Multivariate multilevel analyses

A hands on primer in R

Jürgen Wilbert

2021/03/17

Abstract

A short description how to do Multivariate Multilevel Analyses in R with the `nlme` and `MCMCglmm` packages

Contents

```
cat("*Keywords: ", paste0(rmarkdown::metadata$keywords, collapse = ", "), "*"
)
```

Keywords: statistics, R, Multilevel, Bayesian, nlme, MCMCglmm

Problem

Most of our data have a multilevel structure. But "standard" Manova analyzes do not take a nested structure with various strata into account. What we need is a multivariate extension of the univariate multilevel regression approach.

Multivariate multilevel analyses has various subtypes depending on the assumptions of the intercorrelations and variance of the dependent variables (dvs).

In this paper by Ben Bolker (https://rpubs.com/bbolker/3336), five subtypes are described. Interestingly, the case #1 assumes equal variance of all dvs and equal (positive ?) correlations among the dvs. In this case, is is very easy: A random slope model with the dvs turned into a new random slope categorical variable. When dvs have different variances, the models must be weighted by their variance and when the intercorrelation between the dvs varies, the intercorrelations of the residuals have to be fixed to take these variations into account.

One approach is a Bayesian analyses with a Markov-Chain Monte-Carlo method. There is a vignette of the MCMCglmm package that explains how to do these analyses more in detail (<https://cran.r-project.org/web/packages/MCMCglmm/vignettes/CourseNotes.pdf>). Note: This example is for repeated measures.

A frequentist approach

Snijders and Bosker (2012) describe multivariate multilevel analyses and they provide an R example on this webage https://www.stats.ox.ac.uk/~snijders/mlbook.htm.

> Page 284 of Snijders and Bosker 2012: "To represent the multivariate data in the multilevel approach, three nesting levels are used. The first level is that of the dependent variables indexed by =1,…, , the second level is that of the individuals =1,…, , and the third level is that of the groups, =1,…, . So each measurement of a dependent variable on some individual is represented by a separate line in the data matrix, containing the values , , , , 1 , and those of the other explanatory variables. The multivariate model is formulated as hierarchical linear model using the same trick as in Section 15.1.3. Dummy variables 1,…, are used to indicate the dependent variables, just as in formula (14.2). Dummy variable is 1 or 0, depending on whether the data line refers to dependent variable or to one of the other dependent variables."

A computational example

First we need an example dataset to work with. I want two criteria (av1 and av2) and one explanatory variable (dv1) nested within four groups

```
set.seed(1234)
ncases <- 800
ngroups <- 4
dat <- list()
dat$id <- 1:ncases
dat$group <- rep_len(1:ngroups, ncases)
dat$av1 <- rnorm(ncases, 50, 10)
dat$av2 <- (dat$av1 + rnorm(ncases, 50, 10)) / 2
dat$dv1 <- (dat$av1 + dat$av2 + rnorm(ncases, 50, 10)) /3
dat$av1 <- dat$av1 + (dat$group / ngroups * 10)
dat$av2 <- dat$av2 + (dat$group / ngroups * 10)
dat$dv1 <- dat$dv1 + (dat$group / ngroups * 10)
dat$dv1 <- scale(dat$dv1, scale = FALSE)
#dat$id <- rep(1:ncases, ngroups)
#dat$obs <- 1:(ncases*ngroups)
dat <- as.data.frame(dat)

dat2 <- dat %>%
  pivot_longer(cols = c("av1", "av2"), names_to = "trait") %>%
  mutate(trait = as.factor(trait))
```

Manova without nesting data structure

The simple Manova has some disadvantages here:

- No nested data structure
- Assumes all variances are equal (also: across all strata, which it does not take into account anyway).
- Assumes all intercorrelations of the variables are equal (again also across all strata)

```
model <- lm(cbind(av1+av2) ~ 1 + dv1, data = dat)
summary(model)
```

```
##
## Call:
## lm(formula = cbind(av1 + av2) ~ 1 + dv1, data = dat)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.0270  -5.3806   0.1014   5.6296  24.7045
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 112.1179     0.3033  369.69   <2e-16 ***
## dv1           2.0226     0.0430   47.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.578 on 798 degrees of freedom
## Multiple R-squared:  0.735,  Adjusted R-squared:  0.7346
## F-statistic:  2213 on 1 and 798 DF,  p-value: < 2.2e-16
```

```
car::Anova(model, type = "III")
```

```
## Anova Table (Type III tests)
##
## Response: cbind(av1 + av2)
##               Sum Sq  Df  F value    Pr(>F)
## (Intercept) 10056336   1 136672.2 < 2.2e-16 ***
## dv1           162828   1   2212.9 < 2.2e-16 ***
## Residuals      58717 798
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multilevel regression approach

The following analyses follow the example of Snijders and Bosker (2012, Chapter 16).

Note: Dropping the intercept will set the main effect predictors of the dummy variable to the mean of the first variable. Also, dropping the main effect of the variable for the interaction will have an analogues effect for the interactions: the interaction with the first variable is displayed (otherwise, the main effect would entail the interaction of the first - here dummy - category and the variable).

```
model_1 <- lme(value ~ 0 + trait + dv1:trait, random = ~ 0 + trait |group,
data = dat2)
summary(model_1)
```

```
## Linear mixed-effects model fit by REML
##   Data: dat2
##        AIC      BIC    logLik
##   9937.194 9980.196 -4960.597
##
## Random effects:
##  Formula: ~0 + trait | group
##  Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev    Corr
## traitav1 1.2311183 tratv1
## traitav2 0.5143689 -0.816
## Residual 5.3422224
##
## Fixed effects:  value ~ 0 + trait + dv1:trait
##                  Value Std.Error   DF   t-value p-value
## traitav1     55.97952 0.6438845 1593  86.94032       0
## traitav2     56.13836 0.3190894 1593 175.93303       0
## traitav1:dv1  1.24563 0.0292196 1593  42.62999       0
## traitav2:dv1  0.81402 0.0279074 1593  29.16883       0
##  Correlation:
##              tratv1 tratv2 trt1:1
## traitav2     -0.628
## traitav1:dv1  0.000  0.000
## traitav2:dv1  0.000  0.000 -0.038
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.69589000 -0.66817031  0.02572767  0.63497999  3.13414812
##
## Number of Observations: 1600
## Number of Groups: 4
```

```r
#with weighted variances
model_1b <- update(model_1,  weights=varIdent(form=~1|trait))
summary(model_1b)
```

```
## Linear mixed-effects model fit by REML
##   Data: dat2
##        AIC      BIC    logLik
##   9870.822 9919.199 -4926.411
##
## Random effects:
##  Formula: ~0 + trait | group
##  Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev    Corr
## traitav1 1.2103832 tratv1
## traitav2 0.5547227 -0.774
## Residual 6.0607475
##
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | trait
##  Parameter estimates:
##       av1       av2
## 1.0000000 0.7442685
## Fixed effects:  value ~ 0 + trait + dv1:trait
##                  Value Std.Error   DF   t-value p-value
## traitav1     55.97952 0.6420068 1593  87.19460       0
## traitav2     56.13836 0.3199433 1593 175.46347       0
## traitav1:dv1  1.24415 0.0329916 1593  37.71123       0
## traitav2:dv1  0.81271 0.0239648 1593  33.91260       0
##  Correlation:
##              tratv1 tratv2 trt1:1
## traitav2     -0.632
## traitav1:dv1  0.000  0.000
## traitav2:dv1  0.000  0.000 -0.037
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.13902014 -0.69797783  0.02637369  0.64981098  2.76937510
##
## Number of Observations: 1600
## Number of Groups: 4
```

```
#correlation between dvs
model_1c <- update(model_1b, corr = corSymm(form = ~ as.numeric(trait)|group/id))

summary(model_1c)
```

```
## Linear mixed-effects model fit by REML
##   Data: dat2
##        AIC      BIC    logLik
##   9800.954 9854.707 -4890.477
##
## Random effects:
##  Formula: ~0 + trait | group
##  Structure: General positive-definite, Log-Cholesky parametrization
##          StdDev    Corr
## traitav1 1.2152290 tratv1
## traitav2 0.5592672 -0.832
## Residual 6.0605912
##
## Correlation Structure: General
##  Formula: ~as.numeric(trait) | group/id
##  Parameter estimate(s):
##  Correlation:
##   1
## 2 0.294
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | trait
##  Parameter estimates:
##       av1       av2
## 1.0000000 0.7442721
## Fixed effects:  value ~ 0 + trait + dv1:trait
##                  Value Std.Error   DF   t-value p-value
## traitav1     55.97952 0.6442894 1593  86.88568       0
## traitav2     56.13836 0.3219135 1593 174.38959       0
## traitav1:dv1  1.24557 0.0325430 1593  38.27480       0
## traitav2:dv1  0.81125 0.0236929 1593  34.24036       0
##  Correlation:
##              tratv1 tratv2 trt1:1
## traitav2     -0.633
## traitav1:dv1  0.000  0.000
## traitav2:dv1  0.000  0.000  0.243
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -4.14677763 -0.70322170  0.02487244  0.65471196  2.78718461
##
## Number of Observations: 1600
## Number of Groups: 4
```

```
#sjPlot::tab_model(model_1)


# lme(value ~ 0 + trait + dv1:trait,
#     random = ~ 0 + trait |group,
#     weights = varIdent(form = ~ 1|trait),
#     corr = corSymm(form = ~ as.numeric(trait)|group/id),
#     data = dat2
# )
```

Variation of this model:

By standardizing the dependent variable before calculating the models (‘value <- scale(value)‘), the main effects depict the standardized deviation from the overall mean of both variables and following that, the statistical test (se,t,p) represent the deviation of the mean of a variable from the overall mean.

MCMCglmm solution (most accurate)

The Bayesian solution by means of Markov-Chain Monte-Carlo glmm is able to take variable variances (random slopes) and variable intercorrelations of the dvs into account.

The model sometimes fails, asking for a better prior. Usually a rerun will fit the model (as the priors are set randomly as an iteration starting point). Here I chose a seed to make it work.

```
library(MCMCglmm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: coda
```

```
## Loading required package: ape
```

```
set.seed(12322)
model_2 <- MCMCglmm(cbind(av1, av2) ~ 0 + trait + dv1:trait, random = ~us(trait):id, rcov = ~us(trait):u
summary(model_2)
```

```
##
##  Iterations = 3001:12991
##  Thinning interval  = 10
##  Sample size  = 1000
##
##  DIC: -506.2845
##
##  G-structure:  ~us(trait):id
##
##                      post.mean l-95% CI u-95% CI eff.samp
## traitav1:traitav1.id   31.9930  29.2626  35.3370  454.946
## traitav2:traitav1.id   -3.1208  -3.6026  -2.7050    7.048
## traitav1:traitav2.id   -3.1208  -3.6026  -2.7050    7.048
## traitav2:traitav2.id    0.3063   0.2519   0.4008    3.961
##
```

```
##  R-structure:  ~us(trait):units
##
##                         post.mean l-95% CI u-95% CI eff.samp
## traitav1:traitav1.units     6.987    5.949    8.063    44.63
## traitav2:traitav1.units    11.830   10.412   13.137    96.42
## traitav1:traitav2.units    11.830   10.412   13.137    96.42
## traitav2:traitav2.units    20.101   18.247   22.101   900.90
##
##  Location effects: cbind(av1, av2) ~ 0 + trait + dv1:trait
##
##              post.mean l-95% CI u-95% CI eff.samp  pMCMC
## traitav1       55.9822  55.4701  56.4032   1160.2 <0.001 ***
## traitav2       56.1362  55.8311  56.4564   1000.0 <0.001 ***
## traitav1:dv1    1.1856   1.1252   1.2410   1113.3 <0.001 ***
## traitav2:dv1    0.8372   0.7932   0.8817    975.2 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Repeated measurements