LMX Example

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

I would do the package installation outside of the R markdown file. Running this document will take while, because it estimates all LMX relevant models that were discussed in the manuscript. We recommend looking at the pdf corresponding to this document (LMX_Example.pdf). Also, you will need to have Rstan installed (see RstanInstallation.R instructions). This Rmarkdown file was provided to illustrate how the pdf was made and to provide an illustation of the information that R/brms/stan calculates for you.

```
  ## Install packages
      # install.packages("brms")
      # install.packages("mvtnorm")
      # install.packages("lme4")
      # install.packages("devtools")
      # install.packages("ggplot2")
      # install.packages("processx")
      # install.packages("bayesplot")
      # devtools::install_github("mvuorre/brmstools")


  ## Loads the packages
      library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: ggplot2
```

```
## Loading 'brms' package (version 2.7.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
## Run theme_set(theme_default()) to use the default bayesplot theme.
```

```
      library(mvtnorm)
      library(lme4)
```

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

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:brms':
##
##     ngrps
```

```
      library(bayesplot)
```

```
## This is bayesplot version 1.6.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(ggplot2)
sessionInfo() #Presents version information
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17134)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
## [1] bayesplot_1.6.0 lme4_1.1-19     Matrix_1.2-15   mvtnorm_1.0-8  
## [5] brms_2.7.0      ggplot2_3.1.0   Rcpp_1.0.0     
## 
## loaded via a namespace (and not attached):
##  [1] Brobdingnag_1.2-5    splines_3.5.2        gtools_3.5.0        
##  [4] StanHeaders_2.18.0-1 threejs_0.3.1        shiny_1.1.0         
##  [7] assertthat_0.2.0     stats4_3.5.2         yaml_2.1.19         
## [10] pillar_1.3.1         backports_1.1.2      lattice_0.20-38     
## [13] glue_1.3.0           digest_0.6.15        promises_1.0.1      
## [16] minqa_1.2.4          colorspace_1.3-2     htmltools_0.3.6     
## [19] httpuv_1.4.3         plyr_1.8.4           dygraphs_1.1.1.4    
## [22] pkgconfig_2.0.2      rstan_2.18.2         purrr_0.2.5         
## [25] xtable_1.8-2         scales_0.5.0         processx_3.1.0      
## [28] later_0.7.2          tibble_2.0.1         DT_0.4              
## [31] withr_2.1.2          shinyjs_1.0          lazyeval_0.2.1      
## [34] cli_1.0.1            magrittr_1.5         crayon_1.3.4        
## [37] mime_0.5             evaluate_0.10.1      MASS_7.3-51.1       
## [40] nlme_3.1-137         xts_0.10-2           pkgbuild_1.0.2      
## [43] colourpicker_1.0     prettyunits_1.0.2    rsconnect_0.8.8     
## [46] tools_3.5.2          loo_2.0.0            matrixStats_0.53.1  
## [49] stringr_1.3.1        munsell_0.4.3        bindrcpp_0.2.2      
## [52] callr_2.0.4          compiler_3.5.2       rlang_0.3.1         
## [55] nloptr_1.0.4         grid_3.5.2           ggridges_0.5.1      
## [58] htmlwidgets_1.2      crosstalk_1.0.0      igraph_1.2.1        
## [61] miniUI_0.1.1         base64enc_0.1-3      rmarkdown_1.10      
## [64] gtable_0.2.0         inline_0.3.14        abind_1.4-5         
## [67] markdown_0.8         reshape2_1.4.3       R6_2.2.2            
## [70] gridExtra_2.3        rstantools_1.5.0     zoo_1.8-1           
## [73] knitr_1.20           bridgesampling_0.4-0 dplyr_0.7.8         
## [76] bindr_0.1.1          shinystan_2.5.0      shinythemes_1.1.1   
## [79] rprojroot_1.3-2      stringi_1.2.2        parallel_3.5.2      
## [82] tidyselect_0.2.5     coda_0.19-1
```

There are additional comments inside the code if we felt that it would be beneficial to have the comment close to the code. The set.seed function makes the results reproducible. You should get the same total number of followers (1735), mean group size (6.94), min group size (3) and max group size (10).

```
# Cross-sectional data generation ----
  ## Level 2 information - Allowing for unequal work group sizes
set.seed(1919)
Lev2N <- 250
with_sample_cross <- rbinom(n = Lev2N, prob = 0.7, size = 10)
sum(with_sample_cross) ####Total number of people (followers)
```

```
## [1] 1735
```

```
mean(with_sample_cross) #Mean sample size
```

```
## [1] 6.94
```

```
min(with_sample_cross)  #Minimum sample size
```

```
## [1] 3
```

```
max(with_sample_cross)  #Maximum smaple size
```

```
## [1] 10
```

```
### Creating a group id variable to correspond to the number of individuals on each team.
group_id <- rep(1:Lev2N, times = with_sample_cross)
```

In the next code chunk, we are setting up the data generating values for the location side of the model. All exogenous predictor variables were generated to have a total variance of one and are assumed to be uncorrelated with one another. This was done so the effect sizes (correlations) from the meta-analysis can be best incorporated into the simulation. Similarty (i.e., a Level 1 predictor) is generated to have 20% of its variability between groups and 80% of it within group. The correlation was converted into an r squared and it is assumed that the proportion of variability that similarity explains is the same at both levels (i.e., .5^2).Transformational leadership is assumed to explain an additional 53.29% of the variability in between group LMX. The cross-level interaction was specified in such a way that the effect for level-one similarity is 0 for those in a group with a highly transformational leader (score of 2 standard deviations above the mean). Overall Level 1 variability is 1. Overall between variability is .25 for an ICC of .25/(1.25) = .2. However, it is important to note that this ICC is for a group that is in a group with an average similarity, an individual with that particuar similarity value, and an average level of transformational leadership. In other words, when all predictor variables are zero, because there is also the scale side of the model.

```
similar_L1_var <- .8
similar_L2_var <- .2
trans_tot_var <- 1

#Location intercept
gamma_00 = 0

#Location - simple main effect for similarity within-group effect
#(value converted from meta-analysis)
gamma_10 = sqrt(5)/4

#Location main effect for Level-two similarity (value converted from meta-analysis)
gamma_01 = sqrt(5)/4

#Location transformational leadership simple main effect
gamma_02 = sqrt(.73^2/4)
```

```
#Location interaction term which is created to
#"shut off" the similarity level one effect for
#Those with high transformational leadership
gamma_11 = -gamma_10/2

#Explains 25% of the level 1 variance
avg_res_L1var = 1 - gamma_10^2*similar_L1_var

#Explains 78.29% of random intercept variance
res_RIvar = .25 - gamma_01^2*similar_L2_var -gamma_02^2*trans_tot_var

#Explains 62.5% of the random slope variance.
res_RSvar = .125 - (-gamma_10/2)^2
```

In this code chunk, we set up the scale side of the model. There is not much to go on as far as prior research regarding these data generating values. Thus, they were chosen to align with our hypothetical substantive scenario and to provide an unconditional ICC that is aligned with what is typically found in prior research (i.e., somewhere between .15 and .30, Mathieu, J. E., Aguinis, H., Culpepper, S. A., & Chen, 2012).

```
#Scale fixed intercept log(sqrt(avg_resid_VAR))
tau_00 = log(sqrt(avg_res_L1var))

#Scale - fixed Level-one similarity simple main effect
tau_10 = -.15

#Scale main effect for Level-two similarity (value converted from meta-analysis)
tau_01 = -.15

#Scale transformative leadership simple main effect
tau_02 = -.10

#Scale interaction term which is created to "shut off" the similarity level one effect
#for those with high transformational leadership
tau_11 = .075

#Log residual variances
res_RIvar_sigma <- .01
res_RSvar_sigma <- .01

#### Used to fill in the covariance matrix
off_diag_b0_t0 <- -.3*sqrt(res_RIvar)*sqrt(res_RIvar_sigma)
off_diag_b0_b1 <- -.3*sqrt(res_RIvar)*sqrt(res_RSvar)
off_diag_t0_t1 <- -.3*sqrt(res_RIvar_sigma)*sqrt(res_RSvar_sigma)
off_diag_t0_b1 <- .3*sqrt(res_RIvar_sigma)*sqrt(res_RSvar)
off_diag_t1_b1 <- -.3*sqrt(res_RSvar_sigma)*sqrt(res_RSvar)
off_diag_b0_t1 <- 0

#### G matrix
mus <- c(0, 0, 0, 0)
sigma <- matrix(data = c(res_RIvar, off_diag_b0_t0 , off_diag_b0_t1, off_diag_b0_b1,
                         off_diag_b0_t0, res_RIvar_sigma, off_diag_t0_t1, off_diag_t0_b1,
                         off_diag_b0_t1, off_diag_t0_t1, res_RSvar_sigma, off_diag_t1_b1,
                         off_diag_b0_b1, off_diag_t0_b1, off_diag_t1_b1, res_RSvar),
                nrow = 4, ncol = 4)
```

```r
G_corr <- cov2cor(sigma)
colnames(G_corr) <- c("u0j", "v0j", "v1j", "u1j")
row.names(G_corr) <- c("u0j", "v0j", "v1j", "u1j")
```

```r
# Generating Level-two data.
set.seed(5189)
lev2_rans <- rmvnorm(n = Lev2N, mus, sigma) # u's and v's from equations 1 and 2
trans <- rnorm(Lev2N,mean = 0,sd = 1) #Transformative leadership
L2_similar <- rnorm(Lev2N,mean = 0,sd = sqrt(.2)) # Level 2 similarity uncentered

### Accounting for unbalanced sample size.
trans_unbal <- rep(trans, with_sample_cross)

# Transformative leadership grand mean centered.
trans_gmc <- trans - mean(trans_unbal)

#Level two similarity unbalanced
similar_L2_unbal <- rep(L2_similar, with_sample_cross)




# Creating an empty list of lists to hold the generated data for each group
data_holder_cross_list <-  rep( list(list()), Lev2N )
# Using a for loop to generate the level-one data for each team and
#then bind all 250 groups of the data together

for (GroupID in 1:Lev2N){
# Generate Standard Normal Variates with Specified Correlation

  u0j <- lev2_rans[GroupID, 1]
  v0j <- lev2_rans[GroupID, 2]
  v1j <- lev2_rans[GroupID, 3]
  u1j <- lev2_rans[GroupID, 4]

# Creating the level one similarity variable with the specified correlation.
  similar <- L2_similar[GroupID] + rnorm(n = with_sample_cross[GroupID],mean = 0,
                                         sd = sqrt(.8))
  IndID <- 1:with_sample_cross[GroupID]
   # Similar level 1 group mean centered
   similar_l1_grmc <- similar - L2_similar[GroupID]
   #-# or
   # similar_l1_grmc <- similar - mean(similar)


   log_sigma_ij = (tau_00 + v0j) + tau_01*L2_similar[GroupID] +
                 tau_02*trans_gmc[GroupID] +
                 (tau_10 + tau_11*trans_gmc[GroupID] + v1j)*similar_l1_grmc
   sigma_ij = exp(log_sigma_ij)
   e_ij = rnorm(with_sample_cross[GroupID],mean = 0,sd = sigma_ij)
   # Create Outcome Variable
   Y = (gamma_00 + u0j) + gamma_01*L2_similar[GroupID] +
        gamma_02*trans_gmc[GroupID] +
       (gamma_10 + gamma_11*trans_gmc[GroupID] + u1j)*similar_l1_grmc + e_ij
   data_holder_cross_list[[GroupID]] =
     data.frame(GroupID, IndID, group_size = with_sample_cross[GroupID],
                transform_grandmc = trans_gmc[GroupID], similar,
                similarL1_groupmc = similar_l1_grmc,
                similarL2 = L2_similar[GroupID],
                tau_00, v0j, tau_01, tau_10, tau_11, v1j,
                gamma_00, u0j, gamma_01, gamma_10,
                gamma_11, u1j,LMX = Y, sigma_ij, e_ij)
      }

# Binding all 250 groups of the data together
  data_holder_cross_centered <- do.call(rbind, data_holder_cross_list)
  data_holder_cross_centered <- as.data.frame(data_holder_cross_centered)

#Centering the tranformational leadership variable at two
#standard deviations above the mean to get the effect of
#of similiarity specifically for very high transformational leaders.


data_holder_cross_centered$transform_c2 <-
    data_holder_cross_centered$transform_grandmc - 2
```

The data are now generated. The name of the final dataset is in the data_holder_cross_centered data frame. Estimating the models occurs below this point. The models estimated in this manuscript utilize the default priors in brms (see Burkner, 2017); they are described in the manuscript. We also demonstrate how the prior could be changed if so desired.

This document demonstrates several different versions of the mixed-effects location-scale model that could be estimated for the cross-sectional LMX example. We recommend using the "eval = FALSE" option for some code chunks (see the OtherModels section), because these models can take some time to estimate. Also, within a chunk any model can be commented out by using a hashtag. The version of this file posted online commented out the alternative prior specification for the full model. The first chunk labeled "EstManuscriptmodels" short for estimating the manuscript models contains an unconditional random intercept model to obtain the estimated ICC using the lme4 (Bates et al. 2014) package. Restricted maximum likelihood estimation was used to obtain the estimates. This was done to confirm that our data generation has some correspondence with multilevel data typically found in the organizational sciences (i.e., somewhere between .15 and .30, Mathieu, J. E., Aguinis, H., Culpepper, S. A., & Chen, 2012). That was the case, the ICC was approximately .2036 = 0.3066/(0.3066 + 1.199). The second model estimated in this chunk is the full model that corresponds to the data generation. This is the model that was discussed at length in the illustrative examples portion of the manuscript. The next model provided in the chunk, is the same model with a different prior. It is currently commented out; remove the hashtag and that model can be estimated. The final model that is included in this chunk was estimated to center the transformational leadership variable at two standard deviations above the mean which allows us to interpret the simple main effect of relative similarity (level-one) for groups with leaders that are mucher higher than the mean. These results were mentioned in the manuscript.

```
# Estimating Models of Interest
# Unconditional ICC
unc_ICC <- lmer(LMX  ~ 1 + (1 | GroupID), data = data_holder_cross_centered)
summary(unc_ICC)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: LMX ~ 1 + (1 | GroupID)
##    Data: data_holder_cross_centered
##
## REML criterion at convergence: 5494.8
```

```
##
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -4.9941 -0.5622  0.0916  0.6184  2.9213 
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  GroupID  (Intercept) 0.3066   0.5537  
##  Residual             1.1990   1.0950  
## Number of obs: 1735, groups:  GroupID, 250
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -0.09924    0.04403  -2.254
```

```
# Full Model that matches the data generation model
cross_LMX_full <-
bf(LMX ~ 1  + similarL1_groupmc + similarL2 + transform_grandmc +
          similarL1_groupmc:transform_grandmc + (1 + similarL1_groupmc|ID1|GroupID),
  sigma ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc +
          similarL1_groupmc:transform_grandmc + (1 + similarL1_groupmc|ID1|GroupID))

#Estimating the model with brms
 cross_LMX_full_out <- brm(cross_LMX_full,
   data = data_holder_cross_centered, chains = 4, cores = 4, iter = 5000,
   control = list(adapt_delta = .85, max_treedepth = 15), seed = 1848, save_all_pars = T)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
summary(cross_LMX_full_out)
```

```
##  Family: gaussian 
##   Links: mu = identity; sigma = log 
## Formula: LMX ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc + similarL1_groupmc:transform_g
##          sigma ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc + similarL1_groupmc:transform_
##    Data: data_holder_cross_centered (Number of observations: 1735) 
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects: 
## ~GroupID (Number of levels: 250) 
##                                                Estimate Est.Error l-95% CI
## sd(Intercept)                                      0.25      0.03     0.18
## sd(similarL1_groupmc)                              0.22      0.04     0.14
## sd(sigma_Intercept)                                0.09      0.04     0.01
## sd(sigma_similarL1_groupmc)                        0.05      0.03     0.00
## cor(Intercept,similarL1_groupmc)                  -0.24      0.20    -0.60
## cor(Intercept,sigma_Intercept)                     0.02      0.32    -0.60
## cor(similarL1_groupmc,sigma_Intercept)             0.18      0.36    -0.59
## cor(Intercept,sigma_similarL1_groupmc)             0.25      0.41    -0.65
## cor(similarL1_groupmc,sigma_similarL1_groupmc)    -0.06      0.41    -0.81
## cor(sigma_Intercept,sigma_similarL1_groupmc)       0.05      0.44    -0.78
##                                                u-95% CI Eff.Sample Rhat
## sd(Intercept)                                      0.32       3404 1.00
## sd(similarL1_groupmc)                              0.30       2579 1.00
## sd(sigma_Intercept)                                0.17       1028 1.00
## sd(sigma_similarL1_groupmc)                        0.12       2727 1.00
## cor(Intercept,similarL1_groupmc)                   0.19       2250 1.00
## cor(Intercept,sigma_Intercept)                     0.67       4512 1.00
## cor(similarL1_groupmc,sigma_Intercept)             0.81       2957 1.00
## cor(Intercept,sigma_similarL1_groupmc)             0.89       6882 1.00
## cor(similarL1_groupmc,sigma_similarL1_groupmc)     0.74       7822 1.00
## cor(sigma_Intercept,sigma_similarL1_groupmc)       0.82       6661 1.00
##
## Population-Level Effects:
##                                           Estimate Est.Error l-95% CI
## Intercept                                    -0.05      0.03    -0.11
## sigma_Intercept                              -0.14      0.02    -0.19
## similarL1_groupmc                             0.57      0.03     0.51
## similarL2                                     0.64      0.06     0.53
## transform_grandmc                             0.37      0.03     0.32
## similarL1_groupmc:transform_grandmc          -0.27      0.03    -0.33
## sigma_similarL1_groupmc                      -0.17      0.03    -0.22
## sigma_similarL2                              -0.16      0.04    -0.25
## sigma_transform_grandmc                      -0.07      0.02    -0.10
## sigma_similarL1_groupmc:transform_grandmc     0.10      0.02     0.06
##                                           u-95% CI Eff.Sample Rhat
## Intercept                                    -0.00       9064 1.00
## sigma_Intercept                              -0.10       3610 1.00
## similarL1_groupmc                             0.63       8724 1.00
## similarL2                                     0.76       9028 1.00
## transform_grandmc                             0.42       7802 1.00
## similarL1_groupmc:transform_grandmc          -0.21       9250 1.00
## sigma_similarL1_groupmc                      -0.12       5474 1.00
## sigma_similarL2                              -0.07      11643 1.00
## sigma_transform_grandmc                      -0.03       8952 1.00
## sigma_similarL1_groupmc:transform_grandmc     0.15       9428 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# Full Model that matches the data generation model with an informative prior
# informative <- c(prior(normal(0, 1), class = "b"),
                 # prior(normal(0, 1), class = "b", dpar = "sigma"),
                 # prior(lkj(10), class = "cor"))

#No need to redefine the same equation.

# cross_def_full_info_out <- brm(cross_LMX_full, data = data_holder_cross_centered,
                               # chains = 4, cores = 4, iter = 5000,
                              # control = list(adapt_delta = .85, max_treedepth = 15),
                              # prior = informative,
                              # seed = 14, save_all_pars = T)
# summary(cross_def_full_info_out)


#Transformative leadership centered at two
```

```r
#standard deviations above the mean

cross_LMX_full_transc2 <-
bf(LMX ~ 1  + similarL1_groupmc + similarL2 + transform_c2 +
          similarL1_groupmc:transform_c2 + (1 + similarL1_groupmc|ID1|GroupID),
  sigma ~ 1 + similarL1_groupmc + similarL2 + transform_c2 +
          similarL1_groupmc:transform_c2 + (1 + similarL1_groupmc|ID1|GroupID))

#Estimating the model with brms
 cross_LMX_full_transc2_out <- brm(cross_LMX_full_transc2,
   data = data_holder_cross_centered, chains = 4, cores = 4, iter = 5000,
   control = list(adapt_delta = .85, max_treedepth = 15), seed = 1954, save_all_pars = T)
```

```
## Compiling the C++ model
```

```
## recompiling to avoid crashing R session
```

```
## Start sampling
```

```r
summary(cross_LMX_full_transc2_out)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = log
## Formula: LMX ~ 1 + similarL1_groupmc + similarL2 + transform_c2 + similarL1_groupmc:transform_c2 + (
##          sigma ~ 1 + similarL1_groupmc + similarL2 + transform_c2 + similarL1_groupmc:transform_c2 +
##    Data: data_holder_cross_centered (Number of observations: 1735)
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~GroupID (Number of levels: 250)
##                                                Estimate Est.Error l-95% CI
## sd(Intercept)                                      0.25      0.03     0.18
## sd(similarL1_groupmc)                              0.22      0.04     0.14
## sd(sigma_Intercept)                                0.09      0.04     0.01
## sd(sigma_similarL1_groupmc)                        0.04      0.03     0.00
## cor(Intercept,similarL1_groupmc)                  -0.24      0.20    -0.59
## cor(Intercept,sigma_Intercept)                     0.02      0.31    -0.61
## cor(similarL1_groupmc,sigma_Intercept)             0.19      0.35    -0.55
## cor(Intercept,sigma_similarL1_groupmc)             0.23      0.42    -0.68
## cor(similarL1_groupmc,sigma_similarL1_groupmc)    -0.06      0.42    -0.80
## cor(sigma_Intercept,sigma_similarL1_groupmc)       0.04      0.44    -0.79
##                                                u-95% CI Eff.Sample Rhat
## sd(Intercept)                                      0.32       3258 1.00
## sd(similarL1_groupmc)                              0.30       2461 1.00
## sd(sigma_Intercept)                                0.17       1153 1.00
## sd(sigma_similarL1_groupmc)                        0.12       2503 1.00
## cor(Intercept,similarL1_groupmc)                   0.17       2525 1.00
## cor(Intercept,sigma_Intercept)                     0.61       4452 1.00
## cor(similarL1_groupmc,sigma_Intercept)             0.81       2790 1.00
## cor(Intercept,sigma_similarL1_groupmc)             0.88       7149 1.00
## cor(similarL1_groupmc,sigma_similarL1_groupmc)     0.74       8685 1.00
## cor(sigma_Intercept,sigma_similarL1_groupmc)       0.82       6217 1.00
##
## Population-Level Effects:
```

```
##                                      Estimate Est.Error l-95% CI u-95% CI
## Intercept                                0.68      0.06     0.57     0.80
## sigma_Intercept                         -0.28      0.05    -0.37    -0.19
## similarL1_groupmc                        0.03      0.06    -0.10     0.15
## similarL2                                0.65      0.06     0.53     0.77
## transform_c2                             0.37      0.03     0.32     0.42
## similarL1_groupmc:transform_c2          -0.27      0.03    -0.33    -0.22
## sigma_similarL1_groupmc                  0.04      0.05    -0.06     0.14
## sigma_similarL2                         -0.16      0.04    -0.25    -0.07
## sigma_transform_c2                      -0.07      0.02    -0.10    -0.03
## sigma_similarL1_groupmc:transform_c2     0.10      0.02     0.06     0.15
##                                      Eff.Sample Rhat
## Intercept                                  7788 1.00
## sigma_Intercept                            6188 1.00
## similarL1_groupmc                          5671 1.00
## similarL2                                  8414 1.00
## transform_c2                               7258 1.00
## similarL1_groupmc:transform_c2             5819 1.00
## sigma_similarL1_groupmc                    5200 1.00
## sigma_similarL2                            9657 1.00
## sigma_transform_c2                         9747 1.00
## sigma_similarL1_groupmc:transform_c2       5966 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

This section is presented to demonstrate how the mixed-effect location-scale model is similar to yet distinct from traditional multilevel models. The models are presented in a build up fashion from simplest (i.e., the least number of parameters estimated) to the most complex (i.e., the most number of parameters estimated). Model comparison is also performed in this section. These models are described in detail in the manuscript. Frequentist versions of these models are presented until scale random effects are added to the model.

```r
#Equivalent to an empty linear regression model with no predictors.
empty_mod <- bf(LMX ~ 1 ,
                    sigma ~ 1 )

empty_mod_out <- brm(empty_mod, data = data_holder_cross_centered, chains = 4,
                     cores = 4, iter = 5000, control = list(adapt_delta = .85,
                                                            max_treedepth = 15),
                     seed = 8154, save_all_pars = T)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```r
print(summary(empty_mod_out), digits = 4)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = log
## Formula: LMX ~ 1
##          sigma ~ 1
##    Data: data_holder_cross_centered (Number of observations: 1735)
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
```

```
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample   Rhat
## Intercept         -0.0904    0.0295  -0.1499  -0.0317       6731 0.9997
## sigma_Intercept    0.2043    0.0171   0.1713   0.2381       6376 0.9999
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
#Matching results presented from R's general linear model function
empty_mod_freq <- lm(LMX ~ 1, data = data_holder_cross_centered)
summary(empty_mod_freq)
```

```
##
## Call:
## lm(formula = LMX ~ 1, data = data_holder_cross_centered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7478 -0.7059  0.0944  0.8220  3.3932
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.09033    0.02944  -3.068  0.00219 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.226 on 1734 degrees of freedom
```

```
#Random intercept on the location side
random_int <- bf(LMX ~ 1 + (1 | GroupID),
                 sigma ~ 1 )


random_int_out <- brm(random_int, data = data_holder_cross_centered,
                      chains = 4, cores = 4, iter = 5000,
                      control = list(adapt_delta = .85, max_treedepth = 15),
                      seed = 186, save_all_pars = T)
```

```
## Compiling the C++ model
## Start sampling
```

```
summary(random_int_out)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = log
## Formula: LMX ~ 1 + (1 | GroupID)
##          sigma ~ 1
##    Data: data_holder_cross_centered (Number of observations: 1735)
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~GroupID (Number of levels: 250)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.56      0.04     0.48     0.63       4279 1.00
```

```
##
## Population-Level Effects:
##                   Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept            -0.10      0.04    -0.19    -0.01       5696 1.00
## sigma_Intercept       0.09      0.02     0.06     0.13      11596 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
#Random Intercept frequentist version - REML estimation
#This model was the first one estimated in the EstManuscriptmodels chunk.
unc_ICC <- lmer(LMX  ~ 1 + (1 | GroupID), data = data_holder_cross_centered)
summary(unc_ICC)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: LMX ~ 1 + (1 | GroupID)
##    Data: data_holder_cross_centered
##
## REML criterion at convergence: 5494.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.9941 -0.5622  0.0916  0.6184  2.9213
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  GroupID  (Intercept) 0.3066   0.5537
##  Residual             1.1990   1.0950
## Number of obs: 1735, groups:  GroupID, 250
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -0.09924    0.04403  -2.254
```

```
# Random Intercept for location and scale
# From now on only the Bayesian approach will be estimated.
# The extra ID1 variable is a generic variable that is
# needed to accomodate random intercepts on the scale side
random_ints_loc_scale <- bf(LMX ~ 1   + (1 |ID1|GroupID),
                          sigma ~ 1 +  (1 |ID1|GroupID))

random_ints_loc_scale_out <- brm(random_ints_loc_scale,
                                 data = data_holder_cross_centered,
                                 chains = 4, cores = 4, iter = 5000,
                                 control = list(adapt_delta = .85,
                                                max_treedepth = 15),
                                 seed = 4789, save_all_pars = T)
```

```
## Compiling the C++ model
## Start sampling
```

```
#Random slope for location and random intercept for scale

randomSlopeLoc_RIscale <- bf(LMX ~ 1 + similarL1_groupmc +
                                (1 + similarL1_groupmc|ID1|GroupID),
                                sigma ~ 1 +  (1 |ID1|GroupID))

randomSlopeLoc_RIscale_out <- brm(randomSlopeLoc_RIscale,
                                  data = data_holder_cross_centered,
                                  chains = 4, cores = 4, iter = 5000,
                                  control = list(adapt_delta = .85,
                                                  max_treedepth = 15),
                                  seed = 4, save_all_pars = T)
```

```
## Compiling the C++ model
## Start sampling
```

```
summary(randomSlopeLoc_RIscale_out)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = log
## Formula: LMX ~ 1 + similarL1_groupmc + (1 + similarL1_groupmc | ID1 | GroupID)
##          sigma ~ 1 + (1 | ID1 | GroupID)
##    Data: data_holder_cross_centered (Number of observations: 1735)
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~GroupID (Number of levels: 250)
##                                        Estimate Est.Error l-95% CI
## sd(Intercept)                              0.51      0.03     0.45
## sd(similarL1_groupmc)                      0.37      0.04     0.29
## sd(sigma_Intercept)                        0.21      0.03     0.15
## cor(Intercept,similarL1_groupmc)          -0.76      0.09    -0.92
## cor(Intercept,sigma_Intercept)            -0.50      0.12    -0.74
## cor(similarL1_groupmc,sigma_Intercept)     0.36      0.17     0.02
##                                        u-95% CI Eff.Sample Rhat
## sd(Intercept)                              0.58       4300 1.00
## sd(similarL1_groupmc)                      0.44       3738 1.00
## sd(sigma_Intercept)                        0.27       3315 1.00
## cor(Intercept,similarL1_groupmc)          -0.57       2861 1.00
## cor(Intercept,sigma_Intercept)            -0.26       4617 1.00
## cor(similarL1_groupmc,sigma_Intercept)     0.69       2556 1.00
##
## Population-Level Effects:
##                   Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept            -0.06      0.04    -0.14     0.01       7399 1.00
## sigma_Intercept      -0.12      0.02    -0.17    -0.07       5484 1.00
## similarL1_groupmc     0.56      0.04     0.49     0.63       9264 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# Random slopes on both sides
randomSlope_both <- bf(LMX ~ 1  + similarL1_groupmc +
                         (1 + similarL1_groupmc|ID1|GroupID),
                         sigma ~ 1 + similarL1_groupmc +
                         (1 + similarL1_groupmc|ID1|GroupID))
```

```r
# Estimating the model with brms
randomSlope_both_out <- brm(randomSlope_both, data = data_holder_cross_centered,
                            chains = 4, cores = 4, iter = 5000,
                            control = list(adapt_delta = .85, max_treedepth = 15),
                            seed = 8712, save_all_pars = T)
```

```
## Compiling the C++ model

## recompiling to avoid crashing R session

## Start sampling

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.85 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```r
# Full Model that matches the data generation model
cross_LMX_full <-
bf(LMX ~ 1  + similarL1_groupmc + similarL2 + transform_grandmc +
          similarL1_groupmc:transform_grandmc +
          (1 + similarL1_groupmc|ID1|GroupID),
  sigma ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc +
          similarL1_groupmc:transform_grandmc +
          (1 + similarL1_groupmc|ID1|GroupID))

#Estimating the model with brms
cross_LMX_full_out <- brm(cross_LMX_full,
   data = data_holder_cross_centered, chains = 4, cores = 4, iter = 5000,
   control = list(adapt_delta = .85, max_treedepth = 15),
   seed = 1848, save_all_pars = T)
```

```
## Compiling the C++ model

## recompiling to avoid crashing R session

## Start sampling
```

```r
summary(cross_LMX_full_out)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = log
## Formula: LMX ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc + similarL1_groupmc:transform_g
##          sigma ~ 1 + similarL1_groupmc + similarL2 + transform_grandmc + similarL1_groupmc:transform_
##    Data: data_holder_cross_centered (Number of observations: 1735)
## Samples: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~GroupID (Number of levels: 250)
##                                                 Estimate Est.Error l-95% CI
## sd(Intercept)                                       0.25      0.03     0.18
## sd(similarL1_groupmc)                               0.22      0.04     0.14
## sd(sigma_Intercept)                                 0.09      0.04     0.01
## sd(sigma_similarL1_groupmc)                         0.05      0.03     0.00
## cor(Intercept,similarL1_groupmc)                   -0.24      0.20    -0.60
## cor(Intercept,sigma_Intercept)                      0.02      0.32    -0.60
## cor(similarL1_groupmc,sigma_Intercept)              0.18      0.36    -0.59
## cor(Intercept,sigma_similarL1_groupmc)              0.25      0.41    -0.65
## cor(similarL1_groupmc,sigma_similarL1_groupmc)    -0.06      0.41    -0.81
## cor(sigma_Intercept,sigma_similarL1_groupmc)        0.05      0.44    -0.78
##                                                u-95% CI Eff.Sample Rhat
## sd(Intercept)                                      0.32       3404 1.00
## sd(similarL1_groupmc)                              0.30       2579 1.00
## sd(sigma_Intercept)                                0.17       1028 1.00
## sd(sigma_similarL1_groupmc)                        0.12       2727 1.00
## cor(Intercept,similarL1_groupmc)                   0.19       2250 1.00
## cor(Intercept,sigma_Intercept)                     0.67       4512 1.00
## cor(similarL1_groupmc,sigma_Intercept)             0.81       2957 1.00
## cor(Intercept,sigma_similarL1_groupmc)             0.89       6882 1.00
## cor(similarL1_groupmc,sigma_similarL1_groupmc)     0.74       7822 1.00
## cor(sigma_Intercept,sigma_similarL1_groupmc)       0.82       6661 1.00
##
## Population-Level Effects:
##                                           Estimate Est.Error l-95% CI
## Intercept                                    -0.05      0.03    -0.11
## sigma_Intercept                              -0.14      0.02    -0.19
## similarL1_groupmc                             0.57      0.03     0.51
## similarL2                                     0.64      0.06     0.53
## transform_grandmc                             0.37      0.03     0.32
## similarL1_groupmc:transform_grandmc          -0.27      0.03    -0.33
## sigma_similarL1_groupmc                      -0.17      0.03    -0.22
## sigma_similarL2                              -0.16      0.04    -0.25
## sigma_transform_grandmc                      -0.07      0.02    -0.10
## sigma_similarL1_groupmc:transform_grandmc     0.10      0.02     0.06
##                                           u-95% CI Eff.Sample Rhat
## Intercept                                    -0.00       9064 1.00
## sigma_Intercept                              -0.10       3610 1.00
## similarL1_groupmc                             0.63       8724 1.00
## similarL2                                     0.76       9028 1.00
## transform_grandmc                             0.42       7802 1.00
## similarL1_groupmc:transform_grandmc          -0.21       9250 1.00
## sigma_similarL1_groupmc                      -0.12       5474 1.00
## sigma_similarL2                              -0.07      11643 1.00
## sigma_transform_grandmc                      -0.03       8952 1.00
## sigma_similarL1_groupmc:transform_grandmc     0.15       9428 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```