、 y に仮定された分散の値は左右で同一となっている。

値が生成される確率が高くなり、その一方でデータ数が少ないポイントでは正規分布の密度の高い平均付近から y が生成されやすいといふことがあげられる。

次に、現実世界から抽出したデータへモデルを適用するときのことを考えてみよう。 x や y の値は、現実世界の中で因果関係を持っている場合もある。例えば、 x が y の原因である場合や、その逆。さらには第三の変数が x と y の原因であって、見かけ上で x が y の原因に見える場合もあるかもしれない。単純な線形回帰モデルは、こういった x と y の因果関係についてどのような示唆も与えてはくれない。ただ単に、 x によって y がどの程度「予測できるか」という x と y の線形関係を記述するだけなのである。さて、この回帰モデルでデータを記述しようとするときは、未知なる過程から生まれたデータの散布図を書くことから始める。その上でパラメータを推定し、データ数の少ないポイントについても示唆をしてくれるモデル予測値を求めていくのが手順である。このとき、モデルがデータをよく反映していても（また、反映していないとも）、その数学的な計算過程が因果関係などの情報を示すことはない。しかし、このような記述的モデルのパラメータには、「傾向を示してくれる」という点において、確かな意義があるといえる。

17.2. ロバスト線形回帰

数学的に扱いやすいという理由から、線形モデルを用いた分析には伝統的に正規分布が仮定してきた。しかし、ノイズを含んだデータに対してまで正規分布を使う必然性はない。実際、私たちが扱うデータには、外れ値が含まれていることが多いだろう。そういう場合に対応するためには、正規分布の代わりに裾の重たい別のノイズ分布を利用するという手段をとることができる。これは、現代のソフトウェアであれば簡単に実施する

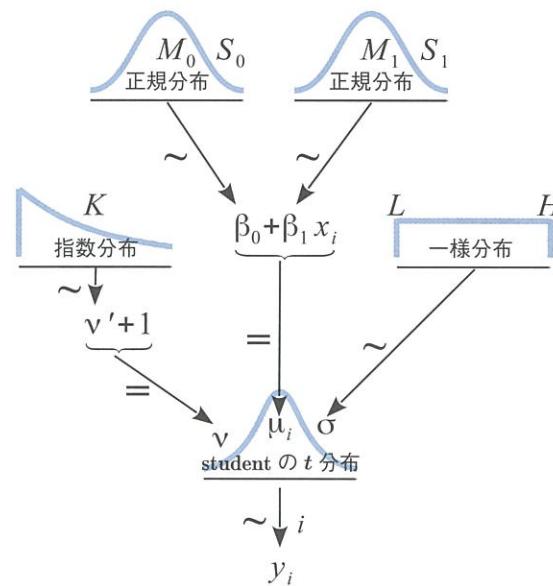


図 17.2 ロバスト線形回帰モデルのパラメータ間の依存関係。ダイアグラムの最下部にあるデータ y_i は、 x_i の線形関数である μ_i の周囲に分布している。図 16.11(p. 477) と比較してほしい。

ことが可能であり、セクション 16.2 (p. 466) では、その一例として、ノイズのあるデータに t 分布を使うことの利便性について紹介している。

図 17.2 は、モデルパラメータに仮定された事前分布と、パラメータ間の階層的な依存関係を示したものである。ダイアグラムの一番下にある y_i は $\mu_i = \beta_0 + \beta_1 x_i$ を中心とする t 分布に従うことが仮定されている。そして、 t 分布のスケールパラメータである σ には無情報の一様分布、正規化パラメータである ν には幅の広い指数分布、そして切片と傾きには幅の広い正規分布が事前分布として与えられている。図 17.2 は、p. 477 の図 16.11 で示した群の中心傾向についてのロバスト推定ダイアグラムに似ている。ただし、図 16.11 では 2 つのパラメータ β_0 と β_1 の正規分布による事前分布は重ね合わせたものになっていた。

例として、ここではランダムに選抜された人々の身長（インチ）と体重（ポンド）のデータが手元にあるとしよう。図 17.3 および図 17.4 は、それぞれ $N = 30$ と $N = 300$ のデータセットを表している。まず、2 つの図の上段にある散布図に注目していただきたい（直線と分布については後ほど説明する）。グラフの軸はそれぞれ身長と体重。各データ点はある個人の身長と体重の組み合わせを示したものである。なお、これらのデータは実際の母集団パラメータを使用し (Brainard & Burmaster, 1992), プログラムによって生成したものだ (`HtWtDataGenerator.R`)。このシミュレーションデータは、実際のところ 3 つの 2 变量正規分布クラスターに基づくものであって、単一の変数の線形従属ではない。しかし、ここではあえてそのデータに線形回帰を適用してみる。では、そのデータを確認しよう。図 17.3 および図 17.4 の散布図を見れば、身長が高くなるほど体重も増加するという傾向があるようだ。

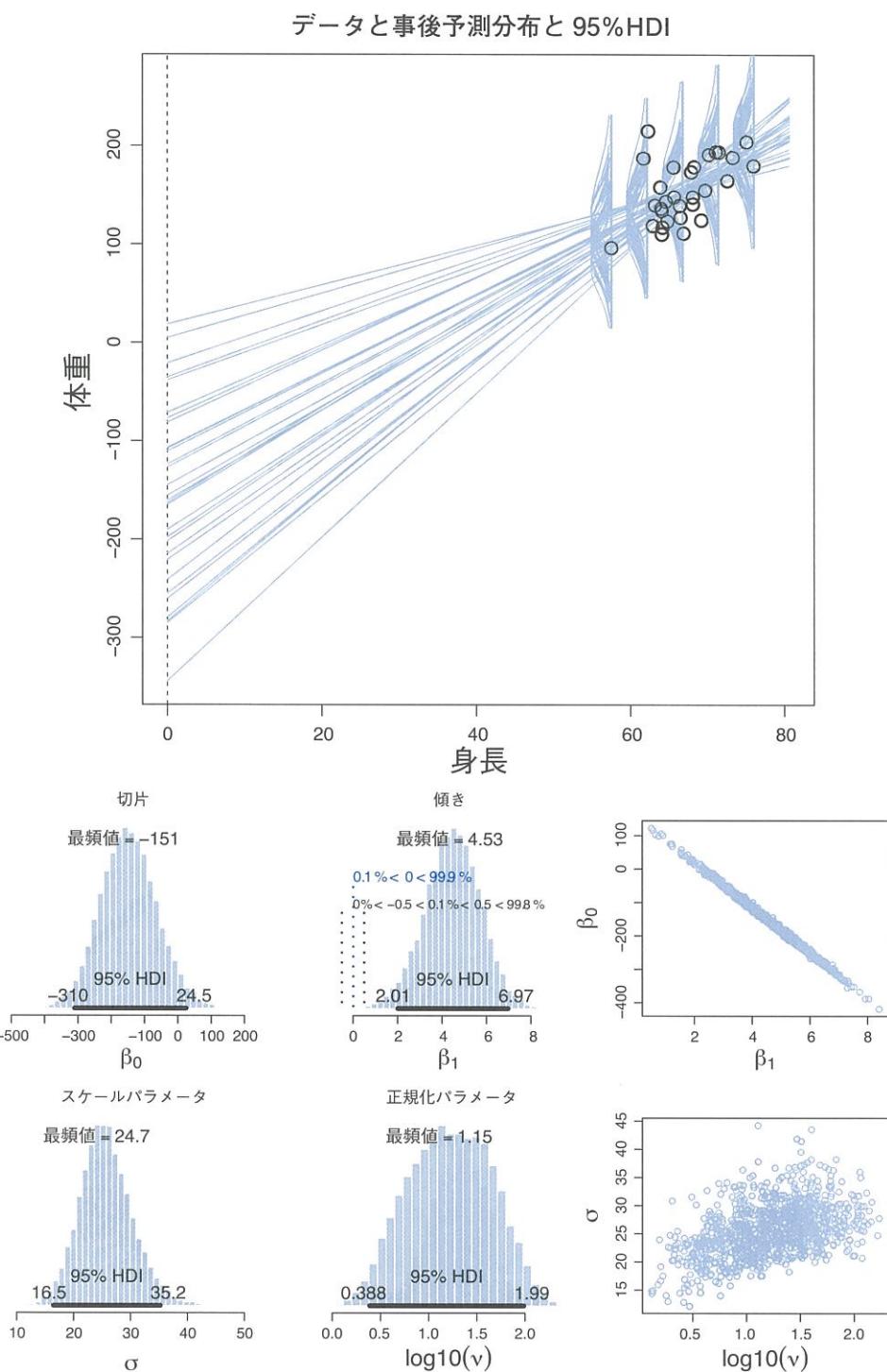


図 17.3 上段：推定区間内にある回帰直線の例 ($N = 30$) とノイズを含む t 分布。下段：パラメータの周辺事後分布。図 17.4 と比較しながら見てほしい。

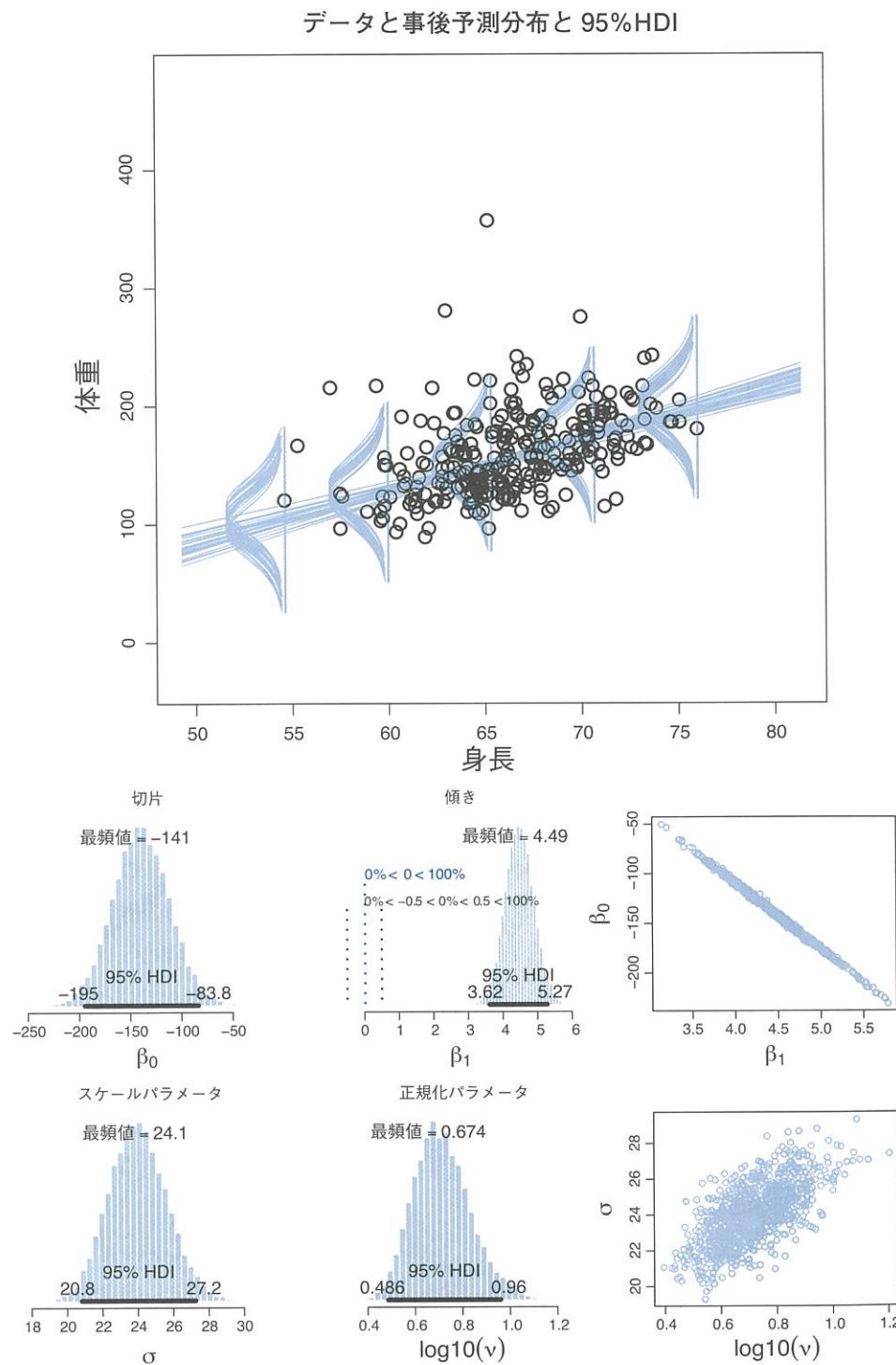


図 17.4 上段：推定区間にある回帰直線の例 ($N = 300$) とノイズを含む t 分布。下段：パラメータの周辺事後分布。図 17.3 と比較しながら見てほしい。

さて、私たちの目標は、データから線形モデルに含まれる β_0 , β_1 , σ , そして v の最適な組み合わせを見つけることになる。そして、その答えは下記のようにしてベイズの公式によって導くことができる。

$$p(\beta_0, \beta_1, \sigma, v | D) = \frac{p(D|\beta_0, \beta_1, \sigma, v) p(\beta_0, \beta_1, \sigma, v)}{\int \int \int d\beta_0 d\beta_1 d\sigma dv p(D|\beta_0, \beta_1, \sigma, v) p(\beta_0, \beta_1, \sigma, v)}$$

難解な数式に思われるかもしれないが、心配は無用だ。私たちがすべき仕事は2つであって、上記の数式に基づく分析は JAGS か Stan が実行してくれる。私たちがすべき仕事、その1つは適切な事前分布を設定することである。そして2つ目は、MCMC の各チェインが十分にミックスされ収束しているかを見ることによって、信頼できる事後分布が得られたかどうかを確認することである。

17.2.1. JAGS によるロバスト線形回帰

これまでの章で強調してきたように（例えば、セクション 8.2）、図 17.2 のような階層的ダイアグラムは、モデルの概念化だけでなくコードを書き上げるときにも有用である。図 17.2 のすべての矢印は、JAGS や Stan のコードにきれいに対応している。この下のセクションでは、JAGS で分析用コードを実装していくが、コード化のポイントはダイアグラムの矢印をそのまま転記すればよいというところにある。図 17.2 と以下の JAGS コードを見ていただければ、7つの矢印がコードの対応する線になっていることがおわかりいただけるだろう。

```
model {
  for ( i in 1:Ntotal ) {
    zy[i] ~ dt( mu[i] , 1/zsigma^2 , nu )
    mu[i] <- zbeta0 + zbeta1 * zx[i]
  }
  zbeta0 ~ dnorm( 0 , 1/(10)^2 )
  zbeta1 ~ dnorm( 0 , 1/(10)^2 )
  zsigma ~ dunif( 1.0E-3 , 1.0E+3 )
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29.0)
}
```

コードに記載されたパラメータの中には、名前に “z” とついているものがある。これは、標準化されたデータが “z” によって表現されることがよくあるからだ。今回もこれに準拠し、標準化されたデータをもとに推定されるパラメータには “z” を付すことにする（標準化の詳細については後述する）。

JAGS（と Stan）には、分布の引数を関数で評価できるという優れた特徴があるので、

変数名の代わりに引数において操作する書き方ができる。例えば、上述の t 分布（コード上では `dt`）では、変数名の代わりに `1/zsigma^2` という式を精度の引数とする。同じことを `dt` の他の引数でもできる。特に、行を変えて `mu[i]` を評価する代わりに、`dt` の平均の引数 `mu[i]` のところに `zbeta0 + zbeta1 * zx[i]` と書くだけで良い。つまり、図 17.2 のダイアグラムにおける引数の矢印は、JAGS コードの中では違う行に書く必要がない。とはいっても、引数に直接数式表現をおくときは、JAGS はチェインの中でのその式の値を記録していくことができない。つまり、前述した書き方をすると、`mu[i]` が各ステップのパラメータ値として記録し続けることが可能になる。各ステップの `mu[i]` には興味がないので、コードを書き直すとき（後で示す）は、`dt` の平均を表す引数に直接数式を書き込む。一方、`nu` の数式は、個別の行のままにしておく（上で示したように）。というのもそれは、その値の記録をしっかり見たいからである。

17.2.1.1. MCMC サンプリングのためにデータを標準化する

上記の JAGS コードは、原理的には標準化していないローデータでも実行可能である。しかし、ローデータのままとなると実践的な問題が生じてしまうことは少なくない。もちろん、コードに数学的・論理的な誤りがあるわけではない。問題というのは、特に切片と傾きのパラメータ間に強い相関関係が認められる場合に生じてくる。そういう場合、通常のサンプリングアルゴリズムで探索することが困難となり、極端に非効率的な分析を行ってしまうことになる。

図 17.3 の上部を見てもらえば、なぜそのようなことが起こるのかが直観的にわかる。図示された複数の回帰直線は、MCMC で求めた事後分布を用いて引かれたものだ。すべての直線はデータの中心を通っているが、切片と傾きには不確実性が存在している。ご覧の通り、傾きが急な場合は切片（すなわち $x = 0$ の回帰直線の y の値）の値が大きな負の値をとっていることがわかる。その一方で、傾きが緩やかな場合は、切片はさほど小さい値ではない。要するに、切片と傾きがトレードオフの関係にあるのである。図 17.3 の中央段の右は、傾き β_1 と切片 β_0 の同時分布を示している。これを見れば、切片と傾きが非常に強い相関関係を持っていることがおわかりいただけるであろう。

こういったタイトな相関関係にある分布から MCMC サンプリングを行うことは難しく、ギブスサンプリングの場合には、「分布の壁にぶつかって抜け出せなくなる」という現象が生じてしまう。ここで、ギブスサンプリングの仕組みについて今一度復習しておきたい。ギブスサンプリングでは、他のパラメータの値を固定した上で、ある 1 つのパラメータの値が探索される。本例の場合だと、 β_0 の値を変化させるときには、 β_1 の値は固定されているという具合である。そのため、パラメータ間に強い相関関係があると、 β_1 を固定したときに β_0 のサンプリングが極めて狭い範囲でしか移動できなくなってしまう。これは β_1 と β_0 を逆転させた場合も同様だ。そのため、2 つのパラメータ値の探索は緩慢にしか生じず、MCMC チェインに高い自己相関が生じるという問題が生じる。なお、動

きが緩慢になりがちな他のサンプリングアルゴリズムについても、これと同様の問題を抱えることになる。以上のこととは、原理としては問題ないが、事後分布からの適切なサンプルを得るまでに非常に長い時間がかかるということを意味している。

サンプリングを効率的に行うには、(少なくとも) 2 つの方法が提案されている。1 つは、サンプリングアルゴリズム自体を変更するというものである。その例として、(JAGS がメインで取り扱う) ギブスサンプリングの変わりに、Stan に実装されたハミルトニアンモンテカルロ法 (HMC) を使うという手がそれに該当する。もう 1 つの方法は、切片と傾きの強い相関関係に悩まされないようデータ自体を変形してしまうというものである。ここからは、この 2 つ目の方法を説明していこう。

では、改めて図 17.3 のデータと回帰直線を見てもらいたい。切片と傾きに相関関係があるという問題は、 x 軸の 0 の点がデータから非常に遠いところにあることに由来している。そのため、もしもこの 0 の点をデータの平均値の直下にずらすことができれば、 x が 0 のときの y の値（切片）が大きく変化することなく傾きの値を探索することができると言えられる。 0 の点を平均の直下にずらすという方法は、データから平均を引き算することで簡単に実現できる。データから平均を減算する方法は「平均中心化 (mean centering)」と呼ばれている。これだけで、パラメータ間の相関によって生じる問題を改善することができるのだ。

しかし、 x の値を平均中心化するのであれば、データの標準化を行ったほうがよいだろう。標準化とは、シンプルにいえば平均と標準偏差を用いてデータの単位をそろえるという作業である。これにより、データがどのようなスケール（単位）であってもパラメータに漠然とした事前分布を仮定することができる。

$$z_x = \frac{(x - M_x)}{SD_x} \quad \text{かつ} \quad z_y = \frac{(y - M_y)}{SD_y} \quad (17.1)$$

ここで、 M_x は x の値の平均値であり、 SD_x はデータ x の標準偏差を示している。簡単な代数計算で導くことができるが、 z_x の平均は 0 で、標準偏差は 1 となる。平均中心化とは異なり、標準化ではデータを伸び縮みさせて標準偏差が 1 となるような計算が行われる。

標準化したデータを JAGS に渡せば、JAGS は確信できるパラメータの値を返してくれる。そうなると、次はパラメータの推定値をもとのデータのスケールに変換する必要がある（身長が 1 インチ増えれば、体重が何ポンド増加するかという単位で考えることができるため）。標準化データを用いた切片と傾きは、それぞれ ζ_0 , ζ_1 と表し（ギリシャ文字で「ゼータ」と読む）、 y の予測値は \hat{y} と表記される。その上で次のようにする。

$$\begin{aligned} z_{\hat{y}} &= \zeta_0 + \zeta_1 z_x && \text{モデルの定義より} \\ \frac{(\hat{y} - M_y)}{SD_y} &= \zeta_0 + \zeta_1 \frac{(x - M_x)}{SD_x} && \text{式 17.1 より} \\ \hat{y} &= \underbrace{\zeta_0 SD_y + M_y}_{\beta_0} - \underbrace{\zeta_1 M_x SD_y / SD_x + \zeta_1 SD_y / SD_x x}_{\beta_1} && (17.2) \end{aligned}$$

式 17.2 が示す通り、 ζ_0 と ζ_1 が含まれる各項は β_0 と β_1 の値に対応している。また、標準化されたデータで推定した σ をもとのスケールに戻す場合は、 σ に SD_y を乗算することでそれが可能となる。一方、正規化パラメータについては変換する必要がない。その理由は、正規化パラメータが分布の相対的な形状を示すものであって、スケールパラメータに影響を受けないからである²⁾。

データの標準化には、実際的な意味でいくつかのやり方がある。例えば、R でデータを標準化して JAGS で推定した後、再度 R でスケールを変換するという方法が1つだ。しかしここでは、オリジナルデータを JAGS に渡して JAGS 内でデータの標準化とパラメータ推定し、さらにはオリジナルスケールへの変換までを行うという方法をとりたい。オリジナルスケールのデータと標準化データを用いた場合の両方の推定過程を JAGS で記録できるということや、それらの収束診断も容易に行うことができるというメリットがあるためである。また、この方法を用いることで、これまで本書で扱っていない JAGS の data ブロックの使い方を紹介することもできる。

なお、JAGS に渡すデータは、標準化されていないベクトル形式の x と y のみである。その他の情報は JAGS には渡していない。JAGS のモデル記述としては、データの標準化は data ブロックにおいて、次のようにする。

```
data {
  Ntotal <- length(y)
  xm <- mean(x)
  ym <- mean(y)
  xsd <- sd(x)
  ysd <- sd(y)
  for ( i in 1:length(y) ) {
    zx[i] <- ( x[i] - xm ) / xsd
    zy[i] <- ( y[i] - ym ) / ysd
  }
}
```

2) 正規化パラメータ v がデータの標準化で変化しないという事実は、 t 分布の式からきているものである。標準化された x の値は式の核から分離されており、標準化以外の役割を担っていないことを示している：

$$p(x|v, \mu, \sigma) = \frac{\Gamma(\frac{v+1}{2})}{\Gamma(\frac{v}{2}) \sqrt{\pi v} \sigma} \left(1 + \frac{1}{v} \left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{v+1}{2}}$$

data ブロックで計算されたすべての値は、model ブロックで用いることができる。ここで、標準化されたデータとそのパラメータについては、変数名に z をつけておく。データが標準化されていると、傾きの値が ± 1 から大きく外れることはない。そのため、標準偏差に 10 を与えれば、傾きがとりうる値の範囲に対して十分にフラットな事前分布を仮定することができる。このような考え方は、切片やノイズについても同様に成り立つ。したがって、データがどのようなもとのスケールであっても、標準化すればこれらのパラメータに漠然とした事前分布を設定することができるようになる。

```
model {
  for ( i in 1:Ntotal ) {
    zy[i] ~ dt( zbeta0 + zbeta1 * zx[i] , 1/zsigma^2 , nu )
  }
  # 標準化スケールでの漠然とした事前分布の設定：
  zbeta0 ~ dnorm( 0 , 1/10^2 )
  zbeta1 ~ dnorm( 0 , 1/10^2 )
  zsigma ~ dunif( 1.0E-3 , 1.0E+3 )
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29.0)
  # オリジナルスケールへの変換：
  beta1 <- zbeta1 * ysd / xsd
  beta0 <- zbeta0 * ysd + ym - zbeta1 * xm * ysd / xsd
  sigma <- zsigma * ysd
}
```

上記 JAGS コードの model ブロックの最後では、パラメータをオリジナルのスケールに変換する手続きとして、式 17.2 にある直接代入方式を使ったものが記述されている。

なお、上の高水準な JAGS コードは `Jags-Ymet-Xmet-Mrobust-Example.R` と名付けられた R ファイルにある。また、そこで用いられている関数は `Jags-Ymet-Xmet-Mrobust.R` で定義されている。これらのスクリプトを実行すれば、図 17.3 および図 17.4 のグラフも出力することができる。さて、分析結果について話をする前に、Stan によるモデルの実装方法も紹介しておこう。

17.2.2. Stan によるロバスト線形回帰

パラメータ空間で値を提案するために、Stan がハミルトン力学を用いていることを紹介したセクション 14.1 (p. 408) を思い出していただければと思う。ハミルトン力学に基づくサンプリングの軌道は、事後分布の勾配を用いることで狭い分布からでも大きく動くことができるという特徴があった。それゆえ、HMC を用いればデータの標準化を行わずとも効率的に事後分布からのサンプリングを実施することができる。

17.2.2.1. 漠然とした事前分布の定数

Stanによる実装を紹介する本項では、事前分布の設定について新たな方法を紹介する。Stanに渡すデータは、今回標準化を行わない。そのため、先述のJAGSモデルを使った事前分布の定数がStanコードに援用できなくなってしまう。そこで、データのスケールがどのような大きさであってもそれを捉えることができる、パラメータの事前分布を設定する新たな方法が必要になる。なお、ユーザーに事前分布の設定をゆだねることもできる。しかしここでは、十分に幅の広い事前分布をデータにあわせて設定することを試みたい（データにくらべて幅の広い事前分布を設定する方法は前章で紹介した）。

回帰直線の傾きは、データに完璧な相関関係がある場合、 SD_y/SD_x がその最大値となる。また、切片の最大値はデータに完全な相関があるときの $M_x SD_y/SD_x$ となる。したがって、傾きや切片の事前分布の標準偏差には、これらの最大値以上の値を設定する必要がある。次のStanモデルでは、上記のアイデアを用いて切片や傾きの事前分布を設定している。

```
data {
  int<lower=1> Ntotal ;
  real x[Ntotal] ;
  real y[Ntotal] ;
  real meanY ;
  real sdY ;
  real meanX ;
  real sdX ;
}
transformed data { //事前分布に与える定数
  real unifLo ;
  real unifHi ;
  real expLambda ;
  real beta0sigma ;
  real beta1sigma ;
  unifLo <- sdY/1000 ;
  unifHi <- sdY*1000 ;
  expLambda <- 1/29.0 ;
  beta1sigma <- 10*fabs(sdY/sdX) ; // fabsは絶対値変換を示す
  beta0sigma <- 10*fabs(meanX*sdY/sdX) ;
}
parameters {
  real beta0 ;
  real beta1 ;
  real<lower=0> nuMinusOne ;
  real<lower=0> sigma ;
}
```

```
transformed parameters {
  real<lower=0> nu ; //実際の最小値は1
  nu <- nuMinusOne + 1 ;
}
model {
  sigma ~ uniform( unifLo , unifHi ) ;
  nuMinusOne ~ exponential( expLambda ) ;
  beta0 ~ normal( 0 , beta0sigma ) ;
  beta1 ~ normal( 0 , beta1sigma ) ;
  for ( i in 1:Ntotal ) {
    y[i] ~ student_t( nu , beta0 + beta1 * x[i] , sigma ) ;
  }
}
```

上記Stanモデルのスクリプトは Stan-Ymet-Xmet-Mrobust-Example.R にある。その中で用いられている関数は Stan-Ymet-Xmet-Mrobust.R で定義されている。これらのコードを実行すると、図17.3および図17.4と同様のグラフが出力される。分析の結果について議論する前に、StanとJAGSのパフォーマンスの違いについて簡単に比較しておきたい。

17.2.3. StanかJAGSか

図17.4のデータを用いた分析をJAGSとStanで行った場合、事後分布の詳細に関する両者の推定結果はほとんど変わらない。比較のために、JAGSとStanの設定を順応ステップ500、バーンインを1000、間引きなしのチェイン系列を4本とした総サンプル20,000と設定した。高いスペックではない私のデスクトップ型パソコンで分析を行った結果、JAGSはおよそ180秒、Stanでは485秒（JAGSの2.7倍ほどの時間）で推定が終了した。JAGSで得られた有効サンプルサイズ（ESS）は、パラメータごとに大きく異なり、 ν と σ で約5000、 β_0 と β_1 で約16,000となった。Stanの場合、パラメータ間でESSに大きな違いがないものの、ESSはJAGSほど高くはなく ν と σ で約8000、 β_0 と β_1 で約7000程度であった。

結果として、パラメータ間に相関がある場合、StanはJAGSにくらべて分析に長い時間が必要となった。しかし、Stanの場合は標準化作業が不要で、正規化パラメータやスケールパラメータの自己相関が小さくなるという利点がある（p.476の図16.10を参照）。このように、HMCは他のサンプリングアルゴリズムが抱える問題を回避した、巧みな事後分布の推定を可能してくれる。ただし、Stanを用いた場合は切片と傾きという関心のあるパラメータの有効サンプルサイズが小さくなり、時間もかかってしまうことも事実である。JAGSで行ったようにデータを標準化することで、Stanでもこのモデルが効率的に改良されるだろう。しかし、これら一連のデモンストレーションは、あくまでHMC

がサンプリングの問題をクリアしながら事後分布の推定を行うことができるということを示すためである。また、Stan の（そして JAGS の）効率性は、 t 分布を再パラメータ化することでも改善できると Stan のリファレンスマニュアルに解説されている。なお、これ以降のプログラムコードは JAGS に焦点を当てていくが、それらはそのまま Stan にも援用できるため、その点ご理解いただければと思う。

17.2.4. 事後分布の解釈

ここまで、分析コードの詳細について見てきた。ここからは図 17.3 および図 17.4 に示された分析の結果について考えていく。思い出してほしいのだが、2つの図に示されたデータは線形モデルを基礎とする同様の手続きで生成されたものであった。なお、図 17.3 はデータ点が 30 であり、一方の図 17.4 のデータ点は 300 だった。周辺事後分布を比較すれば、傾き、切片、そしてスケールパラメータの最頻値がほぼ同じであることがわかる。しかし、推定の確からしさは、データ点が 300 ある場合のほうが 30 ポイントの場合よりも優れており、狭い範囲に収まっていることが確認できる。例えば、 $N = 30$ のデータで傾きを推定した場合の最頻値は 4.5（ポンド/インチ）であり、95% HDI（最高密度区間）は約 2.0 から 7.0 までとなっている。その一方で、 $N = 300$ の場合になると、傾きの最頻値は同じ 4.5 だが、95% HDI はより狭く約 3.6 から 5.2 までの範囲にとどまっている。

興味深いのは、正規化パラメータの推定値が 2 つのデータセットで異なっている点だ。通常、データサイズが大きくなればなるほど正規化パラメータの推定値は徐々に小さくなっていき、 $N = 30$ のところで公平な領域におかれた事前分布を乗り越える（ ν の事前分布に指数分布を仮定したセクション 16.2.1, p. 470 を参照）。このような現象が起こるのは、データセットのサイズが大きいほど外れ値が出現する機会が多くなるが、それは大きな ν の値と矛盾するということに起因している（もちろん、データサイズが大きくなってしまって、それが純粋な正規分布をとるならば正規化パラメータの大きな値がより強められるだけである）。

また、応用的な話かもしれないが、手元のデータ x を使って関心のある変数の具体的な値を予測したいという場合もあるだろう。例えば、身長 50 インチの人の体重が、実際何ポンドくらいなのかを予測したいという場面を考えてみたい。ベイジアン分析の特長の 1 つは、確信できる予測値を点推定ではなく分布の全体像で捉えることができるということにある。推定値の分布を得るために、MCMC チェインの各ステップでは、あるステップで得られたパラメータの値から、次のランダムな値を生成するという作業が行われていく。こういった作業の結果は、図 17.4 に垂直にプロットされたノイズ分布のようにして具現化する。図では、あらゆる身長の値における体重の予測値がノイズ分布によって示されている。予測分布の興味深い側面は、分布が自然に広がっているところであり、データから遠く離れたところほど不確かなものとなっている。このような現象は、図 17.3 に

容易に見てとることができる。データから離れたところを探ろうとするほど、回帰直線間の幅が拡大してゆき、予測値が不確かなものとなっていく。

図 17.3 と図 17.4 の上段には、推定区間内の回帰直線と t 分布が重ねて表示されている。これは視覚的な事後予測チェックそのものであり、このように図示することでモデルとデータ間の系統的な誤差を視認することができる。こういった視覚化を行えば、データの非線形な傾向や体重の分布の対称性を確認することができるかもしれない。また、体重の分布が回帰直線と比較して正の方向に偏っていることなどが示唆されるかもしれない。もしこのような系統的誤差が理論的に無視できない場合や、パラメータの解釈に強く支障をきたす場合には、非対称性を仮定したノイズ分布をモデルに組み込む必要が出てくることがあるだろう。事後予測チェックに関する詳細な情報については、Kruschke (2013b) を参照してほしい。なお、ロバスト線形回帰モデルを適用した今回のケースでは、3 つの 2 変量正規分布から生成したデータをとてもうまく近似することになった。しかし、今回のモデルは y が x に依存する程度を表しただけであって、 x の分布や性質についてモデルは何も記述していない。

17.3. 個人と集団の階層回帰モデル

ここまで扱ってきたのは、ある j 番目の個人が x_j, y_j の 1 つの組しかデータとして持っていないケースだった。しかし、個人ごとに複数の観測データ (x_{ij}, y_{ij}) がある場合も存在するだろう ($i|j$ という表記は、ある j 番目の個人が i 個の観測データを持っているということを意味している)。このようなデータがあれば、私たちは個人ごとの傾向を求めることができる。そして、各個人が集団の代表性を持っていると考えられる場合、個人データから集団レベルのパラメータを推定することも可能となる。

ある子どもの集団に対して、読みの能力を複数年にわたって縦断的に測定したという Walker, Gustafson, and Frimer (2007) の例について考えてみよう。それぞれの子どもから得られるデータは、測定回数分の読みの能力とそのときの年齢である。このようなデータを使えば、年齢が上がるにつれて読みの能力がどの程度向上するかという回帰直線を個人ごとに求めることができる。そして、モデルに上位の分布を設定すれば、集団全体レベルでの切片と傾きを推定することも可能だ。このとき、個人ごとの推定値は上位の分布を設定することで他の個人と相互に情報を共有することになる。なお、この例の場合、私たちは個人の読みの能力と年齢による成長率に加え、集団レベルでの読みの能力とその成長率にも関心を持つかもしれない。

もうひとつの例として、世帯人数によって世帯収入がどの程度変化するか、地域別に検討したいという場合も考えておこう。アメリカ合衆国の国税調査局は、米国 50 州とワシントン D.C.、そしてプエルトリコの世帯収入の中央値と世帯人数のデータを公開している。この場合、特定の地域が「ある個人」に相当し、地域ごとの世帯収入の中央値なら

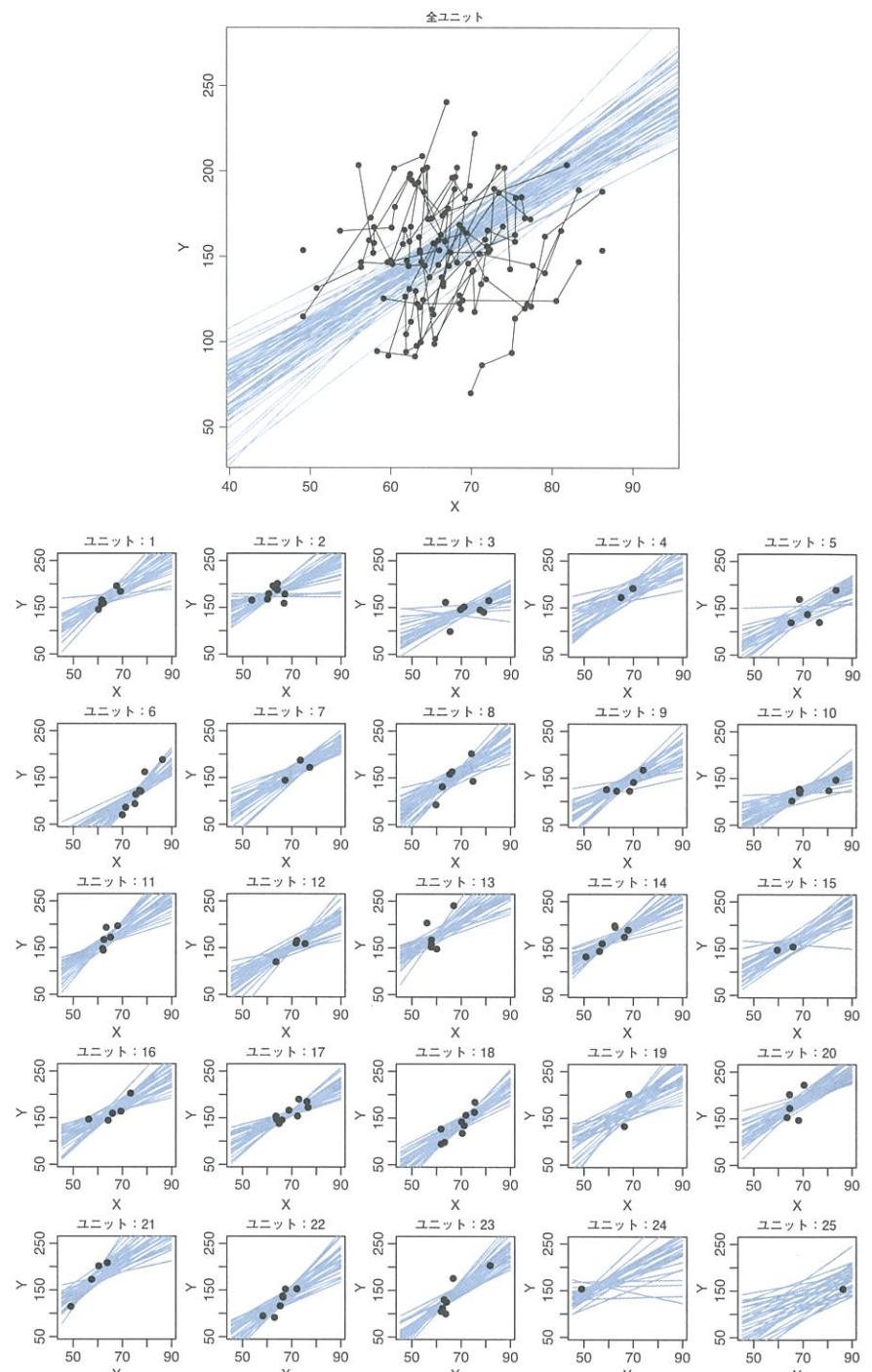


図 17.5 事後予測線を重ねて表記した階層線形回帰のデモンストレーション用架空データ。上：個人ごとのデータが実線でつなげて表示され、すべての個人データがプロットされている。下：個人ごとのデータプロット。最後の 2 つは 1 名のデータしか得られていないが、傾きと切片は階層モデルによって他と同等に推定されている。

びに世帯人数（2人以下や3人以下、6、7人までなど）が「ある個人」に含まれる観測データとして得られることになる。もし、中央値ではなく実際の収入がわかっているならば、その値を用いることもできる。こういったデータの場合も、個々の地域と全体の傾向の両方を推定することが私たちの関心となる。なおここでも、推定された個々の地域の傾向はすべての地域の情報を含む上位の分布を通じて間接的に他の地域と情報を共有することになる。

図 17.5 は、作業用の架空のデータである。 x と y は、前のセクションの身長と体重の例とおよそ同様のものと考えていただいてかまわない。しかし、これらは具体的な調査などのデータであることを想定しておらず、個人に複数の x, y の組み合わせが与えられたものとなっている。また、このデータは個人の成長曲線のように見ることができると、あくまで架空のデータであることは強調しておく。図 17.5 の上部は、すべての個人データが重なって表示されたものであり、個人が持つ複数のデータはすべて実線でつなげられている。図 17.5 の下部は、個人別のデータプロットである。ここで押さえておきたいのは、個人によってデータ点の数が違うということ、そして、 x の値が個人によって全然違う場合があるという点だ。

私たちの目標は、個人ごとの回帰直線を求めるときに同時に、集団レベルの傾きと切片を推定することにある。このとき鍵となるのは、各個人が所属する集団の代表性を持つという前提である。言い換えれば、各個人が同じ集団としての特徴を持っているという仮定があるからこそ、個人の情報を集団レベルの傾きと切片の推定に用いることができるようになる。このような方法によって、個人間での情報共有がなされ、個人パラメータの推定値が集団における最頻値に向かって縮小していく。縮小に関する話は、もう少しモデルに馴染んだ頃に改めて紹介することとしよう。

17.3.1. JAGS によるモデルの実装方法

図 17.6 は今回のモデルのダイアグラムである。ダイアグラムを見ると少し圧倒されるかもしれないが、p. 488 の図 17.2 と比較していただければ、すでに馴染みのあるモデルに 1 つの階層を足しただけということがおわかりいただけるだろう。ダイアグラムはモデル内のパラメータの依存関係を理解しやすく、そのまま JAGS や Stan のコードに使えるという利便性がある（ダイアグラムの矢印とコードが一致するようになっている）。

図 17.6 の最下部では、 j 番目の個人が持つ i 番目のデータが t 分布に従うことが示されている。このとき、 t 分布の平均 μ_{ij} は、 $\mu_{ij} = \beta_{0,j} + \beta_{1,j}x_{ij}$ として定義されている。線形モデルの切片と傾きには、 j が添え字として付されている。これは、すべての参加者に切片と傾きがあることを意味するためのものだ。階層ダイアグラムの上部では、個人ごとの傾きが平均を μ_1 とし、 σ_1 を標準偏差とする上位階層の正規分布に従っていること

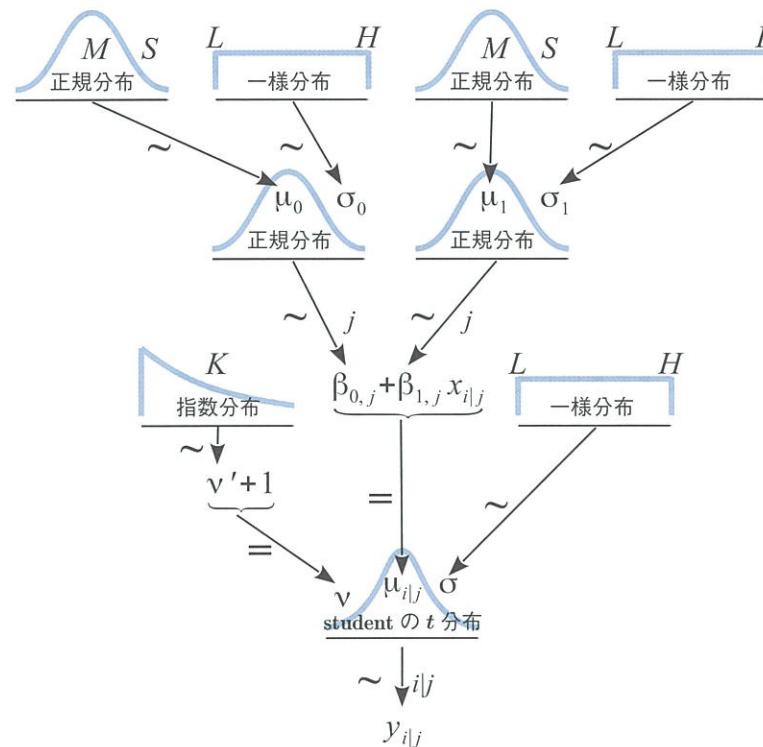


図 17.6 階層ロバスト線形回帰モデルにおけるパラメータ間の依存関係。図 17.2 (p. 488) と比較してほしい。

が表現されている（これは、 $\beta_{1,j} \sim \text{normal}(\mu_1, \sigma_1)$ と表記することもできる）。ここで、 μ_1 は個人ごとの傾きを代表するものであり、 σ_1 は傾きの個人間のばらつきを示すものとなる。したがって、もしすべての個人がほぼ同じ傾きを有しているならば σ_1 は小さな値で推定されることになる。反対に、個人ごとに傾きの値が大きく異なる場合は、 σ_1 の推定値も大きくなる。なお、これは切片についても同様である。また、階層の最上位で与えられた集団レベルのパラメータの事前分布には、漠然とした事前分布を仮定している。

このモデルでは、個人のノイズを示す標準偏差がどの個人についても同じであるという仮定をとっている。図 17.6 を見ると、パラメータ σ が全体で 1 つしか設定されていないことがわかる。このような仮定は、個人ごとにノイズパラメータを与え、その上に上位の階層パラメータを設定するという方法で緩和することも可能だ。個人ごとにノイズパラメータを設定し、集団レベルでそれらをプールする方法については、図 19.6 (p. 583) にて解説する。

モデルの JAGS コードを理解するためには、今回使用するデータの形式を理解しておく必要がある。データのファイルは 3 列からなる。各行がデータの 1 つの点を表していて、 x, y がポイントの値、そしてその値を提供した参加者を表す数字からなる。データファイルの i 行目における x の値は $x[i]$ 、 y の値は $y[i]$ で特定される。なお、参加者番号は $s[i]$ である。ここで、 i 番目という表記は j 番目の参加者の i 番目のデータという意

味ではなく、単純にデータセットの i 行目を意味しているということに注意していただきたい。今回使用する JAGS モデルは、参加者番号が連続した整数值で与えられていることを前提としているため、複数回登場する参加者番号と、それに対応するデータはプログラム内で同一参加者のものとして統合される仕様になっている。したがって、参加者番号は複数個あるが、切片と傾きは個人ごとに推定される仕組みとなっている。

JAGS コードの冒頭では、前のセクションで示した、非階層的なモデルと同じ方法でデータの標準化が行われる。

```
data {
  Ntotal <- length(y)
  xm <- mean(x)
  ym <- mean(y)
  xsd <- sd(x)
  ysd <- sd(y)
  for ( i in 1:length(y) ) {
    zx[i] <- ( x[i] - xm ) / xsd
    zy[i] <- ( y[i] - ym ) / ysd
  }
}
```

次は、階層モデルの構成に進もう。下記のモデル指定区画では、 j 人目の切片と傾きが 1 つの上位パラメータに入れ子状にコード化されている。なお、標準化した y の値は $zy[i]$ と表記するため、それを予測する傾きにも z を付して $zbeta1[s[i]]$ としておくと整理されやすい。そして、データならびにパラメータ間の関係は、図 17.6 に示したダイアグラムに対応して記述されている。

```
model {
  for ( i in 1:Ntotal ) {
    zy[i] ~ dt( zbeta0[s[i]] + zbeta1[s[i]] * zx[i], 1/zsigma^2, nu )
  }
  for ( j in 1:Nsubj ) {
    zbeta0[j] ~ dnorm( zbeta0mu, 1/(zbeta0sigma)^2 )
    zbeta1[j] ~ dnorm( zbeta1mu, 1/(zbeta1sigma)^2 )
  }
  #標準化スケールでの漠然とした事前分布の設定：
  zbeta0mu ~ dnorm( 0, 1/(10)^2 )
  zbeta1mu ~ dnorm( 0, 1/(10)^2 )
  zsigma ~ dunif( 1.0E-3, 1.0E+3 )
  zbeta0sigma ~ dunif( 1.0E-3, 1.0E+3 )
  zbeta1sigma ~ dunif( 1.0E-3, 1.0E+3 )
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29.0)
```

```
# オリジナルスケールへの変換：
for ( j in 1:Nsubj ) {
  beta1[j] <- zbeta1[j] * ysd / xsd
  beta0[j] <- zbeta0[j] * ysd + ym - zbeta1[j] * xm * ysd / xsd
}
beta1mu <- zbeta1mu * ysd / xsd
beta0mu <- zbeta0mu * ysd + ym - zbeta1mu * xm * ysd / xsd
sigma <- zsigma * ysd
}
```

17.3.2. 事後分布：縮小と予測

分析を実施するための完全版コードは Jags-Ymet-XmetSsubj-MrobustHier-Example.R にある。その中で使用されている関数は別途 Jags-Ymet-XmetSsubj-MrobustHier.R を参照してほしい。上記コードをご覧になれば、間引き間隔が 0 ではないことに気づかれるだろう。これは単に、パラメータの ESS を少なくとも 10,000 に保ちつつ、サンプリング過程を記録するファイルサイズを小さくするための対処である。推定時に保管される MCMC ファイルはパラメータの数が多くなるほど大きくなっていく。例えば、25 人の参加者で、標準化した場合とオリジナルスケールの場合のパラメータを推定すれば、107 の変数が JAGS によって記録されていくことになる。また、このモデルでは時間を節約するために runjags で指定したチェインを 3 本にとどめているが、それでも分析に数分の時間がかかる。なお、本コードを実行すると、時折だが正規化パラメータのチェインがうまく振る舞わないことがある。このような理由で分析結果に疑惑が生じる場合には、サンプリングを増やして JAGS コードを再度実行してみてほしい（あるいは Stan を用いることでも解消可能である）。

図 17.5 は、推定区間にある回帰直線を何本かまばらに重ねて表示したものである。回帰直線は全体的にデータに重なっていることがわかる。図 17.5 の上段は集団レベルの傾きと切片を示している。データプロットの実線を無視すれば、右肩上がりの傾向を見ることはできないだろう。しかし、個人ごとの傾きを統合することで、集団レベルの傾きが正の値をとることが明らかになる。

図 17.5 の下段では、推定区間内の回帰直線が個人ごとに重なって表示されている。上位の分布を通じて個人間の情報が相互に共有されるため（言い換えれば個人ごとの傾きが集団にプールされているため）、切片と傾きの推定に顕著な縮小が生じている（縮小が生じていることは、特に最後の 2 名を見れば明らかである）。この 2 名の切片と傾きは（データ点が 1 つしかないにもかかわらず）驚くほど狭い範囲で推定することができている。これと同時に注目していただきたいのは、推定された彼らの切片が、それぞれのデータの値に引っ張られているというところだ。これは、異なる個人には異なる予測が適切に

なされていることを示している。本例は、セクション 9.3 (p. 250) で紹介したものとは異なった階層モデルにおける縮小の例である。

17.4. 線形 2 次式とデータの重み付け

アメリカ合衆国国税調査局は、世帯収入と世帯人数を含む American Community Survey と Puerto Rico Community Survey の調査データを公開している³⁾。ここでは、世帯人数と世帯収入の関係と、それが地域ごとに（米国 50 の州とプエルトリコ、ワシントン D.C.）どのように異なるのか、というところに私たちの関心を定めたい。データは図 17.7 に視覚化した。これを見ると、世帯収入と世帯人数が逆 U 字の曲線関係にあることは明らかだ。こういったデータに対して前セクションのような線形回帰を適用すれば、系統的な予測の誤りが生じてしまうことは一目瞭然である。

今回のデータを捉えるために単純な線形モデルを用いるのは不適当と思われる所以、二乗項を加えた曲線のモデルへと拡張していく。数学的に曲線を描く方法はこれだけではないが、簡便かつ慣習的な方法であるため、今回は二乗項を用いたモデルを使用する。モデル式は $y = b_0 + b_1x + b_2x^2$ である。このとき、 b_2 が 0 となるならば、本モデルは直線を描くことになる。言い換えれば、この拡張モデルは単純な線形モデルを内包している。なお、 b_2 が正の値の場合、モデルの曲線は下に凸の放物線となり、反対に負の値のときは上に凸の放物線を描く。なお、世帯収入の曲率 b_2 が正確に放物線を描くかどうかはここでは問題ではなく、ポイントは線形 2 次式を用いることで単純な線形モデルよりもデータを適切に表現できるというところにある。もしパラメータの事後分布が b_2 の確信できる値として 0 よりずっと小さい値になったとしたら、線形モデルは適切でなかったという確認を得たことになる（線形モデルでは $b_2 = 0$ であるから）。

それでは、ここからは二乗項を含んだ階層モデルへの拡張を行っていこう。MCMC を効率化するためデータを標準化し、標準化データで推定したパラメータを再度オリジナルのスケールに変換するという前例の手続きをここでも実施する（式 17.2, p. 494）。なお、今回は二乗項を含むモデルのため、式 17.2 よりも少しばかり代数計算が増えることになる。

3) 2009 年から 2011 年の間の世帯収入の中央値に関するデータで、http://www.census.gov/hhes/www/income/data/Fam_Inc_SizeofFam1.xls から取得したもの（2013 年 12 月 11 日時点）。

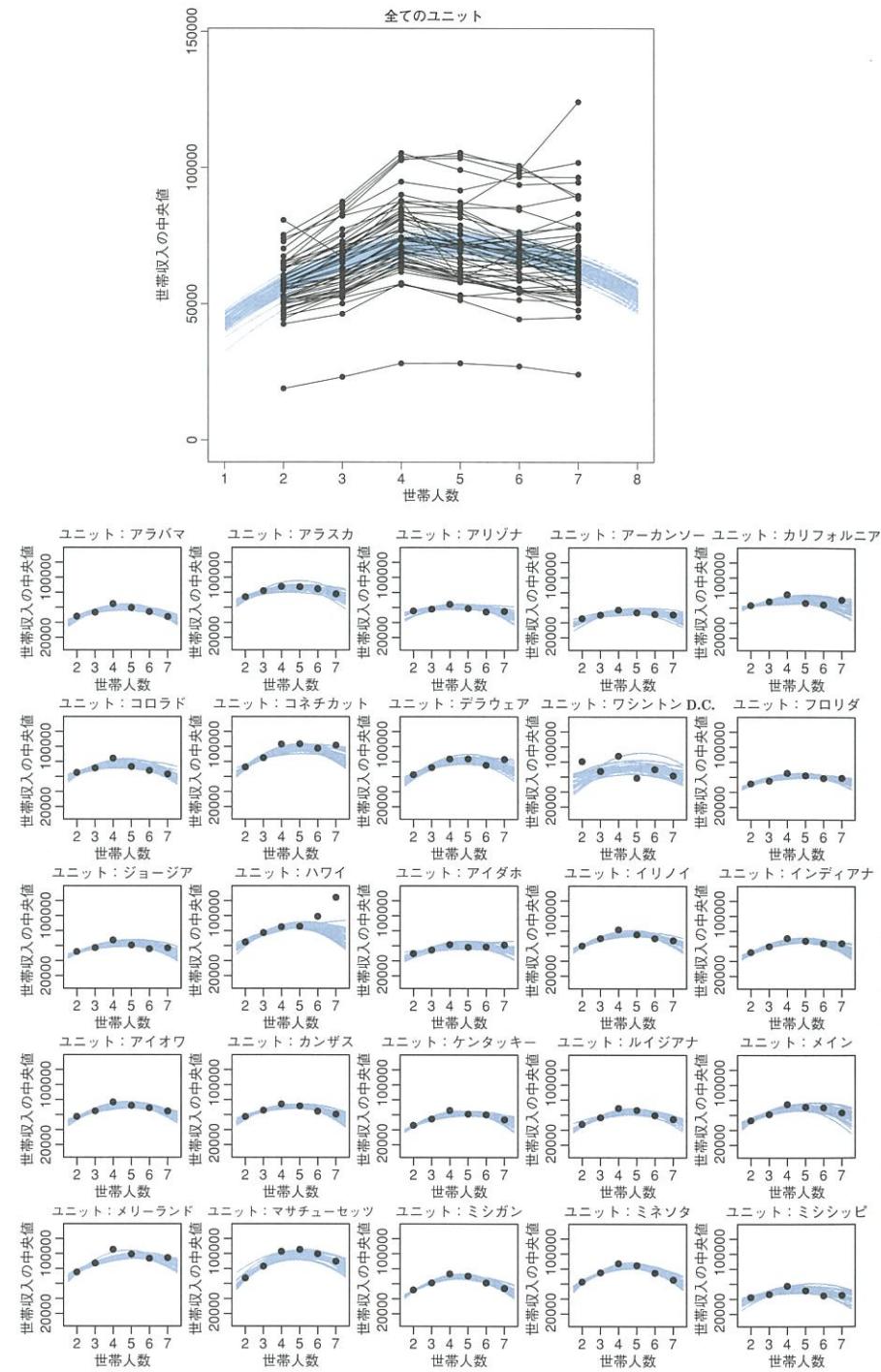


図 17.7 上段：様々な世帯人数における世帯収入の中央値。プエルトリコ、ワシントンD.C.と米国の50州それぞれに、個別で曲線が引かれている。下段：州別のデータ。推定区間内の2次曲線をデータに重ねて表示している。

$$\begin{aligned}
 z_{\hat{y}} &= \zeta_0 + \zeta_1 z_x + \zeta_2 z_x^2 && \text{モデルの定義より} \\
 \frac{(\hat{y} - M_y)}{SD_y} &= \zeta_0 + \zeta_1 \frac{(x - M_x)}{SD_x} + \zeta_2 \frac{(x - M_x)^2}{SD_x^2} && \text{式 17.1 から} \\
 \hat{y} &= \underbrace{\zeta_0 SD_y + M_y - \zeta_1 M_x SD_y / SD_x + \zeta_2 M_x^2 SD_y / SD_x^2}_{\beta_0} \\
 &\quad + \underbrace{\left(\zeta_1 SD_y / SD_x - 2\zeta_2 M_x SD_y / SD_x^2 \right) x}_{\beta_1} + \underbrace{\zeta_2 SD_y / SD_x^2 x^2}_{\beta_2} \quad (17.3)
 \end{aligned}$$

続いて、式 17.3 を JAGS にそのまま実装していく（モデル式と、標準化データを用いた式の両方をコード化する）。これで、標準化された係数と、オリジナルスケールの係数のサンプリングを JAGS によって記録することができる。

データの標準化に加えてもう1点、今回は世帯収入のデータに対する修正を行っていこう。今回使用するデータは世帯人数別での世帯収入の中央値だが、世帯人数の少ない家族集団と世帯人数の多い家族集団の間にはサンプルサイズの違いがある。それゆえ、世帯収入には、世帯人数ごとに異なる大きさのノイズが混入してしまっている。典型的には、世帯人数の少ない家族が多い一方で世帯人数の大きな家族はデータ数が少ないため、大家族における収入の中央値には大きなノイズが含まれてしまっていることが考えられる。このことを考慮するためには、できるだけデータのノイズを取り除き、タイトなパラメータ推定が可能なモデルを構築することが必要になってくる。

幸いなことに、アメリカ合衆国国税調査局は、世帯収入の中央値とともにその「誤差の幅」を報告している。今回はその誤差の幅を用いてモデルのノイズパラメータを調整したい。誤差の幅が広ければ、ノイズパラメータの値はそれにつれて大きくなる。誤差の幅が小さければ、ノイズパラメータは相対的に小さくなる。もっと形式的にいえば、すべてのデータに対して同じ σ を使って $y_i \sim \text{normal}(\mu_i, \sigma)$ とする代わりに、 $y_i \sim \text{normal}(\mu_i, w_i \sigma)$ とする。ここで w_i は第 i 番目のデータに対応した相対標準誤差である⁴⁾。 w_i が小さいときは、 μ_i の値はデータにベストフィットする位置から比較的近い範囲にとどまるよう推定される。一方、 w_i の値が大きい場合、 μ_i は尤度の大きな低下ではなく、幅広い範囲の値をとることができるようになる。総じて、このモデルでは、誤差範囲の小さいデータ点ではパラメータ値を狭い区間で求め、誤差範囲の大きいデータ点ではゆとりのある範囲からパラメータの推定値を探すことができるようになっている。以下に示す JAGS コードでは、オリジナルスケールの各標準誤差は、全体で平均した標準誤差の値で除算されている。したがって、重み付けの値が 1.0 の場合は全体の標準誤差と同じということになる。

4) 重み付けデータを用いた別の回帰分析などでは、重み付けの係数が精度や誤差分散の逆数であるために、ノイズパラメータを重み付け係数で除算する場合がある。しかし、この式での w_i はノイズや誤差そのものを意味している。したがって、今回は重み付け係数の除算ではなく σ による乗算でデータに重み付けを行う。

ノイズに対する重み付けと、二乗項を含んだモデルの JAGS コードは以下の通りである。

```
#データの標準化:
data {
  Ntotal <- length(y)
  xm <- mean(x)
  ym <- mean(y)
  wm <- mean(w) #y の標準誤差
  xsd <- sd(x)
  ysd <- sd(y)
  for ( i in 1:length(y) ) {
    zx[i] <- ( x[i] - xm ) / xsd
    zy[i] <- ( y[i] - ym ) / ysd
    zw[i] <- w[i] / wm #平均に比した誤差の重み
  }
}

#標準化データを用いたモデルの記述:
model {
  for ( i in 1:Ntotal ) {
    zy[i] ~ dt( zbeta0[s[i]]+zbeta1[s[i]]*zx[i]+zbeta2[s[i]]*zx[i]^2 ,
      1/(zw[i]*zsigma)^2 , nu )
  }
  for ( j in 1:Nsubj ) {
    zbeta0[j] ~ dnorm( zbeta0mu , 1/(zbeta0sigma)^2 )
    zbeta1[j] ~ dnorm( zbeta1mu , 1/(zbeta1sigma)^2 )
    zbeta2[j] ~ dnorm( zbeta2mu , 1/(zbeta2sigma)^2 )
  }
  #標準化スケールでの漠然とした事前分布の設定:
  zbeta0mu ~ dnorm( 0 , 1/(10)^2 )
  zbeta1mu ~ dnorm( 0 , 1/(10)^2 )
  zbeta2mu ~ dnorm( 0 , 1/(10)^2 )
  zsigma ~ dunif( 1.0E-3 , 1.0E+3 )
  zbeta0sigma ~ dunif( 1.0E-3 , 1.0E+3 )
  zbeta1sigma ~ dunif( 1.0E-3 , 1.0E+3 )
  zbeta2sigma ~ dunif( 1.0E-3 , 1.0E+3 )
  nu <- nuMinusOne+1
  nuMinusOne ~ dexp(1/29.0)
  #オリジナルスケールへの変換:
  for ( j in 1:Nsubj ) {
    beta2[j] <- zbeta2[j]*ysd/xsd^2
    beta1[j] <- zbeta1[j]*ysd/xsd - 2*zbeta2[j]*xm*ysd/xsd^2
    beta0[j] <- zbeta0[j]*ysd + ym - zbeta1[j]*xm*ysd/xsd
      + zbeta2[j]*xm^2*ysd/xsd^2
  }
  beta2mu <- zbeta2mu*ysd/xsd^2
  beta1mu <- zbeta1mu*ysd/xsd - 2*zbeta2mu*xm*ysd/xsd^2
  beta0mu <- zbeta0mu*ysd + ym - zbeta1mu*xm*ysd/xsd
    + zbeta2mu*xm^2*ysd/xsd^2
  sigma <- zsigma * ysd
}
```

上記 JAGS コードでは、 t 分布の平均パラメータに二乗項を含んだ新しい引数が使われており、精度の引数には各データの標準誤差を用いた重み付け係数である $zw[i]$ と、 $zsigma$ の積が与えられている。モデルブロックにおける尤度関数の記述後には、二乗項に対する事前分布の設定とオリジナルスケールへの変換を行う記述が後続している（式 17.3 を参照）。上記 JAGS コードの完全版は `Jags-Ymet-XmetSsubj-MrobustHierQuadWt-Example.R` にあり、その中で用いられている関数は `Jags-Ymet-XmetSsubj-MrobustHierQuadWt.R` で定義されている。なお、完全版のスクリプトでは、標準誤差の情報がないデータを用いた場合に重み付けの変数名が省略される設定（あるいは明示的に `wName=NULL` と書くとよい）となっている。

上記モデルコードの Stan バージョンについては、エクササイズ 17.3 で紹介する。

17.4.1. 結果と解釈

図 17.7 に重ねられた回帰曲線を見れば、明らかな湾曲があることがおわかりいただけただろう。実際、二乗項の係数における 95% HDI は、-2200 から -1700 であり、0 から遠い負の値をとっている（サンプルコードを実行して確認してほしい）。このことから、今回のデータには非線形なトレンドがあると判断できる。

また、階層モデルを用いたことによる縮小も生じている。例えば、図 17.7 の図の中にあるハワイのデータ（上から 3 段目左から 2 列目）をご覧いただきたい。そこでは、データが右肩上がりであるにもかかわらず、推定結果は他の州と同様の逆 U 字曲線になっている。これは、他の多くの州が逆 U 字曲線で表現されるデータであることに起因しており、その推定値に向かって縮小が生じた結果を反映したものである。ただし、もしハワイのデータサイズが十分に大きかったならば（それゆえノイズの重み付けが極めて小さくなっていた場合には）、ハワイ単独の傾向はデータにもっとフィットしていく、右肩上がりに推定されていたことだろう。こういったことから、集団レベルからの縮小というのは個別データへのベストなフィッティングを妥協することとも言い換えられるかもしれない。

なお、結果の解釈、特に傾きの解釈には注意が必要となる。線形回帰に二乗項を加えた場合、傾きの値に大きな意味はない。今回推定された傾きの 95% HDI は 16,600 から 21,800 だった。このような大きな値となるのは、モデル式に二乗項が含まれているからである。しかしこの傾きが、それ自身で、二乗項が大きく値を減らすため、データをかなり通り越してしまう。

標準誤差によるデータの重み付けの作用は、世帯人数の少ない家族より多い家族で推定の不確実性が高くなっているという結果にも確認することができる。図 17.7 のカリフォルニア（やその他の多くの州）を例にとって考えてみよう。推定区間内の曲線は、世帯人数が 2 人のところで狭く束ねられ、7 人のところでは広く散らばっている。これは、世帯人数の多い家族の標準誤差が大きいことが一因である。標準誤差を用いた重み付けを行う

ことで、世帯人数の多い家族についての柔軟な予測がなされている（曲線が拡散してしまう他の原因としては、世帯間・世帯内の可変性がある）。

そして事後分布からは、各州のデータに外れ値が含まれていることが示されている。というのも、推定された正規化パラメータの値が極めて小さく、ほとんどすべての事後分布で ν の値が4以下となっているためである。これは、各州内のデータ点のほとんどが2次曲線によって近似されていることを意味している一方で、 t 分布の裾にあるような外れ値の存在を示唆しているものだ。

17.4.2. モデルのさらなる拡張

今回のモデルは、話を明確にするために多くの部分を単純化してきた。1つは、線形モデルと二乗項のみを使用したというところにある。ベイズ用ソフトウェアを使えば、高次の多項式や三角関数、指數関数などを用いた非線形モデル、その他数学的に定義されるいかなるトレンドの実装も簡単に行うことができる。

また、今回のモデルでは、個人（あるいは州）全体で1つのノイズ分布を仮定した。このノイズは各データの相対的な標準誤差によって調整することができるが、基準量となる σ はすべての個人で同じものを使用していた。このような仮定に必然性はなく、個人ごとに個別で一貫したノイズがデータに含まれているという仮定をおいてもよい。例えば、1日の様々なタイミングで何度も血圧を測定したとしよう。対象者の何人かは、他とくらべて血圧の変動が小さく一貫した血圧データとなるかも知れない。このような場合、各個人に十分なデータがあるならば、個人ごとに標準偏差やノイズパラメータを仮定したモデルへと拡張することもできる。そうなると、ノイズの少ない個人はノイズの多い個人にくらべて集団レベルの推定により多くの情報を与えることになるだろう。このとき、個人ごとのノイズパラメータは集団レベルのガンマ分布から派生したものとして、 $\sigma_i \sim \text{gamma}(r, s)$ のように階層化することもできる。ここでの r と s は推定されるパラメータである。このようにすれば、個人ごとのノイズパラメータを集団レベルのガンマ分布で表すことが可能となる。この例は、後に図19.6(p. 583)で紹介する。

さらに、今回は二乗項や傾き、切片の階層パラメータに正規分布を使用したことでも単純化した点の1つである。本来的には、各パラメータの値にも外れ値が含まれている可能性がある。そのような場合が想定されるならば、集団レベルの階層パラメータに t 分布を仮定することもできる。そのようなモデルの拡張もベイズ用ソフトウェアを使えば簡単に実装することが可能であり、これによって複数の階層において外れ値にロバストなモデルを構築することができるようになる。しかし、正規化パラメータの推定は外れ値の存在を前提にしているため、データサイズが小さい場合や極端な値を持つ個人のデータがない場合には、その力を發揮しない。データが少ない場合は、 t 分布の仮定によって得られるものが少ないとということには留意してほしい。

もう1つの単純化は、傾き、切片、そして二乗項の係数間に共変関係を仮定するパラ

メータを設定しなかったというものである。例えば、低体重で生まれたラットは成長率が低く、大きな体で生まれたラットの成長率は高いといわれている。このような場合、切片（出生時の体重）と傾き（成長率）には共変関係があるといえる。もちろん、どんな場合でも切片と傾きが共変するというわけではないが、ここでいいたいのは、今回はそのような共変関係を捉えられるパラメータを明示的に設定していなかったということだ。共変関係を捉えたい場合には、多変量正規分布を切片や傾き、二乗項の係数パラメータに設定するという方法がある。こちらも、ベイズ用ソフトウェアを使えば比較的簡単に実装可能である。多変量正規分布を用いた共変関係の推定例については、WinBUGS Examples Volume 2 の、the “birats” example を参照してほしい (http://www.mrc-bsu.cam.ac.uk/wp-content/uploads/WinBUGS_Vol2.pdf)。

17.5. モデル拡張の手続きとリスク

17.5.1. 事後予測チェック

データと事後分布を比較する手続きは事後予測チェックと呼ばれる。事後予測チェックによって事後分布とデータの間に一貫したずれが確認されたときは、よりデータにフィットするようなモデルの拡張や変更を考える必要がある。

かといって、どのようなモデルを代替的に用いればよいのか、そこに明確な答えがあるわけではない。代替するモデルというのは、理論的に意味のあるモデルや、計算上の利点が明解である必要はある。理論的に意味があるモデルというのは、教科書で目にするような馴染みのある関数を用いたものもありうる。その馴染み深さや単純さというものの一例が、今回の二乗項を用いたモデルだった。それとは別に、応用領域の理論を背景に構築したモデルを用いることにも意義があるだろう。例えば、世帯収入の高低においては、世帯人数に占める大人の割合が大きくかかわってくると考えられる。そういった背景を踏まえれば、世帯収入が大人の人数に従ってその平方根の関数として増加するというモデル（なぜなら大人はそれが家族にお金を持ってくる可能性があるから）を考えることもできる。また逆に、子どもの人数に従って指數関数的に収入が低下するというモデル（なぜなら子どもは大人がお金を持ってくる時間を奪ってしまうから）も想定できるだろう。これが事実かどうかはわからないゆえ、あくまでたとえ話の範囲を出ないが、そういったモデルを用いることも可能となる。

モデルの拡張が意図するところは、よりよくデータを捉えるところにある。拡張の1つの方法は、パラメータを追加することである。すでに紹介した二乗項を含むモデルがパラメータ追加の一例であった（二乗項のあるモデルは、線形モデルにパラメータを追加したものといえる）。パラメータ追加前と追加後のモデルが入れ子になった関係にある場合、従来のモデルに比して拡張モデルのほうがデータをうまく表現できているかを確認するこ

ともできる。この例でいえば、二乗項の係数が0になるか否かによって、もとの線形モデルで十分かどうか判断できる。そして、二乗項にかかるパラメータの推定値が0からかけ離れている場合には、拡張したモデルのほうがうまくデータを表現できているということになる。

モデル拡張の別の方法として、従来のモデルとはまったく異なるモデルを適用してみるという手段もある。新しいモデルとオリジナルのモデルのどちらがよりデータに適合するかについては、ベイジアンモデル比較という方法で原理的には実現できる。ただし、まったく異なるモデルの比較を行う場合には、両モデルの事前分布の情報量が同等である必要が生じてくる。モデル比較の詳細については、セクション10.6.1(p. 299)をご覧いただきたい。

データに合わせてモデルを再構成することは発展の1つではあるが、再構成したモデルを同一データの分析に還元するということを問題視する立場もある。このような批判から、同じデータにモデルを「二度づけ」することが慣習化されてしまっている背景さえうかがえよう。もちろん、私たちはどのようなモデル空間についても考えたいように考えることができると、異なる理論や異なる研究背景に合わせてモデルを発展させていくモチベーションも持っている。分析の結果、もしも手元のデータが新しい傾向や新しい関数の必要性を示唆するのであれば、おそらくそれは分析者にとって思いがけないものだったことだろう(二度づけしたくなるであろう)。その意味で、私たちはデータから予想外の傾向が示されるという可能性を常に胸にとどめておくべきだといえる。また、特定のデータセットに見られた新奇な傾向と、それを踏まえた新たなモデルは、新たなデータによって確かめられるものであることも覚えておかなければならない。

事後予測チェックのもう1つのアプローチとして、事後予測サンプリング分布を生成し、実データと予測値の不一致を測るという方法がある。このアプローチでは、ペイズ p 値を用いて、データと予測の不一致が生じる確率を考えていく。もしもペイズ p 値がとても小さい場合、モデルは棄却される。第11章で行った p 値についての議論から、このような方法に疑問を持つ方もいるかもしれない。ペイズ p 値に関する議論の詳細は、(Kruschke, 2013b) を参照してほしい。

17.5.2. JAGSあるいはStanによるモデル拡張の手続き

ペイズ用ソフトウェアの大きな利点の1つは、データに対して適切で理論的に意味のあるモデルを柔軟に実装できることである。この柔軟性を武器にして、既存のプログラムを修正して使っていきたいと思われることだろう。本章で紹介した線形モデルから二乗項を含むモデルへの拡張はその一例である。以下に、モデル拡張のポイントを整理しておく。

- 新規のパラメータを加える際には、丁寧なモデル構成を行うこと。図17.6のようなダイアグラムを書くことで、モデルに必要なすべてのパラメータとその事前分布を

整理しやすくなる。また、ダイアグラムとモデルコードにおけるパラメータ間の関係(矢印)は対応するため、このダイアグラムはコードを書くときにも有用となる。例えば、線形モデルに二乗項を加えて拡張したいときには、係数パラメータとその事前分布を枝葉のように書き込む。これだけで、傾きと二乗項の係数を区別したモデルが概念的に構成できる。具体的には、平均値の式に二乗項を加えてモデルを拡張することになる。これは、 $\mu_{ij} = \dots + \beta_{2,j} x_{ij}^2$ のように、確立された既存の式へ二乗項を付け加えるだけでよい。二乗項の係数を集団レベルで階層化する場合も、傾きの階層化と同じ手続きで実現することができる。

- 新しく追加するパラメータに対する適切な事前分布の設定を確認すること。もとからあったパラメータの事前分布の設定が、新しいモデルにおいても適切かどうか確認すること。
- 新しいパラメータについても慎重に初期値を設定すること。既存のパラメータの初期値が拡張したモデルにおいても妥当かどうか検討すること。JAGSは、明確な初期値を指定しない限りすべてのパラメータをデフォルトの設定で扱うよう自動化されている。特に初期値指定の必要がない場合、JAGSのデフォルト設定を借りれば手間は大きく省くことができる。
- JAGSで新しいパラメータのサンプリング記録を行うこと。JAGSによるサンプリング記録は、明確に指定したパラメータについてのみ行われる。一方、Stanの場合はデフォルトですべての非局所変数(パラメータや変換パラメータ)の記録が行われる。
- 拡張モデルを適切に表現するために、要約グラフの修正を行うこと。この作業には頻繁にエラーがつきまと、多くの時間を費やしてしまう。JAGSではなくRでグラフの出力をを行うため、修正したJAGSモデルに合わせてRコードを改変する必要がある。パラメータがモデルで果たす役割を確認するときやデータの解釈を考えるにあたっては、パラメータの周辺事後分布をプロットする必要が出てくることもあるだろう。パラメータの事後分布間に相関関係があるかどうかについて確認する際には、pairs関数を用いたプロットが必要になるかもしれない。データの傾向を示す事後予測分布とデータを重ねて図示したい場合や、パラメータや他の識別子間の差異を図示したい場合などもある。時間のかかる作業だが、グラフ出力のためにRコードを書いていくことが求められる。

17.5.3. パラメータの追加に伴うリスク

新しいパラメータを追加するとき、拡張モデルはデータにフィットする柔軟性を獲得することになる。これに伴い、従来のモデルにも含まれたパラメータの確信区間がより広くなる可能性も生じてしまう。図17.5(p. 500)の階層線形モデルの例に戻って考えてみ

よう、図に示されたような線形傾向を持つデータに対してであっても、二乗項を含む階層線形モデルを適用するのは原理的に可能である。このモデルを用いれば、例えば図17.8のような予測を行うことができる⁵⁾。図中のトレンドラインは、データに一貫した正か負の曲率を持つ曲線で示されている。オリジナルスケールのデータを用いた場合、曲率と傾きには強いトレードオフな関係が生まれる。つまり、曲率が正の値をとると傾きが小さくなり、曲率が負の値になると傾きが大きくなるということが起こってしまう。それゆえ、二乗項の係数パラメータを導入することで（たとえその値が0であっても）傾きパラメータの確からしさは低下してしまう。線形傾向を持つデータに対して曲率の値を0に固定したモデルを適用した場合、傾きの95% HDIは+2.2から+4.1の値をとるが、係数を固定せず純粹に二乗項モデルを適用した場合には、曲率が0に近い値で推定されても（95% HDIは約-0.10から+0.07となる）、傾きの推定区間は大きく広がってしまう（95% HDIが約-7.0から+16.5となる）。このことから、2次式的傾向がないデータに二乗項を含むモデルを適用した場合、傾きの推定値をぼやけさせてしまうだけで、ほとんど何の情報も得られないことがわかる。しかし、傾きの推定が不確かになる程度はデータのスケールに依存する部分もある。標準化したデータを使用した場合、傾きパラメータの95% HDIはわずかしか変化しない。

こういったパラメータ推定における不確実性の増加は、モデルの複雑化に伴う副作用の1つである。入れ子になったモデルの場合、その中の単純なモデルは複雑なモデルのパラメータを固定したもの（例えば0とする）であるから、より複雑なモデルにくらべて入れ子になっているパラメータ推定値の精度が低いものになる。推定精度が低下する度合いは、モデルの構造やパラメータ設定、データの特性に依存して大きくなっていく。特に、推定精度の低下は、パラメータ間にトレードオフの関係がある状態でデータをフィットした際に明白になる。

複雑なモデルは、ベイジアンモデル比較を行うときにも不利な局面を生じさせる。複雑なモデルであるほど、パラメータ空間は広大となる。このような大きなパラメータ空間に事前分布を設定すれば、すべてのパラメータ値の組み合わせに関して、その事前確率は希釈されていくことになる。それゆえ、複雑なモデルでしか捉えることのできないデータを除けば、一般的にモデル比較においてはシンプルなモデルが採択されやすくなる。したがって、ベイジアンモデル比較の際には、各モデルの事前分布の情報量が同等になるような工夫を講じる必要が生じてくる。この詳細については、セクション10.6.1(p. 299)にて紹介している。

5) JAGSでは、推定時に多くのサンプリングやバーンイン期間を必要とし、正規化パラメータのサンプリングに高い自己相関が生じてしまう。しかし、正規化パラメータの周辺事後分布にかかるすべてのチェイン系列の振る舞いは似ていて、他のパラメータについては問題が生じることはない。Stanでは、必要なバーンイン期間は短く、正規化パラメータは滑らかにサンプリングされていく。ただし、一時的にサンプリングの移動幅が小さくなるという落とし穴にはまるることはできない。エクササイズ17.3は、StanとJAGSによる分析を比較し、これらのことと実際に確かめてもらうという内容である。

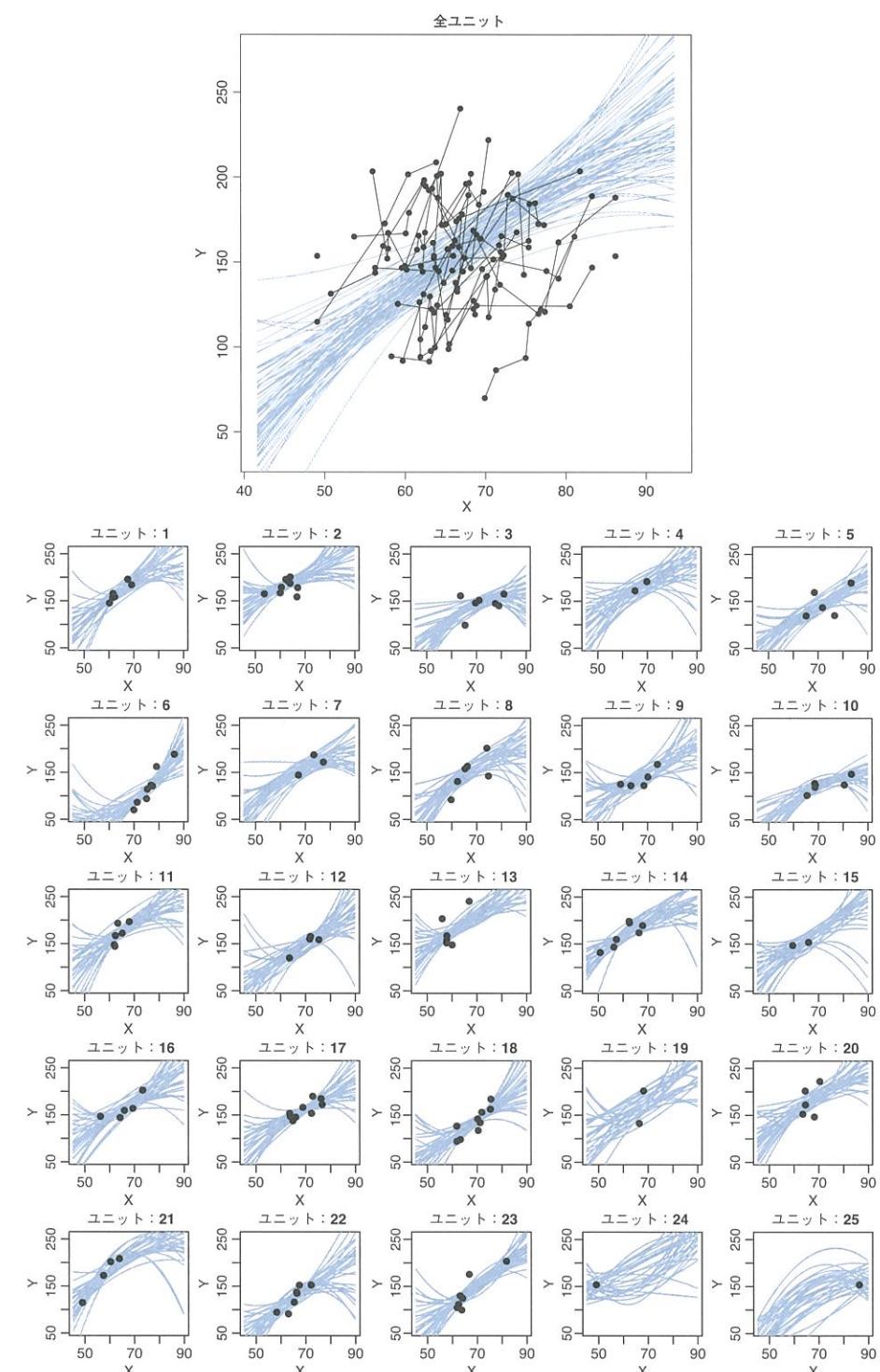


図 17.8 二乗項を含むモデルを図17.5のデータに適用した場合、二乗項にかかる係数はほぼ0となっているにもかかわらず、二乗項の柔軟性はより不確かになっている。

17.6. エクササイズ

以下のエクササイズは一例である。すべてのエクササイズに取り組みたい場合は、<https://sites.google.com/site/doingbayesiandataanalysis/>へアクセスしてもらいたい。
エクササイズ 17.1. [目的：2次式のトレンドを持つデータに線形モデルを適用し、事後予測チェックを実施しよう]

(A) Jags-Ymet-XmetSsubj-MrobustHier-Example.R のコードを変えて、世帯収入に関するデータの分析を実施してみよう。線形モデルによってデータをうまく表現することはできるだろうか。適用した線形モデルとデータ間に「系統的な不一致」とは何を指し示したものだろうか。データが曲がっている、というのでは答えにならない。モデルとデータの差異が、どのように、どこの部分で生じているかという観点から、系統的な不一致について言葉にしてみよう。

(B) 他のモデルの適用結果を参照することなしに、線形モデルを棄却する方法はあるだろうか。言い換えれば、この状況ではどのようにベイズ p 値を計算すればよいだろう。あるいは、それをすべきだろうか。この問題に関する詳細な情報は Kruschke (2013b) を参照してほしい。

エクササイズ 17.2. [目的：データが標準化されていないときの JAGS の自己相関を確認する]

標準化されていないデータを用いた分析を実施するために、階層化されていない単純な回帰モデルの JAGS コード Jags-Ymet-Xmet-Mrobust.R を修正しよう（オリジナルコードを失わないよう、複製したファイルを修正することをおすすめする）。修正では、データの標準化に関するコードとオリジナルスケールへ変換するためのコードを削除することになる。当該部分の修正が完了したら、事前分布の設定が標準化前のデータスケールに適しているか確認する。そして、コードを実行して診断グラフを示し、その結果について議論しよう。最後に、サンプリング数を増やして（パソコンの容量に負荷がかけられない場合は、間引きも行う）、標準化したデータと同等に収束するか確認してみよう。

エクササイズ 17.3. [目的：Stan と JAGS の収束の違いを検証する]

図 17.5 に示された架空データの分析に二乗項を含んだモデルを適用した場合、p. 514 の脚注 5 で紹介したような問題が JAGS と Stan に生じてしまう。このときの JAGS コードは Jags-Ymet-XmetSsubj-MrobustHierQuadWt-Example.R と Jags-Ymet-XmetSsubj-MrobustHierQuadWt.R で、Stan コードは Stan-Ymet-XmetSsubj-MrobustHierQuadWt-Example.R と Stan-Ymet-XmetSsubj-MrobustHierQuadWt.R にある。分析の前に、Stan コードの一部を以下に示す。セクション 17.4 の JAGS のコードと、どのような点が異なるか探してみよう。

```

data {
  int<lower=1> Nsubj ;
  int<lower=1> Ntotal ;
  real y[Ntotal] ;
  real x[Ntotal] ;
  real<lower=0> w[Ntotal] ;
  int<lower=1> s[Ntotal] ;
}
transformed data {
  // データの標準化手続き：
  real zx[Ntotal] ;
  real zy[Ntotal] ;
  real zw[Ntotal] ;
  real xm ;
  real ym ;
  real xsd ;
  real ysd ;
  xm <- mean(x) ;
  ym <- mean(y) ;
  xm <- mean(w) ;
  xsd <- sd(x) ;
  ysd <- sd(y) ;
  for ( i in 1:Ntotal ) { // ベクトル化できるだろうか？
    zx[i] <- ( x[i] - xm ) / xsd ;
    zy[i] <- ( y[i] - ym ) / ysd ;
    zw[i] <- w[i] / xm ;
  }
}
parameters {
  real zbeta0[Nsubj] ;
  real zbeta1[Nsubj] ;
  real zbeta2[Nsubj] ;
  real<lower=0> zsigma ;
  real zbeta0mu ;
  real zbeta1mu ;
  real zbeta2mu ;
  real<lower=0> zbeta0sigma ;
  real<lower=0> zbeta1sigma ;
  real<lower=0> zbeta2sigma ;
  real<lower=0> nuMinusOne ;
}
```

```

transformed parameters {
  real<lower=0> nu ;
  real beta0[Nsubj] ;
  real beta1[Nsubj] ;
  real beta2[Nsubj] ;
  real<lower=0> sigma ;
  real beta0mu ;
  real betalmu ;
  real beta2mu ;
  nu <- nuMinusOne+1 ;
  //オリジナルスケールへ変換：
  for ( j in 1:Nsubj ) { // ベクトル化できるだろうか？
    beta2[j] <- zbeta2[j]*ysd/square(xsd) ;
    beta1[j] <- zbeta1[j]*ysd/xsd - 2*zbeta2[j]*xm*ysd/square(xsd) ;
    beta0[j] <- zbeta0[j]*ysd + ym - zbeta1[j]*xm*ysd/xsd
      + zbeta2[j]*square(xm)*ysd/square(xsd) ;
  }
  beta2mu <- zbeta2mu*ysd/square(xsd) ;
  betalmu <- zbeta1mu*ysd/xsd - 2*zbeta2mu*xm*ysd/square(xsd) ;
  beta0mu <- zbeta0mu*ysd + ym - zbetalmu*xm*ysd/xsd
    + zbeta2mu*square(xm)*ysd/square(xsd) ;
  sigma <- zsigma * ysd ;
}
model {
  zbeta0mu ~ normal( 0 , 10 ) ;
  zbetalmu ~ normal( 0 , 10 ) ;
  zbeta2mu ~ normal( 0 , 10 ) ;
  zsigma ~ uniform( 1.0E-3 , 1.0E+3 ) ;
  zbeta0sigma ~ uniform( 1.0E-3 , 1.0E+3 ) ;
  zbeta1sigma ~ uniform( 1.0E-3 , 1.0E+3 ) ;
  zbeta2sigma ~ uniform( 1.0E-3 , 1.0E+3 ) ;
  nuMinusOne ~ exponential(1/29.0) ;
  zbeta0 ~ normal( zbeta0mu , zbeta0sigma ) ; // ベクトル化されている
  zbeta1 ~ normal( zbetalmu , zbeta1sigma ) ; // ベクトル化されている
  zbeta2 ~ normal( zbeta2mu , zbeta2sigma ) ; // ベクトル化されている
  for ( i in 1:Ntotal ) {
    zy[i] ~ student_t(
      nu ,
      zbeta0[s[i]] + zbeta1[s[i]] * zx[i] + zbeta2[s[i]] * square(zx[i]) ,
      zw[i]*zsigma ) ;
  }
}

```

Stan のプログラミングに関するヒントはセクション 14.4 (p. 422) を参照してほしい。

- (A) 世帯収入データの分析を Stan で実行しよう。分析の結果、集団レベルの傾きパラメータについては、有効サンプルサイズが JAGS とほぼ同じになるはずである。では、JAGS と Stan それぞれの分析には、どれくらいの時間が必要だっただろうか。また、Stan の収束の具合は JAGS よりも優れているだろうか。正規化パラメータやノイズパラメータの推定にかかるチェイン系列の振る舞いは、Stan のほうがより優れているだろうか。
- (B) 図 17.8 の架空データを今一度使用する。このデータを使った際に生じる JAGS や Stan 上の問題は、脚注 5 (p. 514) の通りである。JAGS と Stan による分析を実施し、各問題点を実際に確認すること。あなたが見過ごすことのできない問題は、JAGS で生じる正規化パラメータの自己相関の高さだろうか。それとも、Stan で生じる集団平均から外れたサンプリングの問題だろうか。