Rの利用に関する備忘録

奥村太一

1 デフォルトの関数

1.1 rbind() によるデータフレームの合併

共通の変数と個別の変数を含むデータフレーム data01, data02, data03 から、共通の変数部分を抜き出して縦につなげて 1 つのデータセット data04 を作りたいとする。

これらに共通の変数名が idnames, fsnames, bsnames, scnames に格納されているとすると、

```
data04 <- rbind(data01[,c(idnames,fsnames,bsnames,scnames)],
                data02[,c(idnames,fsnames,bsnames,scnames)],
                data03[,c(idnames,fsnames,bsnames,scnames)])
```

でよい。merge() を使ったり、変数名の指定に data01[,which(colnames(data01) %in% c(idnames, fsnames))] などとしたくなるのだが、これだとうまくいかない。

1.2 subset()

データフレームのうち条件を満たす行を抜き出す。

使用例 今、data01 の変数 position について、値 "1" か "2" を取る行のみを抜き出したいとする。このとき、

```
data01 <- subset(data01, position=="1"|position=="2")
```

とすればよい。なお、

```
attach(data01)
data01 <- data01[(position=="1")|(position=="2"),]
detach(data01)
```

としても同じことができる気がするのだが、うまくいかない（他の変数に謎の NA が発生する）。

1.3 boxplot()

箱ひげ図の描画。上から順に、上限の極値（ヒゲ）、上側四分位数、中央値、下側四分位数、下限の極値（ヒゲ）が描画される。上下のヒゲは、それぞれ

$$\begin{aligned} whisker &= median \pm 1.5 \times (\text{3rd hinge} - \text{1st hinge}) && (1.1)\\ &\approx median \pm 1.5 \times (\text{3rd quartile} - \text{1st quartile}) && (1.2)\\ &= median \pm 1.5 \times interquartile\ range && (1.3) \end{aligned}$$

である。これより極端な値は外れ値としてプロットされる。

2 psych

2.1 fa()

因子分析を実行する関数。

- r：相関行列、共分散行列、ローデータ行列
- nfactors：因子数
- n.obs：サンプルサイズ（相関行列を指定した場合に必要）
- rotate：因子の回転
  - 直交回転："none", "varimax", "quartimax" など
  - 斜交回転："promax", "oblimin", "geominQ" など
- fm：解の推定法
  - "minres"：OLS
  - "wls"：WLS
  - "gls"：GLS
  - "ml"：最尤法
- missing：TRUE にすれば、欠測部分が中央値もしくは平均で代入される。指定しなければ、ペアワイズ除去される。
- impute：欠測の代入（"median", "mean"）

分析結果のうち、因子負荷は $loadings、共通性は $communality、複雑性は $complexity、因子間相関は $Phi に格納される。

因子負荷行列と因子間相関行列は、因子番号が自動的に入れ替えられて表示される。抽出された因子の順序に従って表示させるには、

```
fact$loadings[1:32,order(colnames(fact$loadings))]
fact$Phi[order(rownames(fact$Phi)),order(colnames(fact$Phi))]
```

などとする必要がある[1]。

2.2 reverse.code()

逆転項目の処理をする関数。

- keys：逆転処理する変数を指定（1 と -1 からなり、-1 を指定した項目が逆転される）
- items：対象となるデータを指定

例えば、

```
data03[,1:10] <- reverse.code(keys=c(1,1,1,1,1,-1,-1,-1,-1,-1), items=data03[,1:10])
```

とすれば 10 項目のうち後半 5 項目が逆転処理される。

[1] fact に因子分析結果を代入、項目数 32 とする。

2.3 scrub()

最大値と最小値から外れた異常値を除去するための関数。

- x：データ行列
- where：チェックする変数（変数名か列数）
- min：取りうる最小値
- max：取りうる最大値

例えば、

```
data01[,14:30] <- scrub(data01[,14:30],min=1,max=4)
```

とすれば、data01 のうち 14-30 列の項目について、最小値 1 から 最大値 4 に収まらない値は NA に置き換えられる。

3 miceadds: Some additional multiple imputation functions, especially for 'mice'

3.1 紹介

不完全な多変量データを補完するための mice パッケージへの補完的な関数を含む[2]。

- plausible value の代入
- マルチレベル代入
- 多次元の説明変数に対応した部分最小二乗法（partial least squares: PLS）
- ネストされた代入

3.2 mi.anova()

χ^2 統計量の統合にもとづく D_2 統計量を用いて分散分析の F 値を統合する関数 [3]
以下の引数を指定する。

- mi.res: mids クラス[4]のオブジェクトを指定する
- formula: lm 関数の式を文字式で指定する。
- type: 平方和の種類。type=3 とすれば、car パッケージの Anova() 関数が使用される

使用例 今、mice() によって多重代入された結果が data05mi に格納されているとする。これを用いて、ee を従属変数、sex と school を要因とした分散分析を行い、タイプ III の平方和にもとづく結果を統合するには、以下のようにする。

```
> library("miceadds")
> mi.anova(mi.res=data05mi,formula="ee~sex*school", type=3)
Univariate ANOVA for Multiply Imputed Data  Type 3)

lm Formula:  ee~sex*school
```

[2] https://cran.r-project.org/web/packages/miceadds/index.html
[3] Allison, P. D. (2002). *Missing data.* Newbury Park, CA: Sage.
[4] mice パッケージで多重代入された結果を含む。

```
R^2=0.052704
.....................................................................
ANOVA Table
                  SSQ df1       df2 F value  Pr(>F)    eta2 partial.eta2
sex          20.93089   1  399.4617 20.6394 0.00001 0.00931      0.00974
school       90.87520   2 2792.2570 48.5409 0.00000 0.04044      0.04094
sex:school    6.64428   2 3705.6280  3.5264 0.02951 0.00296      0.00311
Residual   2129.00412  NA        NA      NA      NA      NA           NA
```

分散分析表には、効果量（η^2 および 偏 η^2）が報告される。

4 mice: Multivariate imputation by chained equations

4.1 紹介

不完全な多変量データを補完するためのパッケージ (van Buuren & Groothuis-Oudshoorn, 2011)[5]

多変量データの補完には、joint modeling (JM) と fully conditional specification (FCS)[6] とが知られているが、このパッケージは後者を行うもの。

4.2 一般的枠組み

4.2.1 表記

- Y を p 変量データとして、変数 j の観測部分と欠測部分をそれぞれ Y_j^{obs} と Y_j^{mis}
- m を代入数、h 番目の代入データセットを $Y^{(h)}$（$h = 1, \ldots, m$）
- 変数 j を除いたデータセットを Y_{-j}
- 興味のある量を Q

4.2.2 多重代入法のモジュラーアプローチ

1. 多重代入の実行：`mice()` で実行し、代入されたデータ（$Y^{(1)}, \ldots, Y^{(m)}$）は `mids` クラスで保存
2. 代入されたデータによる分析：`with.mids()` で実行、結果（$\hat{Q}^{(1)}, \ldots, \hat{Q}^{(m)}$）は `mira` クラスで保存
3. 分析結果の統合：`pool()` で実行、結果（$\bar{Q}$）は `mipo` クラスで保存

4.2.3 MICE アルゴリズム

連鎖方程式 *chained equations* によって、多変量データの代入に係る様々な問題[7]に対処

多変量の完全データ Y は未知のパラメータ θ のみによって発生するとする。今、t 番目の連鎖方程式を考えると、Gibbs sampler

$$\theta_1^{*(t)} \sim P(\theta_1 \mid Y_1^{\text{obs}}, Y_2^{(t-1)}, \ldots, Y_p^{(t-1)}) \tag{4.1}$$

$$Y_1^{*(t)} \sim P(Y_1 \mid Y_1^{\text{obs}}, Y_2^{(t-1)}, \ldots, Y_p^{(t-1)}, \theta_1^{*(t)}) \tag{4.2}$$

$$\vdots$$

$$\theta_p^{*(t)} \sim P(\theta_p \mid Y_p^{\text{obs}}, Y_2^{(t)}, \ldots, Y_{p-1}^{(t)}) \tag{4.3}$$

$$Y_p^{*(t)} \sim P(Y_p \mid Y_p^{\text{obs}}, Y_2^{(t-1)}, \ldots, Y_p^{(t)}, \theta_p^{*(t)}) \tag{4.4}$$

[5] https://www.jstatsoft.org/article/view/v045i03

[6] 別名、multivariate imputation by chained equations (MICE)

[7] 予測に使うデータそのものに欠測がある、2 値変数と連続変数など異なるタイプが混在している、現実的にありえない組み合わせ（妊娠している父親など）が生じる、などなど。

となる。ただし、$Y_j^{(t)} = (Y_j^{\mathrm{obs}}, Y_j^{*(t)})$ は t 番目の繰り返しにおける代入値である。
　通常の MCMC と異なり、10-20 回程度の更新で収束する。

4.2.4　簡単な例

欠測データの検査　データが data01 に data frame として格納されているとする。このとき、

```
md.pattern(data01)
```

で各欠測パターンとその人数の内訳を、

```
> md.pairs(data01)
$rr
     x1  x2  x3  y
x1   25  16  17  15
x2   16  16  16  13
x3   17  16  17  14
y    15  13  14  15

$rm
     x1  x2  x3  y
x1    0   9   8  10
x2    0   0   0   3
x3    0   1   0   3
y     0   2   1   0

$mr
     x1  x2  x3  y
x1    0   0   0   0
x2    9   0   1   2
x3    8   0   0   1
y    10   3   3   0

$mm
     x1  x2  x3  y
x1    0   0   0   0
x2    0   9   8   7
x3    0   8   8   7
y     0   7   7  10
```

で「変数 × 変数」形式の行列で欠測パターンを出力することができる。r は観測、m は欠測を表す。例えば、(x2, y) の組み合わせでは、双方とも欠測のなかったケースが 13、前者が欠測で後者は欠測がないケースが 3 あった、といったことがわかる。

代入値の作成　データ data01 から代入値を発生させて imp に代入し結果を閲覧するには、関数 mice() を用いて

```
imp <- mice(data01, seed=23109)
print(imp)
```

のようにする[8]。
　デフォルトでは、代入の数は $m = 5$。もし代入回数を増やしたいのであれば、引数として m=50 のように追加する。
　また、Gibbs sampler の更新回数は maxit で指定できる。デフォルトでは 5 回。

[8] seed は乱数発生の種。なお、mice() 実行時に収束結果など出力させないためには、引数 print=F を指定すれば良い。

代入値のチェック　負のカウント値、妊娠中の男性など、代入によってありえないデータが発生していないかチェックすること。代入結果が `imp` に格納されており、変数 x の代入結果をチェックしたいなら、

```
imp$imp$x
```

で代入値が行列（欠測ケース ID × 代入回数）で表示される。

全変数の代入済みデータセットは、関数 `complete()` で返される。代入結果が `imp` に格納されているとすると、例えば 2 回目の代入済みデータセットは、

```
complete(imp, 2)
```

で返される。

また、代入された全てのデータを一気に縦につなげて表示させるには、

```
complete(imp, "long")
```

と指定する。

代入済みデータの分析　代入済みの各データセットに分析を行うには、関数 `with.mids()` を用いる。

例えば、`imp` に代入された変数 `y,x1,x2` を用いて回帰分析を行うのだとすると、

```
fit <- with(imp, lm(y~x1+x2))
```

でよい。

!?　`with.mid()` でなく、ただの `with()` でよいのか？　**!?**

この結果をプールして表示させるには、

```
print(pool(fit))
summary(pool(fit))
```

などとする。すると、

```
               est    se    t    df Pr(>|t|)   lo 95  hi 95 nmis  fmi lambda
(Intercept)  5.96 74.53 0.08  9.22     0.94 -162.04 173.96   NA 0.46   0.36
x1          29.73 14.88 2.00  4.33     0.11  -10.37  69.82    0 0.73   0.63
x2           5.14  2.19 2.35 12.91     0.04    0.41   9.87    9 0.33   0.23
```

のような出力が得られる。

出力について、

- `fmi` は *the fraction of missing information*
- `rambda` は欠測データに帰属できる分散の割合（$\lambda = (B + B/m)/T$）

である（Rubin, 1987）。

4.3　代入モデル

4.3.1　7 つの選択肢

以下のことを**常に**考慮する必要がある。

1. MAR が仮定できるか、それとも MNAR になっているか。MICE は両方扱えるが、後者なら代入値の発生に追加のモデリング仮定が必要となる。
2. 代入モデルの形式（代入される変数の尺度に応じて）
3. 代入に用いる説明変数の選択
4. 他の変数の関数になっている変数を代入すべきかどうか
5. 代入される変数の順番
6. 初期値と更新回数、収束の確認手段
7. 代入回数 m

4.3.2 一変量の代入法

一変量の代入法は表 4.3.2 の通り。この名前を、代入すべき変数の列ごとに `mice()` 内の引数 `method=` で指定する。

もし `method="norm"` のように一つだけ指定すれば全ての変数がその方法で代入される。

一方、`method=c("","norm","pmm","mean")` のように列ごとに異なる代入法を指定することもできる[9]。

Table 4.1: 一変量の代入法

方法	意味	尺度のタイプ	デフォルト
`pmm`	Predictive mean matching	numeric	Y
`norm`	Bayesian linear regression	numeric	
`norm.nob`	Linear regression, non-Bayesian	numeric	
`mean`	Unconditional mean imputation	numeric	
`2L.norm`	Two-level linear model	numeric	
`logreg`	Logistic regression	factor, 2 levels	Y
`polyreg`	Multinomial logit model	factor, >2 levels	Y
`polr`	Ordered logit model	ordered, >2 levels	Y
`lda`	Linear discriminant analysis	factor	
`sample`	Random sample from the observed data	any	

各変数の型については、`str(data01)` のように確認する。

4.3.3 説明変数の選択

代入対象となる変数ごとに、説明変数の組を指定することができる。

代入結果が `imp` に格納されているとき、どの変数がどの変数によって説明されているかは、次のように確かめられる[10]。

```
> imp$predictorMatrix
    x1  x2  x3   y
x1   0   0   0   0
x2   1   0   1   1
x3   1   1   0   1
 y   1   1   1   0
```

行が代入対象となる変数、列が説明変数を表す。値が 1 であればその説明変数が使われたことを、0 であれば使われなかったことを表す。この例では、`x1` は欠測がないので代入の対象となっておらず、`x2` は `x1, x3, y` を説明変数として代入されていることがわかる。

説明変数の除去　仮に `x2` を説明変数に用いたくなかったとする。この場合、

```
> pred <- imp$predictorMatrix # あるいは imp$pred
> pred[, "x2"] <- 0 # x2 の列を全て 0 にする
> imp <- mice(data01, pred=pred)
```

でよい。

代入のスキップ　仮に、`x2` を代入対象としたくなかったとする。この場合、

```
> meth <- imp$meth
> meth["x2"] <- "" # x2 の代入方法を空欄にする
> imp <- mice(data01, meth=meth)
```

[9] `""` とされた変数は、不完全であっても代入されない。なお、欠測のない変数は自動的に代入対象から除外される。

[10] `predictorMatrix` は `pred` と省略できる。

で代入がスキップされる。
　時間をかけずに imp を作成したい場合、更新回数をゼロとした代入を実行すれば良い。これには、mice() に引数 maxit = 0 を指定する。これで、大きなデータでも長い時間かけずに imp$pred や imp$meth を得て編集できる。

マルチレベルデータへの代入

- JM 法：Schafer & Yucel (2002); Yucel (2008); Goldstein et al. (2009) など
- FCS 法：Jacobusse (2005) など

MICE では、van Buuren (2010) を改良した手法を採用しており、Kasim & Raudenbush (1998) による、級内誤差分散の変動を許す線形マルチレベルモデルに対する Gibbs sampler を実行する。
　Hox (2002) のデータ popmis を用いる。

```
> head(popmis,5)
  pupil school popular sex texp const teachpop
1     1      1      NA   1   24     1        7
2     2      1      NA   0   24     1        7
3     3      1       7   1   24     1        6
4     4      1      NA   1   24     1        6
5     5      1      NA   1   24     1        7
```

ここで、

```
> md.pattern(popmis)
     pupil school sex texp const teachpop popular
1152     1      1   1    1     1        1       1   0
 848     1      1   1    1     1        1       0   1
         0      0   0    0     0        0     848 848
```

より、欠測は popular にのみある（848 ケース）ことがわかる。
　さて、ここでは

- school: 集団を表す変数
- sex: 性別（1 or 0）（レベル 1 の変数）
- texp: 教師の経験（レベル 2 の変数）
- const: 切片（全員 1）

の 4 変数を説明変数として、

- popular: 人気（レベル 1 の変数）

の欠測を代入したいとする。
　predictorMatrix では、

- 0: 説明変数に用いない
- 1: 説明変数に用いる
- 2: ランダム変数
- −2: 集団を表す変数（1 つだけ指定可能）

と指定することになっており、代入法は 2l.norm を用いることになる（method で指定）。
もし、popular について

$$POPULAR_{ij} = \underbrace{\gamma_0 + u_{0j}}_{\text{random}} + \underbrace{\gamma_1 + u_{1j} \times SEX_{ij}}_{\text{random}} + \underbrace{\gamma_1 \times TEXP_j}_{\text{fixed}} + r_{ij} \tag{4.5}$$

のように予測を行うなら、

```
> imp <- mice(popmis, maxit = 0)
> pred <- ini$pred
> pred["popular", ] <- c(0, -2, 0, 2, 1, 2, 0)
> imp <- mice(popmis, meth = c("", "", "2l.norm", "", "", "", ""),
+ pred = pred, maxit = 1, seed = 71152)
```

のように predictorMatrix を指定すれば良い。

!? レベル 2 の説明変数や、所属グループ ID を代入することはできないのか？ **!?**

説明変数の選択についてのアドバイス 変数の数が数百にも及ぶような場合、多重共線性や計算上の問題により、全ての変数を説明変数として用いることは難しい。せいぜい 15 から 25 変数くらいが妥当なところか。
van Buuren et al. (1999) による、説明変数選択の手続き

1. 代入後分析モデルに登場する変数は全て含める
2. 欠測の発生に関係している変数を全て含める
3. 分散説明率の高い説明変数を含める（それを含めることで代入の不確実性が減るなら入れる）
4. 2 と 3 に該当するものでも、欠測の多すぎる説明変数は取り除く

説明変数の簡便な選択法 変数間の相関関係をチェックする。ペアワイズ除去した変数間の相関行列や、欠測の有無と観測値の相関行列など。後者なら、

```
> cor(y = data01, x = !is.na(data01), use = "pair")
       x1     x2     x3    y
x1      NA     NA     NA    NA
x2   0.086     NA  0.139 0.053
x3   0.008     NA     NA 0.045
y   -0.040 -0.012 -0.107    NA
```

とすると、data01 の各変数について、ある変数の値と別の変数の欠測の有無との相関（点双列相関）行列が返される[11]。上の例では、x2 の欠測と x1 の観測値には $r = .008$ の相関があるということになる。
また、代入対象となる変数が欠測で予測に用いる変数が観測であるケースが、代入可能となる変数が欠測で予測に用いる変数が観測であるケースと欠測であるケースの合計に対してどれくらい存在するかを算出して参照することもできる。予測に用いる変数に欠測が多ければ、そもそも説明変数として用いるほどの情報を持っていないということで外す。
関数 quickpred() によって、以上の基準を用いて説明変数行列 predictorMatrix を自動的に指定することができる。例えば、

```
imp <- mice(data01, pred = quickpred(data01, minpuc = 0.25, include = "x1"))
```

のようにすれば、予測に用いることのできるケースの割合が 0.25 以上の説明変数を自動的に含むこと、また x1 は常に含むこと、が指定できる。
quickpred() の引数としては、mincor（点双列相関の最小値）、minpuc（予測に利用できるケースの割合の最小値）、include, exclude（使用 or 不使用）がある。

[11] !is.na() は、欠測でなければ TRUE、欠測であれば FALSE を返す。

4.3.4 受動的代入法

変数変換を伴う代入について、大きく以下の 2 つがある。

- 不完全データを元のまま代入してから、完全データを変数変換する
- 不完全データを変数変換してから、変換後のデータを代入する

一方、代入アルゴリズムの中で変換前の変数と変換後の変数の双方が必要になる場合、**受動的代入法** *passive imputation* を用いることで、異なる変換同士の一貫性を保つことができる。

例えば、x1 を予測するには、x2 よりも log(x2) の方が適切だと考えたとする。

```
> data01 <- cbind(data01, logx2 = log(x2)) # 変数を変換して追加
> ini <- mice(data01, max=0, print=F)
> meth <- ini$meth # 代入方法を指定する行列
> meth["logx2"] <- "~log(x2)" # logx2 については、x2 を変換したものであることを明記
> pred <- ini$pred # 説明変数を指定する行列
> pred[c("x2","x3","y"), "logx2"] <- 0  # x2, x3, y は logx2 を予測に用いない
> pred["x1", "x2"] <- 0 # x1 の予測には x2 は用いない（logx2 がある）
> imp <- mice(data01, meth=meth, pred=pred, seed=38788, print=F)
```

とすればよい。新たな（欠測入りの）log(x2) は x1 の代入にのみ用いられ、それ以外の変数の代入には x2 が用いられることになる。

4.4 MICE の実行

4.4.3 収束診断

mice() が返した値を plot() にわたしてやれば良い。例えば、

```
> imp <- mice(data01, seed=23109,maxit=20)
> plot(imp,c("b1","b2","b3"))
```

とすれば、変数 b1, b2, b3 の更新回ごとの代入値の平均と SD が図 4.1 のようにプロットされる[12]。

!? 「更新回ごとの代入値の平均と SD」の意味がよくわからない。Gibbs sampler の各更新では、候補値が 1 つではなく複数サンプリングされるのだろうか？ **!?**

4.4.5 代入値のチェック

元の不完全データの分布と、代入値を含む完全データの分布を描いてチェックしてみる。例えば、

```
> densityplot(imp,~age+b1)
```

で変数 age, b1 の分布が描画される（図 4.2）。

また、**傾向スコア** *propensity score* を条件付けたもとでの観測値と代入値の分布を比較することもある。

```
> x1.na <- is.na(data01$x1) # x1 の欠測の有無を格納
> fit.x1 <- with(imp, glm(x1.na~x2+x3+y, family=binomial)) # x1 の欠測を完全データから予測
> ps <- rep(rowMeans(sapply(fit.x1$analyses, fitted.values)), 6) # 傾向スコア（5 つ分を平均）
> xyplot(imp, x1~ps|.imp) # プロット
```

4.5 MICE 実行後

4.5.1 データの繰り返し分析

関数 with.mids()[13] により各データセットを分析し、pool() で出力を統合する。

[12] シミュレーションによる検討では、5-10 回程度の更新で収束するとのこと。

[13] with() だけでも OK。

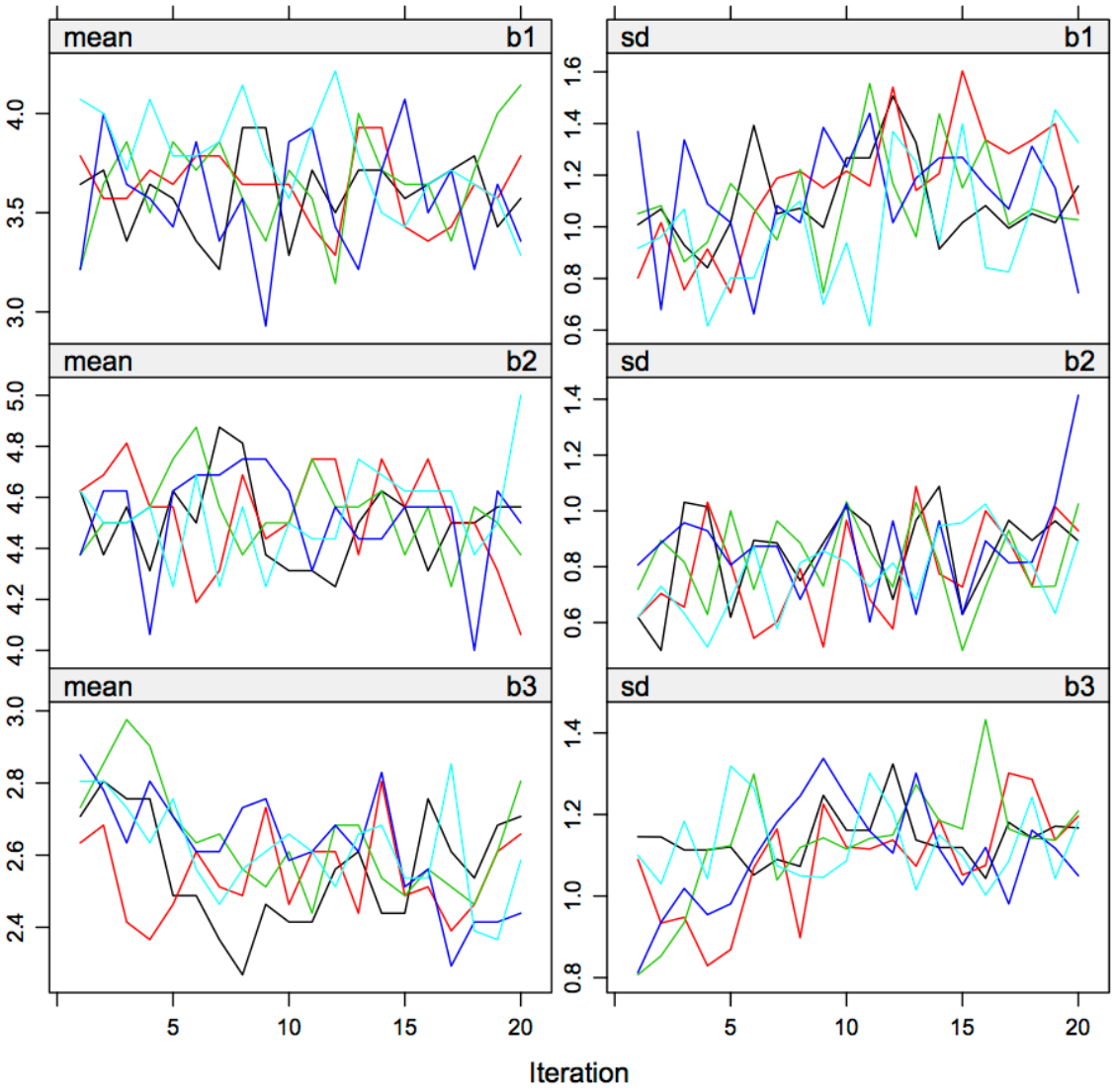

Figure 4.1: 収束診断

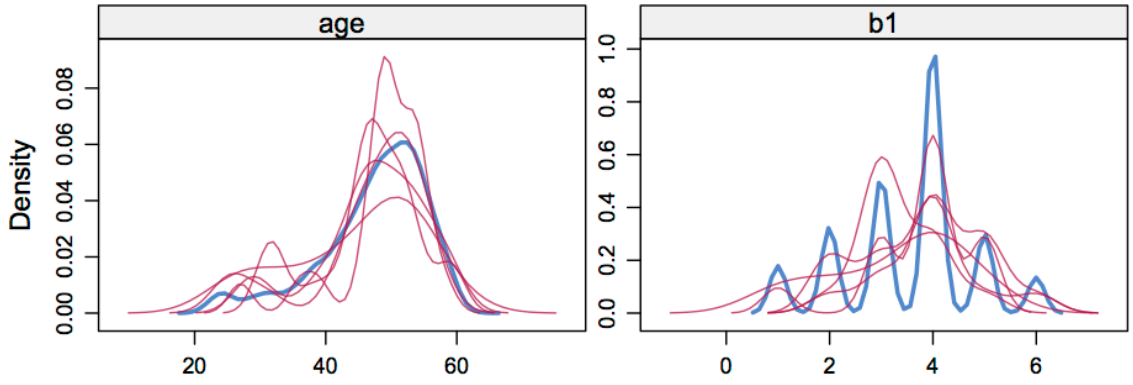

Figure 4.2: 代入値のチェック

4.5.2 結果の統合

関数 `pool()` は、`coef()` と `vcov()` メソッドを両方持っているオブジェクトであれば、どんなものにでも適用できる。また、`nlme` パッケージの `lme` クラスにも対応している。

- `pool.scalar()`：単一の推定値を統合する
- `pool.r.squared`：R^2、自由度調整済み R^2 を統合する
- `poo.compare()`：ネストされたモデルの比較を行う（ワルド検定、尤度比検定）