Mixed Effects Models 3: Random Slopes

Yury Zablotski

Last updated on Oct 23, 2019 · 17 min read · 0 Comments · R

Previous topics or when do we need it

To keep this post short, I'll skip lots of explanations which were made in the previous posts. Especially **Mixed Effects Model 1** below is recommended to improve a digestion of this post. However, the **Repeated Measure ANOVA** corresponds to a mixed-effect model with both random intercepts and slopes. Thus, I'll recommend to read at least two first posts below:

- Repeated Measures ANOVA
- Midex Effects Model 1: Random Intercept
- Midex Effects Model 2: Crossed vs. Nested Random Effects

Why do we need it? What are the benefits?

To simply make more realistic models.

How to compute Random Slopes Mixed Effects Models in R

Load all needed packages to avoid interruptions.

```
library(tidyverse) # data wrangling and visualization
library(reshape2)  # data wrangling
library(lattice)   # for plotting
library(sjPlot)    # to visualizing random effects
library(ggeffects) # for plotting predictions of MEM
library(knitr)     # beautifying tables
library(lme4)      # "golden standard" for mixed-effects modelling in R (no p-values)
library(lmerTest)  # p-values for MEMs based on the Satterthwaite approximation
library(report)    # to report the model results
library(lsmeans)   # for a post-hoc analysis
library(broom)     # for tidy results
```

Get the politeness data from here

```
politeness =  read_csv("politeness_data.csv")
```

Simple Random Slope model

In order to connect to the previous post, let's have a look at the politeness data and one of the **random intercept models** again:

```
rim <- lmer(frequency ~ attitude + (1|subject), data=politeness, REML = F)
coef(rim)
```

```
## $subject
##    (Intercept) attitudepol
## F1    241.0250   -19.37241
## F2    266.7092   -19.37241
## F3    259.3919   -19.37241
## M3    179.0908   -19.37241
## M4    155.8311   -19.37241
## M7    113.4805   -19.37241
##
## attr(,"class")
## [1] "coef.mer"
```

Random intercepts assume that some people are more and some less polite. However, near the random intercept, we see a fixed slope (attitudepol = -19.4)! This is weird, because if we assume that subjects differ in their politeness from the very beginning (Intercept), why do assume that they all change their politeness-behavior (slope) in exactly the same way (namely -19.4)? Can completely different people all become identically less polite, if circumstances around them change (different scenarios)? This does not make much sense! If individuals (subjects) are different, they are different not only in the beginning.

Thus, it is only reasonable to assume that they would differently react to different scenarios in life. That is where **random slope models** come into play.

Writing up the model with both random intercept (Intercept is always 1) and random slope (attitudepol) for a subject at the same time in `lme4` package is very intuitive, you just add `+` them: `(1+attitude|subject)`. Models with random slopes automatically model random intercepts as well. Thus you might see syntax like that: `(attitude|subject)`, where Intercept `1+` is included implicitly.

```
rirsm = lmer(frequency ~ attitude + (1+attitude|subject), data=politeness, REML=F)
```

```
## boundary (singular) fit: see ?isSingular
```

```
coef(rirsm)
```

```
## $subject
##    (Intercept) attitudepol
## F1    243.5599  -24.089665
## F2    270.5533  -27.240880
## F3    262.9829  -26.357111
## M3    177.6010  -16.389596
## M4    152.7062  -13.483368
## M7    108.1253   -8.278974
##
## attr(,"class")
## [1] "coef.mer"
```

Now the slope (attitudepol) also changes for every subject, which makes the model more realistic.

Multiple Random Slope model

Adding new sources of variation is just as intuitive. For example, *subjects* and the *scenarios* can be defined as two different sources of variation in both intercept and slope. Moreover, we could model different intercepts and slopes for different (fixed) predictors:

```
politeness.model = lmer(frequency ~ attitude + (attitude|subject) + (attitude|scenario),
                        data=politeness, REML = F)
```

```
## boundary (singular) fit: see ?isSingular
```

```
coef(politeness.model)
```

```
## $scenario
##   (Intercept) attitudepol
## 1    189.1577   -19.60012
## 2    207.5845   -15.46383
## 3    214.5360   -21.52596
## 4    222.0009   -16.85213
## 5    200.4519   -19.12079
## 6    191.3214   -21.41048
## 7    193.0642   -23.58427
##
## $subject
##    (Intercept) attitudepol
## F1    243.8021  -24.234721
## F2    270.8640  -27.244435
## F3    263.2684  -26.399676
## M3    177.7194  -16.885295
## M4    151.7812  -14.000560
## M7    108.0935   -9.141802
##
## attr(,"class")
## [1] "coef.mer"
```

Interpretation: Different slopes of *attitudepol* show that voice frequency differ for every individual (subject) and every scenario. However, voice (*frequency*) of them all goes down (slope is always negative) when changing from *informal* attitude to the *polite* one, only for some people it goes down slightly more than for others.

```
summary(politeness.model)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: frequency ~ attitude + (attitude | subject) + (attitude | scenario)
##    Data: politeness
##
##      AIC      BIC   logLik deviance df.resid
##    823.3    845.1   -402.7    805.3       74
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0729 -0.6244 -0.1076  0.5560  3.3143
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  scenario (Intercept)  200.31  14.153
##           attitudepol   35.84   5.986   0.08
##  subject  (Intercept) 3774.29  61.435
##           attitudepol   46.68   6.833   -1.00
##  Residual              612.68  24.752
## Number of obs: 83, groups:  scenario, 7; subject, 6
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  202.588     25.928   6.507   7.814 0.000154 ***
## attitudepol  -19.651      6.518   7.050  -3.015 0.019361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## attitudepol -0.494
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

The summary of MEM also reports the correlation between the random intercepts and slopes. A high correlation is usually fine. However, a perfect or near perfect correlation (close to 1 or −1) indicates problems with fitting the model. Comparing two models, one with and one without random slopes might help to decide whether random slopes are necessary at all (even if it makes sense conceptually!):

```
politeness.model = lmer(frequency ~ attitude + (attitude|subject) + (attitude|scenario),
                        data=politeness, REML = F)
politeness.model_slopes = lmer(frequency ~ attitude + (1|subject) + (1|scenario),
                        data=politeness, REML = F)
anova(politeness.model, politeness.model_slopes)
```

```
## Data: politeness
## Models:
## politeness.model_slopes: frequency ~ attitude + (1 | subject) + (1 | scenario)
## politeness.model: frequency ~ attitude + (attitude | subject) + (attitude | scenario)
##                         npar    AIC    BIC  logLik deviance  Chisq Df
## politeness.model_slopes    5 817.04 829.13 -403.52   807.04
## politeness.model           9 823.32 845.09 -402.66   805.32 1.7166  4
##                         Pr(>Chisq)
## politeness.model_slopes
## politeness.model            0.7877
```

and a p-value of 0.7877, indicates that **intercept only model is more then enough** and **there is no need to use a more complex random slope model**. However, since the focus of this post is **random slope**, we'll continue with a more complex model.

On how to select and compare models

You may have realized, that every time we compare models, we use `REML = FALSE` argument while building them. REML stands for *restricted maximum likelihood*, and is `REML = TRUE` by default because it estimates the parameters more precise then maximum likelihood (`REML = FALSE`).

`REML` assumes that the fixed effects structure is correct. You should use maximum likelihood (ML) when comparing models with different fixed effects, as ML doesn't rely on the coefficients of the fixed effects - and that's why we are refitting our full and reduced models above with the addition of REML = FALSE as an argument.

Even though you use ML to compare models, you should report parameter estimates from your final "best" REML model, as ML may underestimate variance of the random effects.

The `anova()` function applies the Likelihood Ratio Test to compare models with Akaike Information Criterion (AIC), which is a measure of model quality, and is the lower the better. Markov Chain Monte Carlo (MCMC) or parametric bootstrap confidence intervals is even better then Likelihood ratio tests, but this Bayesian method is way outside of the scope of this post and will be treated separately.

Don't just compare all the possible combinations of variables to find something significant. Because it's **p-hacking**. But think about your experiment and hypothesis. Also, don't put to many variables into the model, because you'll overfit. Especially if you don't have enough data. As a rule of thumb, you need 10 times more data than parameters you are trying to estimate.

The model comparison is usually about the fixed effects. If you **carefully plan your experimental design and record data in a meaningful way**, you won't be needed to choose the random effects.

Always try to compare similar models:

- vary only random or only fixed effects at the same time,
- try not to compare "lmer" models with "lm" models (or "glmer" with "glm").
- compare "top-down" (from most complex to simple) rather then "bottom up" (from simple to complex). Unfortunately, you might arrive at different final models by using those strategies, thus, think what makes more sense.

Top-down procedure:

- fit a full model (more complex than you'd expect or want it to be)
- sort out the random effects
- sort out fixed effects
- once you arrive at the final model present it using REML estimation

NOTE: just because something is non-significant doesn't necessarily mean you should always get rid of it.

The golden rule

```
The solution to pseudoreplication is good experimental design, not fancier statistics.
  - Stuart H. Hurlbert
```

How to report results

Include p-values or AIC values from your comparison into your report/paper.

```
report(politeness.model)
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Random effect variances not available. Returned R2 does not account for random effects.
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Random effect variances not available. Returned R2 does not account for random effects.
## Random effect variances not available. Returned R2 does not account for random effects.
```

```
## boundary (singular) fit: see ?isSingular
```

```
## We fitted a linear mixed model (estimated using ML and nloptwrap optimizer) to predict frequency with a
##
##   - The effect of attitudepol is negative and can be considered as small and significant (beta = -19.65
##   - The effect of R2 (conditional) is NA and can be considered as very large and not significant (beta
```

```
tab_model(politeness.model, p.style = "both")
```

		frequency	
Predictors	*Estimates*	*CI*	*p*
(Intercept)	202.59 ***	151.77 – 253.41	**<0.001**
attitude [pol]	-19.65 **	-32.43 – -6.88	**0.003**
Random Effects			
σ^2	612.68		
$\tau_{00\ scenario}$	200.31		
$\tau_{00\ subject}$	3774.29		
$\tau_{11\ scenario.attitudepol}$	35.84		
$\tau_{11\ subject.attitudepol}$	46.68		
$\rho_{01\ scenario}$	0.08		
$\rho_{01\ subject}$	-1.00		
$N_{subject}$	6		
$N_{scenario}$	7		
Observations	83		
Marginal R^2 / Conditional R^2	0.138 / NA		

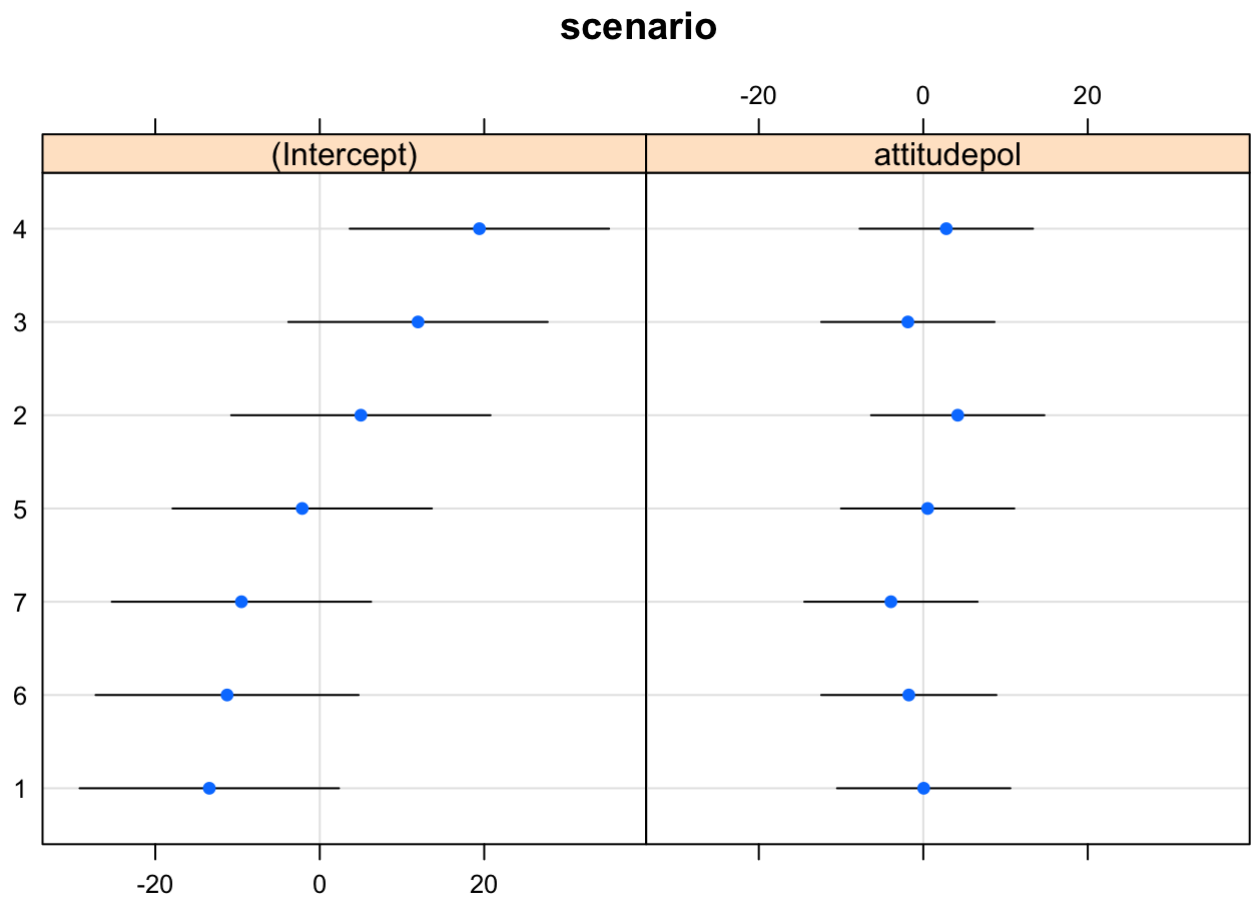
- *p<0.05 ** p<0.01 *** p<0.001

How to visualize a Random Slope Model

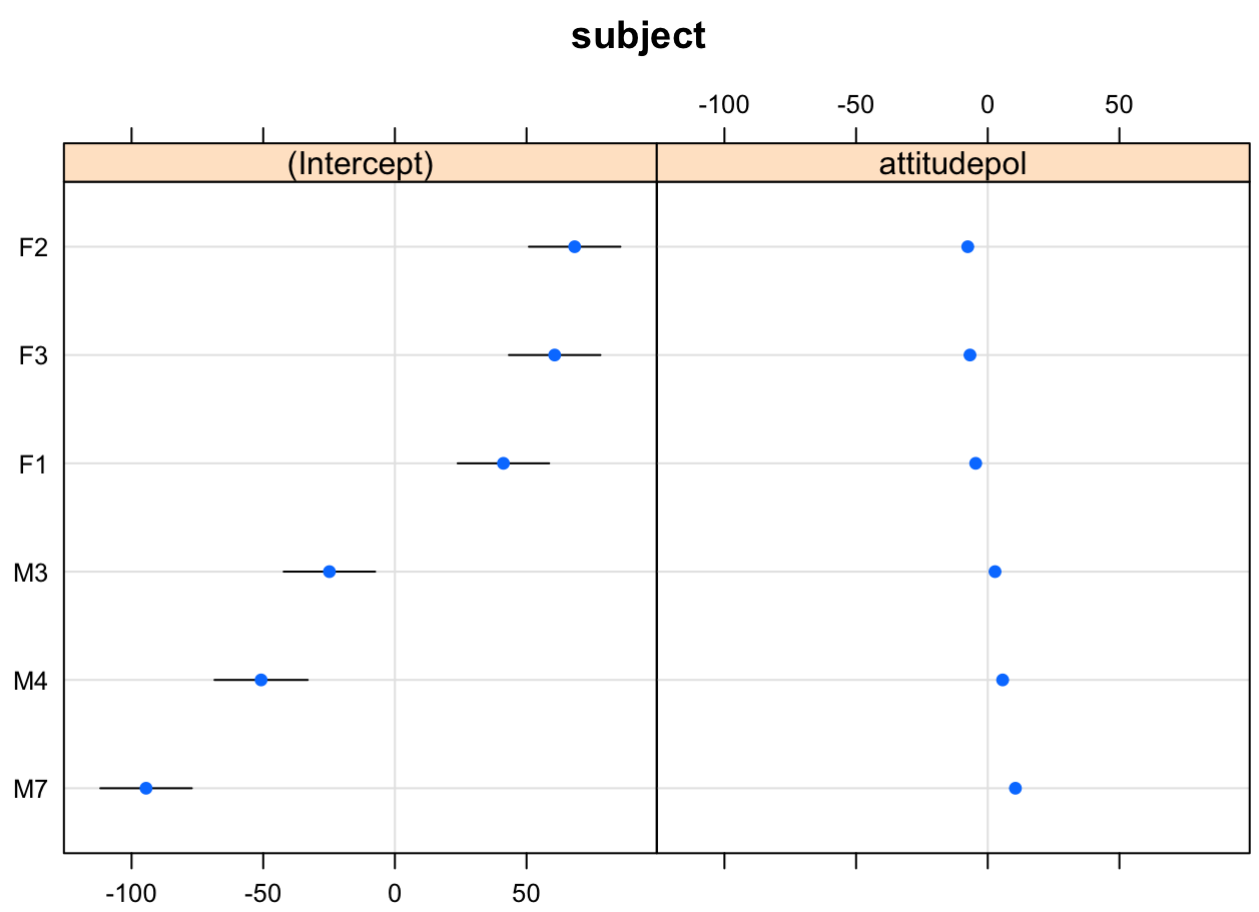
The variation among levels of your random effects can be shown by plotting the departure from the overall model estimate for intercepts and slopes:

```
dotplot(ranef(politeness.model, condVar=T))
```

```
## $scenario
```

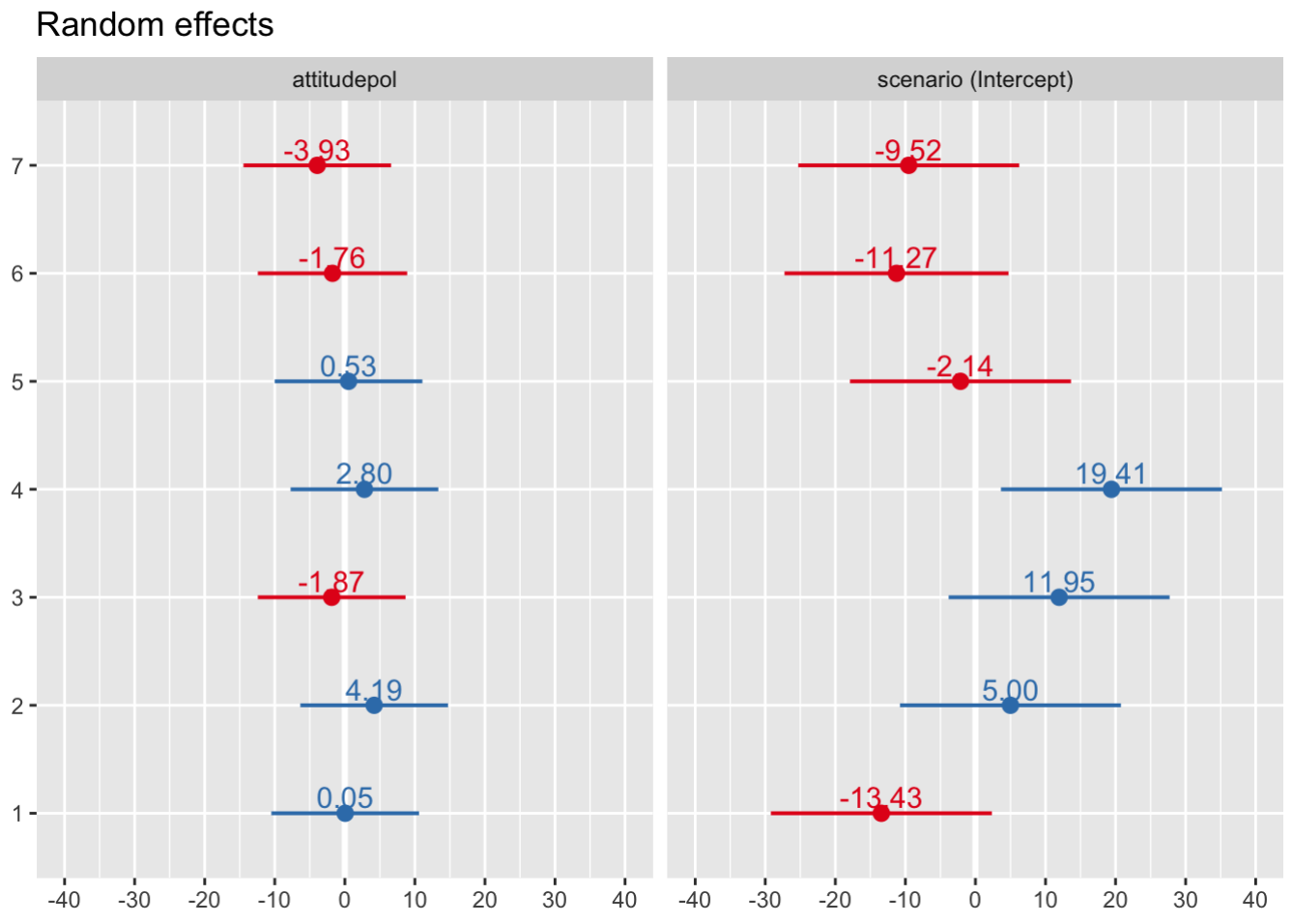



```
##
## $subject
```

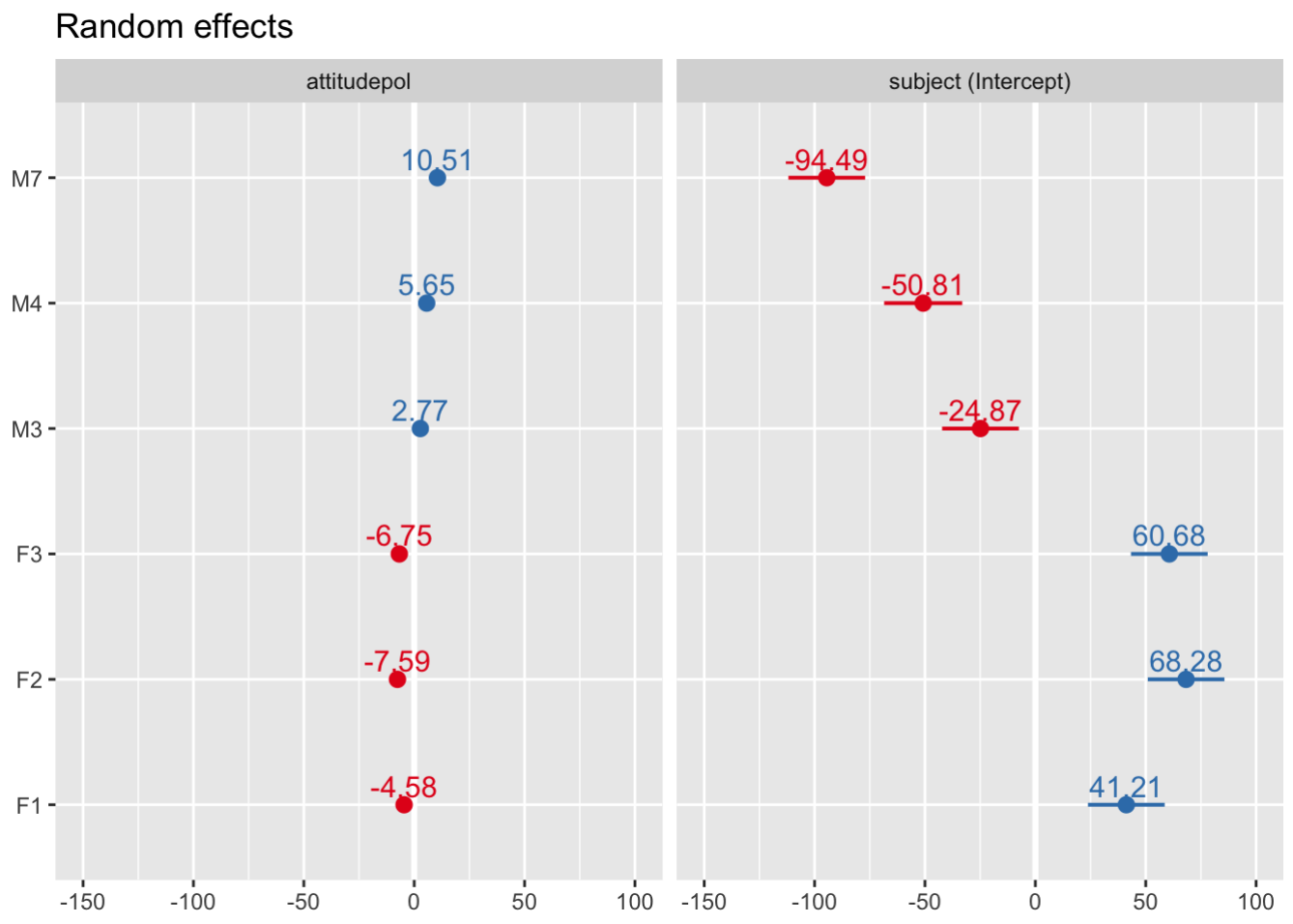



```
plot_model(politeness.model, type = "re", show.values = TRUE)
```

```
## [[1]]
```




```
##
## [[2]]
```



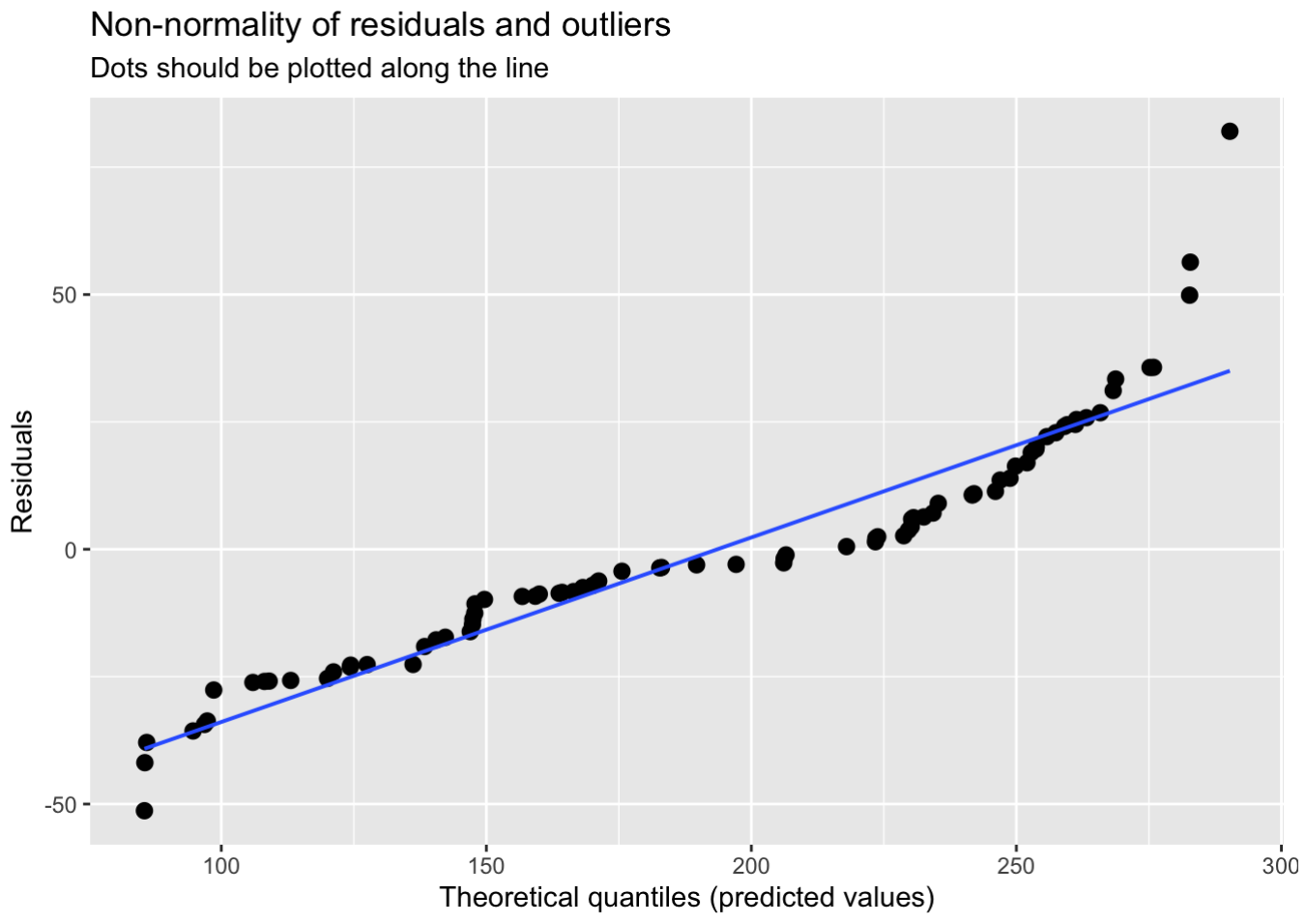
The values are the difference between the general estimate of the model and the specific level of random effect, be it intercept or slope. For instance, the relationship for the intercept of the first scenario is 202.588-189.1594 = 13.43, and since the intercept of the random effect is smaller, the answer gets a negative sign -13.43, meaning, the voice goes down. Slope works in the same way: -19.60342+19.651 = 0.05.

Besides, `plot_model()` function from the `sjPlot` package offers other visualization possibilities for MEMs. Simply put one the following into `type =` argument: “est”, “re”, “eff”, “emm”, “pred”, “int”, “std”, “std2”, “slope”, “resid” or “diag”. For example “diag” checks the normality and heteroscedasticity assumptions:

```
plot_model(politeness.model, type = "diag", show.values = TRUE)
```

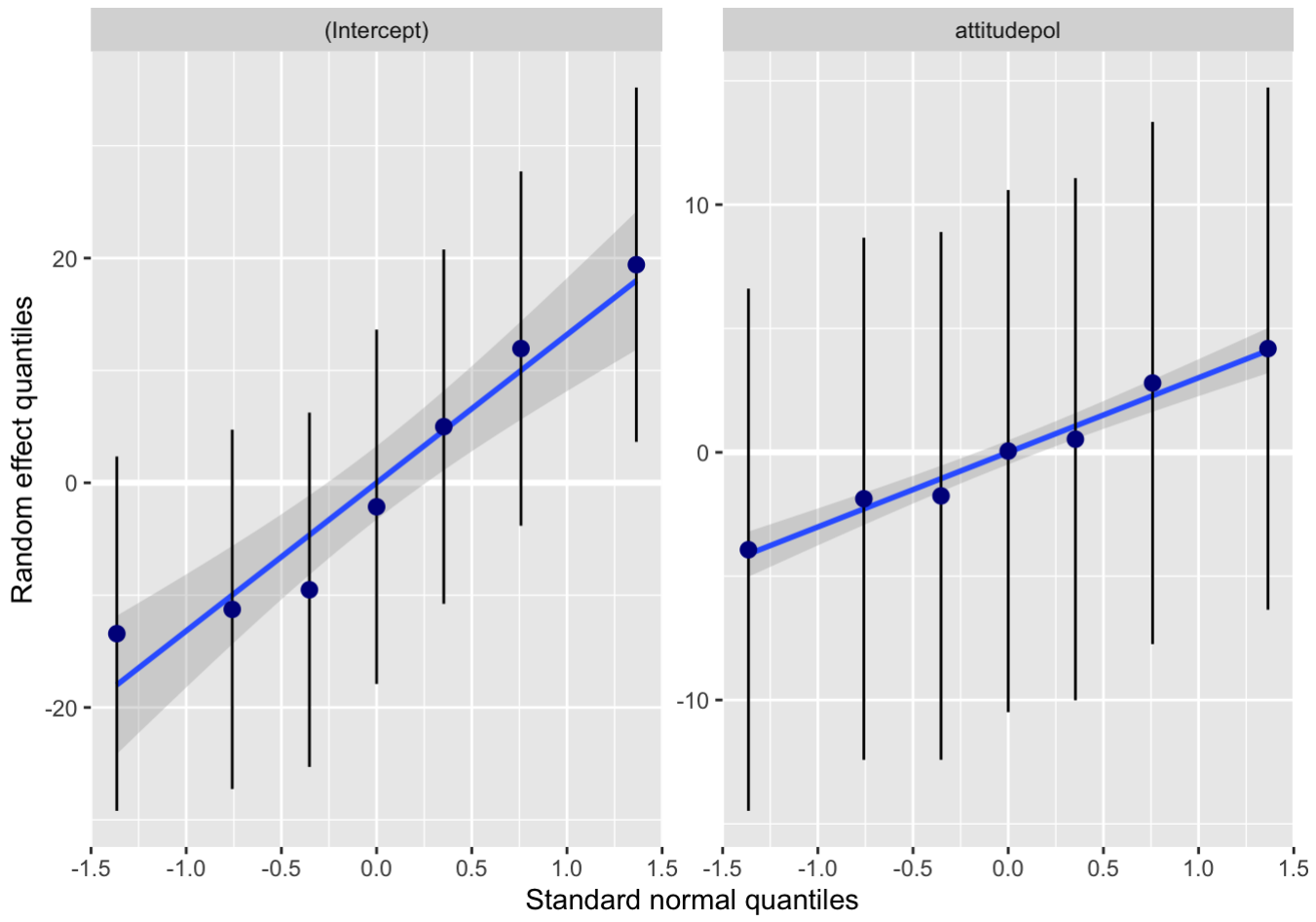
```
## [[1]]
```

```
## `geom_smooth()` using formula 'y ~ x'
```



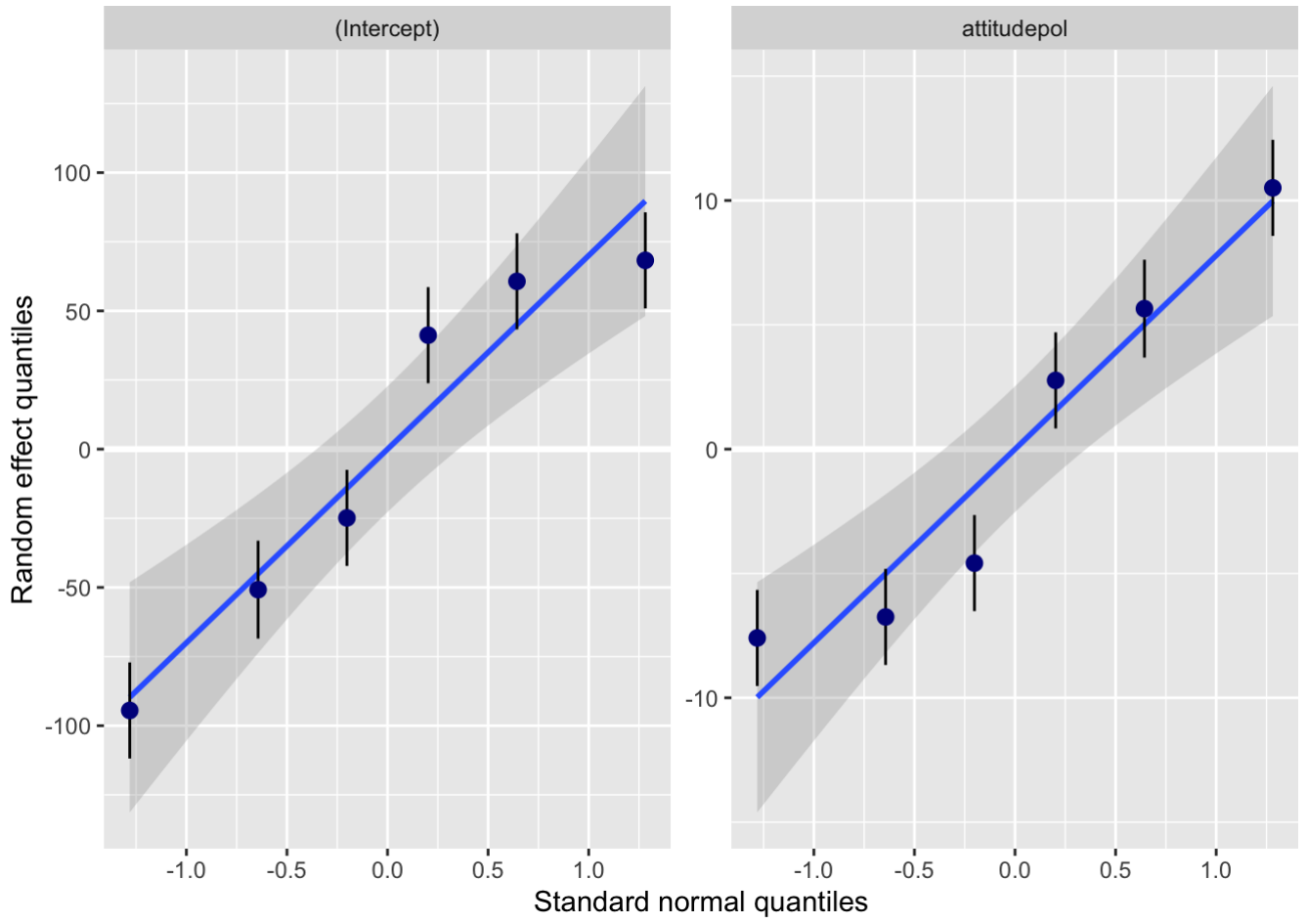

```
##
## [[2]]
## [[2]]$scenario
```

```
## `geom_smooth()` using formula 'y ~ x'
```

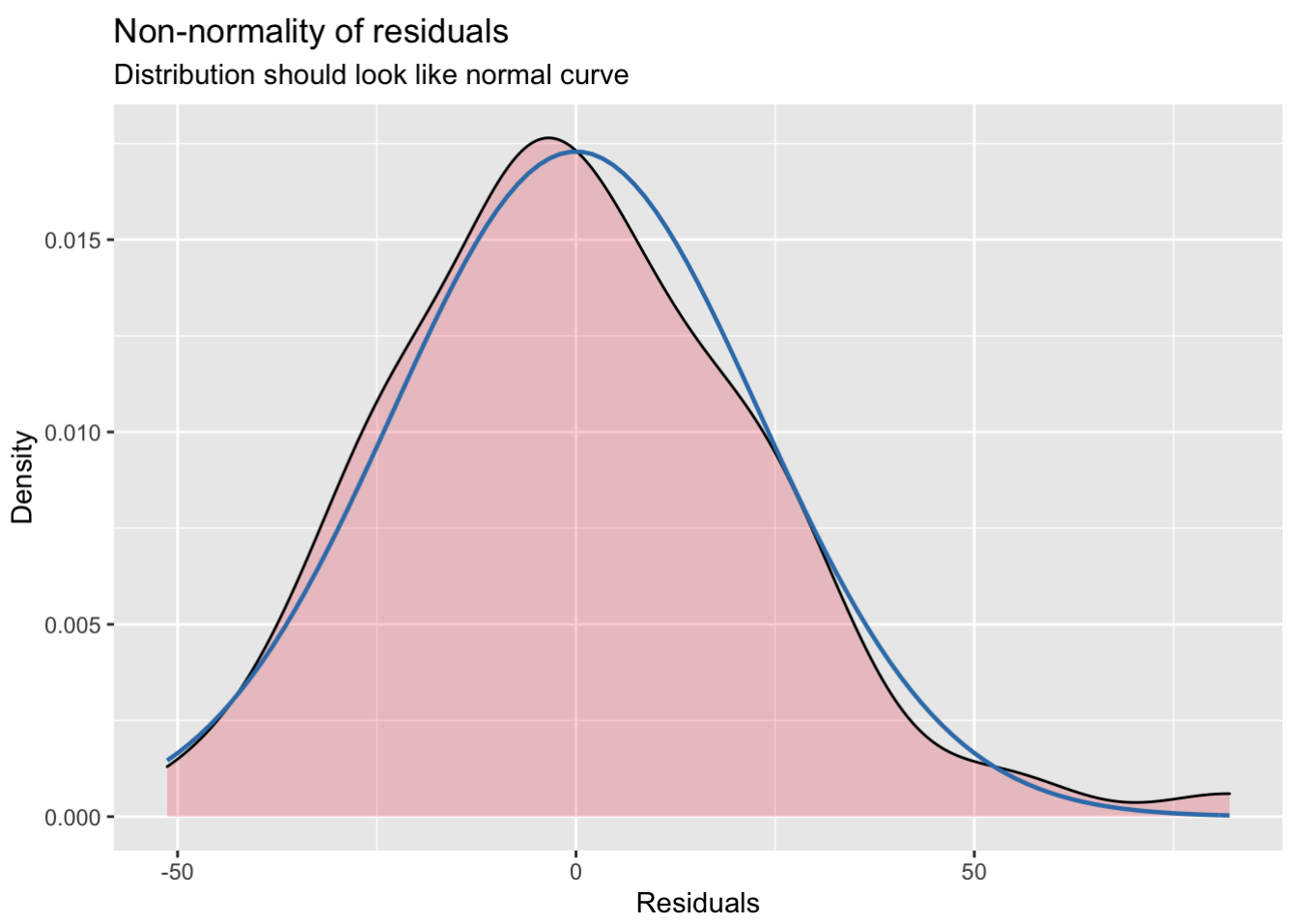



```
##
## [[2]]$subject
```

```
## `geom_smooth()` using formula 'y ~ x'
```

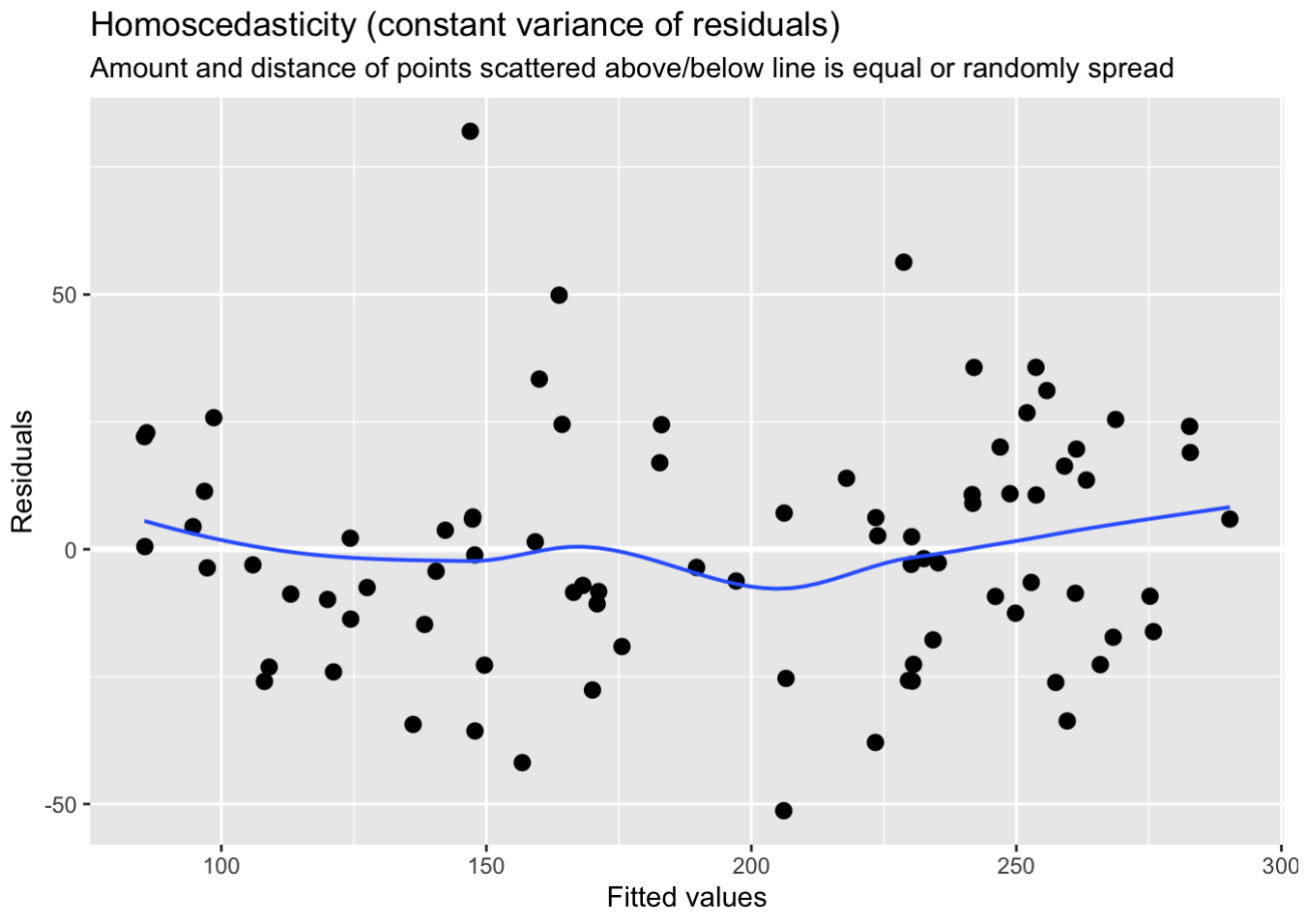



```
##
##
## [[3]]
```




```
##
## [[4]]
```

```
## `geom_smooth()` using formula 'y ~ x'
```

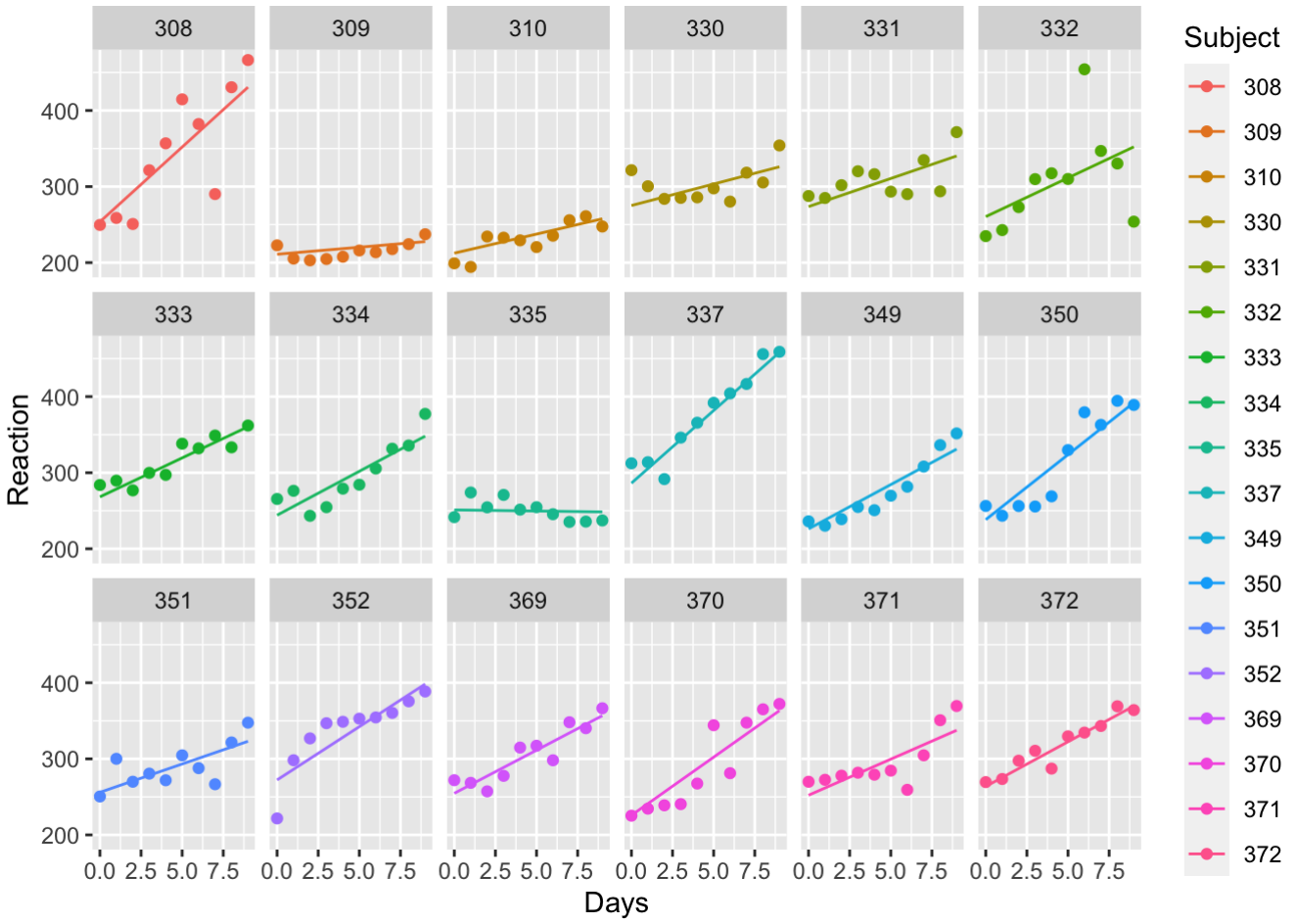


Visualize predictions of MEM

Below is an example of how can we visualize the predictions of MEMs. This example was taken from here.

In this case two parameters (the intercept and the slope of the deprivation effect) will be allowed to vary between the subject and one can plot the different fitted regression lines for each subject:

```
#load example data
data("sleepstudy")

#fit the model
m_slp <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)

#the next line put all the estimated intercept and slope per subject into a dataframe
reaction_slp <- as.data.frame(t(apply(ranef(m_slp)$Subject, 1,function(x) fixef(m_slp) + x)))

#to get the predicted regression lines we need one further step, writing the linear equation: Intercept +
pred_slp <- melt(apply(reaction_slp,1,function(x) x[1] + x[2]*0:9), value.name = "Reaction")

#some re-formatting for the plot
names(pred_slp)[1:2] <- c("Days","Subject")
pred_slp$Days <- pred_slp$Days - 1
pred_slp$Subject <- as.factor(pred_slp$Subject)

#plot with actual data
ggplot(pred_slp,aes(x=Days,y=Reaction,color=Subject))+
  geom_line()+
  geom_point(data=sleepstudy,aes(x=Days,y=Reaction))+
  facet_wrap(~Subject,nrow=3)
```



Check all the predictors

But, if you remember the effect of gender from the previous post, you'd know that it is significant and it has a lot of variation (simply because the frequency of males and females are very different). Thus, let's make our model as complex as we can by adding not only gender as a fixed effect to it, but also a random slopes of gender for both random effects - subject and scenario.

Particularly: **We model the *voice frequency* of people as a function of *attitude* and *gender*, knowing that people (subjects) and situations (scenarios) influence *voice frequency* baselines and that the relationship may vary depending on person and situation.**

```
politeness.full = lmer(frequency ~ attitude   + gender +
                         (1+attitude|subject) + (1+attitude|scenario)+
                         (1+gender|subject)   + (1+gender|scenario),
                       data=politeness, REML=FALSE)
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(politeness.full)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: frequency ~ attitude + gender + (1 + attitude | subject) + (1 +
##     attitude | scenario) + (1 + gender | subject) + (1 + gender |
##     scenario)
##    Data: politeness
##
##      AIC      BIC   logLik deviance df.resid
##    822.3    861.0   -395.2    790.3       67
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1080 -0.5692 -0.0579  0.3314  3.5910
##
## Random effects:
##  Groups     Name        Variance  Std.Dev. Corr
##  scenario   (Intercept) 3.589e+02 18.94577
##             genderM     1.000e+02  9.99980 -1.00
##  scenario.1 (Intercept) 3.521e+00  1.87631
##             attitudepol 4.198e+01  6.47919 -1.00
##  subject    (Intercept) 1.818e-03  0.04264
##             genderM     6.202e+02 24.90417 0.73
##  subject.1  (Intercept) 8.702e+01  9.32861
##             attitudepol 2.863e+00  1.69215 1.00
##  Residual               5.941e+02 24.37476
## Number of obs: 83, groups:  scenario, 7; subject, 6
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  257.572     10.224    6.558  25.193 8.98e-08 ***
## attitudepol  -19.622      5.929    7.000  -3.310  0.01295 *
## genderM     -109.969     17.831    4.434  -6.167  0.00248 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) atttdp
## attitudepol -0.221
## genderM     -0.413  0.003
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
report(politeness.full)
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Random effect variances not available. Returned R2 does not account for random effects.
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Random effect variances not available. Returned R2 does not account for random effects.
## Random effect variances not available. Returned R2 does not account for random effects.
```

```
## boundary (singular) fit: see ?isSingular
```

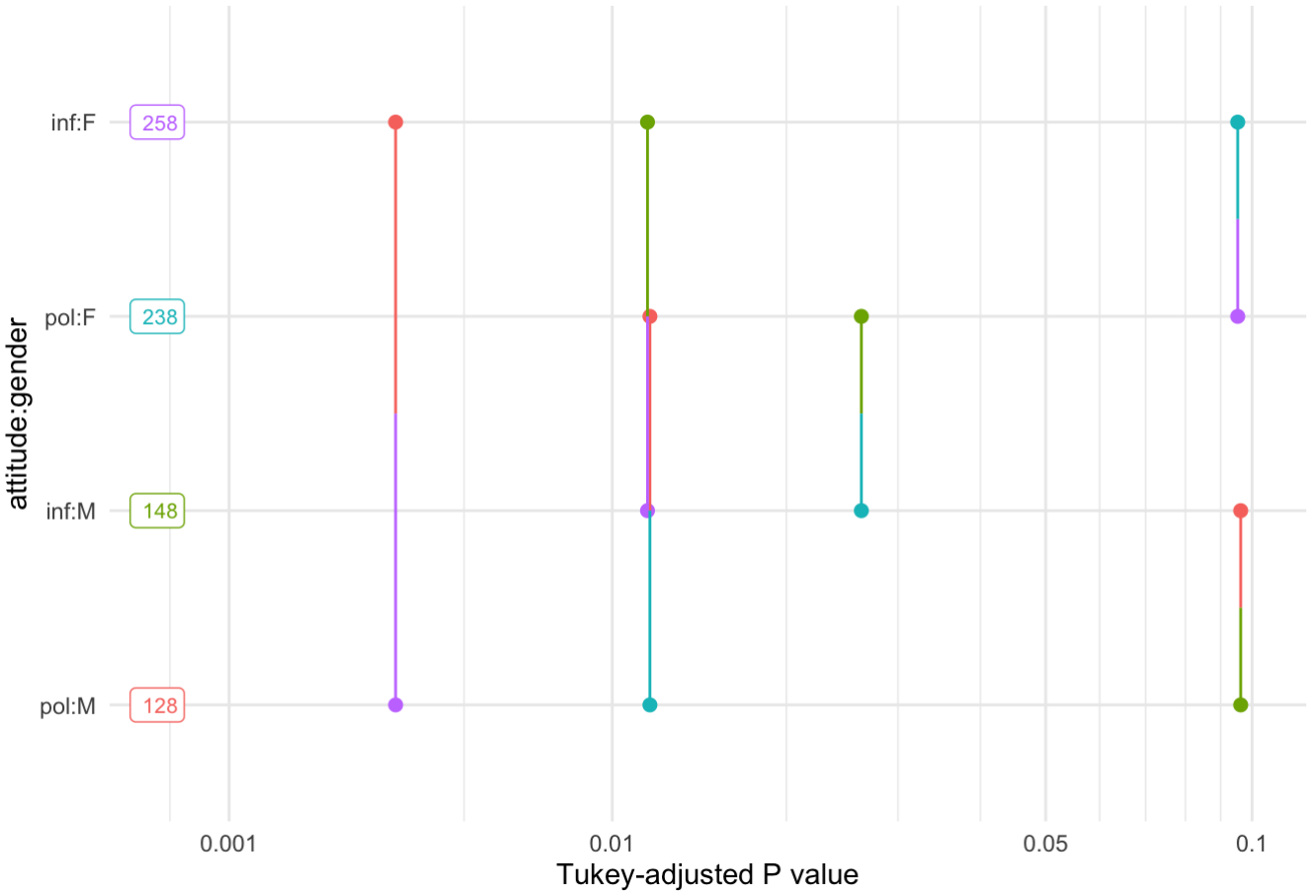
```
## We fitted a linear mixed model (estimated using ML and nloptwrap optimizer) to predict frequency with a
##
##   - The effect of attitudepol is negative and can be considered as small and significant (beta = -19.62
##   - The effect of genderM is negative and can be considered as very large and significant (beta = -109.
##   - The effect of R2 (conditional) is NA and can be considered as very large and not significant (beta
```

Post-Hos analysis

```
# post-hoc for MEMs
emm <- emmeans(politeness.full, pairwise ~ attitude * gender, adjust = "Bonferroni")
emm
```

```
## $emmeans
##  attitude gender emmean   SE   df lower.CL upper.CL
##  inf      F         258 12.2 7.55    217.8      297
##  pol      F         238 12.6 8.21    197.7      278
##  inf      M         148 20.6 4.84     68.0      227
##  pol      M         128 20.9 5.15     49.6      206
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 4 estimates
##
## $contrasts
##  contrast      estimate    SE   df t.ratio p.value
##  inf,F - pol,F     19.6  6.41 4.96 3.063   0.1699
##  inf,F - inf,M    110.0 23.63 6.45 4.654   0.0173
##  inf,F - pol,M    129.6 24.50 7.86 5.290   0.0047
##  pol,F - inf,M     90.3 24.47 7.78 3.692   0.0385
##  pol,F - pol,M    110.0 23.63 6.45 4.654   0.0173
##  inf,M - pol,M     19.6  6.41 4.96 3.063   0.1699
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: bonferroni method for 6 tests
```

```
pwpp(emmeans(politeness.full, ~ attitude * gender), type = "response")+theme_minimal()
```



Choose the final (best) model

Let’s now do the likelihood ratio test to see which model is better:

```
anova(politeness.full,politeness.model)
```

```
## Data: politeness
## Models:
## politeness.model: frequency ~ attitude + (attitude | subject) + (attitude | scenario)
## politeness.full: frequency ~ attitude + gender + (1 + attitude | subject) + (1 +
## politeness.full:     attitude | scenario) + (1 + gender | subject) + (1 + gender |
## politeness.full:     scenario)
##                  npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## politeness.model    9 823.32 845.09 -402.66   805.32
## politeness.full    16 822.30 861.00 -395.15   790.30 15.021  7    0.03573 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Similarly to the **intercept only model** in the previous post, the **random slope model** with *gender* is also significantly better, then without gender. Moreover, the model with random slopes is also significantly better then the model with only random intercepts (not shown). Which makes sense, since the

literature suggests, that Intercept only models are less conservative then random slopes models and have a higher Type I error rate, meaning that they find more significant results then in reality exist (Schielzeth & Forstmeier, 2009, Barr, Levy, Scheepers, & Tilly, 2013).

Thus, **include random slopes for all fixed effects that turned out to be important in your study.**

However, the more complex your MEM becomes, the more often you'll get warnings. Some warnings can be ignored, some not. One such message notes that the convergence code is 1 and/or specifies that the "maximum number of function evaluations was exceeded". In such case you can simply increase the maximum number of function evaluations and rerun the model. The following code increases the maximum number of iterations to 1 million: `control = lmerControl(optCtrl = list(maxfun = 1e6))`.

When NOT to use Mixed Effects Models

- in simple cases where the assumptions of Repeated Measures ANOVA hold
- when the assumptions are not satisfied

Assumptions

- normality
- linearity
- collinearity
- heteroscedasticity
- influential points and outliers

What's next

- Midex Effects Model 4: GLMM

Further readings and references

- Makowski, (2018). The psycho Package: an Efficient and Publishing-Oriented Workflow for Psychological Science. Journal of Open Source Software, 3(22), 470. https://doi.org/10.21105/joss.00470
- Good practical guide with exercises http://coltekin.net/cagri/R/r-exercisesse12.html and the solutions to these exercises http://coltekin.net/cagri/R/r-exercisesse13.html#x20-74000A.11
- amazingly well written article!!!: https://ourcodingclub.github.io/2017/03/15/mixed-models.html#FERE
- Douglas M. Bates paper on his `lme4` package: Fitting Linear Mixed-Effects Models Using lme4: https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf
- Douglas M. Bates book about `lme4` package: http://lme4.r-forge.r-project.org/book/
- Thanks to Bodo Winter for the data! http://www.bodowinter.com/tutorial/bw_LME_tutorial2.pdf
- How to visualize predictions of MEMs: https://biologyforfun.wordpress.com/2017/04/03/interpreting-random-effects-in-linear-mixed-effect-models/
- Pseudoreplication and the Design of Ecological Field Experiments. Stuart H. Hurlbert. First published: February 1984 https://doi.org/10.2307/1942661
- https://ase.tufts.edu/gsc/gradresources/guidetomixedmodelsinr/mixed%20model%20guide.html

statistics linear regression repeated measures

Yury Zablotski

Data Scientist at LMU Munich, Faculty of Veterinary Medicine

Passion for applying Biostatistics and Machine Learning to Life Science Data

Related

- Statistical tests vs. linear regression
- Model diagnostics
- Multiple linear regression
- Constraints for linear regression
- Simple linear regression

0 Comments Login ▾

G Start the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data

comments powered by Disqus

Powered by the Academic theme for Hugo.