Online Appendix A: *r2MLM* R function

r2MLM R function Description:

This function reads in raw data and multilevel model (MLM) parameter estimates and outputs all relevant R^2 measures and barchart decompositions (e.g., Figures 6-8). That is, when predictors are cluster-mean-centered, all R^2's in Table 1 and decompositions in Figure 1 are outputted. When predictors are not cluster-mean-centered, the total R^2's from Table 5, as well as barchart decompositions are outputted. Any number of level-1 and/or level-2 predictors is supported. Any of the level-1 predictors can have random slopes.

r2MLM R function Input:

data – Dataset with rows denoting observations and columns denoting variables

within_covs – List of numbers corresponding to the columns in the dataset of the level-1 predictors used in the MLM (if none used, set to NULL)

between_covs – List of numbers corresponding to the columns in the dataset of the level-2 predictors used in the MLM (if none used, set to NULL)

random_covs – List of numbers corresponding to the columns in the dataset of the level-1 predictors that have random slopes in the MLM (if no random slopes, set to NULL)

gamma_w – Vector of fixed slope estimates for all level-1 predictors, to be entered in the order of the predictors listed by *within_covs* (if none, set to NULL)

gamma_b – Vector of fixed intercept estimate (if applicable; see has_intercept below) and fixed slope estimates for all level-2 predictors, to be entered intercept first (if applicable) followed by level-2 slopes in the order listed by *between_covs* (if none, set to NULL)

Tau – random effect covariance matrix; note that the first row/column denotes the intercept variance and covariances (if intercept is fixed, set all to 0) and each subsequent row/column denotes a given random slope's variance and covariances (to be entered in the order listed by *random_covs*)

sigma2 – level-1 residual variance

has_intercept – if set to TRUE, the first element of *gamma_b* is assumed to be the fixed intercept estimate; if set to FALSE, the first element of *gamma_b* is assumed to be the first fixed level-2 predictor slope; set to TRUE by default

clustermeancentered – if set to TRUE, all level-1 predictors (indicated by the *within_covs* list) are assumed to be cluster-mean-centered and function will output all decompositions; if set to FALSE, function will output only total decompositions (see Description above); set to TRUE by default

r2MLM R function Code:

```
r2MLM <- function(data,within_covs,between_covs,random_covs,
                  gamma_w,gamma_b,Tau,sigma2,has_intercept=T,clustermeancentered=T){
  if(has_intercept==T){
  if(length(gamma_b)>1) gamma <- c(1,gamma_w,gamma_b[2:length(gamma_b)])
  if(length(gamma_b)==1) gamma <- c(1,gamma_w)
  if(is.null(within_covs)==T) gamma_w <- 0
  }
  if(has_intercept==F){
  gamma <- c(gamma_w,gamma_b)
  if(is.null(within_covs)==T) gamma_w <- 0
  if(is.null(between_covs)==T) gamma_b <- 0
  }
  if(is.null(gamma)) gamma <- 0
  ##compute phi
  phi <- var(cbind(1,data[,c(within_covs)],data[,c(between_covs)]),na.rm=T)
  if(has_intercept==F) phi <- var(cbind(data[,c(within_covs)],data[,c(between_covs)]),na.rm=T)
```

```
if(is.null(within_covs)==T & is.null(within_covs)==T & has_intercept==F) phi <- 0
phi_w <- var(data[,within_covs],na.rm=T)
if(is.null(within_covs)==T) phi_w <- 0
phi_b <- var(cbind(1,data[,between_covs]),na.rm=T)
if(is.null(between_covs)==T) phi_b <- 0
##compute psi and kappa
var_randomcovs <- var(cbind(1,data[,c(random_covs)]),na.rm=T)
if(length(Tau)>1) psi <- matrix(c(diag(Tau)),ncol=1)
if(length(Tau)==1) psi <- Tau
if(length(Tau)>1) kappa <- matrix(c(Tau[lower.tri(Tau)==TRUE]),ncol=1)
if(length(Tau)==1) kappa <- 0
v <- matrix(c(diag(var_randomcovs)),ncol=1)
r <- matrix(c(var_randomcovs[lower.tri(var_randomcovs)==TRUE]),ncol=1)
if(is.null(random_covs)==TRUE){
  v <- 0
  r <- 0
  m <- matrix(1,ncol=1)
}
if(length(random_covs)>0) m <- matrix(c(colMeans(cbind(1,data[,c(random_covs)]),na.rm=T)),ncol=1)
##total variance
totalvar_notdecomp <- t(v)%*%psi + 2*(t(r)%*%kappa) + t(gamma)%*%phi%*%gamma + t(m)%*%Tau%*%m + sigma2
totalwithinvar <- (t(gamma_w)%*%phi_w%*%gamma_w) + (t(v)%*%psi + 2*(t(r)%*%kappa)) + sigma2
totalbetweenvar <- (t(gamma_b)%*%phi_b%*%gamma_b) + Tau[1]
totalvar <- totalwithinvar + totalbetweenvar
##total decomp
decomp_fixed_notdecomp <- (t(gamma)%*%phi%*%gamma) / totalvar
decomp_fixed_within <- (t(gamma_w)%*%phi_w%*%gamma_w) / totalvar
decomp_fixed_between <- (t(gamma_b)%*%phi_b%*%gamma_b) / totalvar
decomp_fixed <- decomp_fixed_within + decomp_fixed_between
decomp_varslopes <- (t(v)%*%psi + 2*(t(r)%*%kappa)) / totalvar
decomp_varmeans <- (t(m)%*%Tau%*%m) / totalvar
decomp_sigma <- sigma2/totalvar
##within decomp
decomp_fixed_within_w <- (t(gamma_w)%*%phi_w%*%gamma_w) / totalwithinvar
decomp_varslopes_w <- (t(v)%*%psi + 2*(t(r)%*%kappa)) / totalwithinvar
decomp_sigma_w <- sigma2/totalwithinvar
##between decomp
decomp_fixed_between_b <- (t(gamma_b)%*%phi_b%*%gamma_b) / totalbetweenvar
decomp_varmeans_b <- Tau[1] / totalbetweenvar
#NEW measures
if (clustermeancentered==TRUE){
R2_f <- decomp_fixed
R2_f1 <- decomp_fixed_within
R2_f2 <- decomp_fixed_between
R2_fv <- decomp_fixed + decomp_varslopes
R2_fvm <- decomp_fixed + decomp_varslopes + decomp_varmeans
R2_v <- decomp_varslopes
R2_m <- decomp_varmeans
R2_f_w <- decomp_fixed_within_w
R2_f_b <- decomp_fixed_between_b
R2_fv_w <- decomp_fixed_within_w + decomp_varslopes_w
R2_v_w <- decomp_varslopes_w
R2_m_b <- decomp_varmeans_b
}
if (clustermeancentered==FALSE){
  R2_f <- decomp_fixed_notdecomp
  R2_fv <- decomp_fixed_notdecomp + decomp_varslopes
  R2_fvm <- decomp_fixed_notdecomp + decomp_varslopes + decomp_varmeans
  R2_v <- decomp_varslopes
  R2_m <- decomp_varmeans
}
if(clustermeancentered==TRUE){
decomp_table <- matrix(c(decomp_fixed_within,decomp_fixed_between,decomp_varslopes,decomp_varmeans,decomp_sigma,
                        decomp_fixed_within_w,"NA",decomp_varslopes_w,"NA",decomp_sigma_w,
                        "NA",decomp_fixed_between_b,"NA",decomp_varmeans_b,"NA"),ncol=3)
rownames(decomp_table) <- c("fixed, within","fixed, between","slope variation","mean variation","sigma2")
colnames(decomp_table) <- c("total","within","between")
R2_table <- matrix(c(R2_f1,R2_f2,R2_v,R2_m,R2_f,R2_fv,R2_fvm,
                  R2_f_w,"NA",R2_v_w,"NA","NA",R2_fv_w,"NA",
                  "NA",R2_f_b,"NA",R2_m_b,"NA","NA","NA")
```

```
                  ,ncol=3)
  rownames(R2_table) <- c("f1","f2","v","m","f","fv","fvm")
  colnames(R2_table) <- c("total","within","between")
  }
  ##barchart
   if(clustermeancentered==TRUE){
  contributions_stacked <- matrix(c(decomp_fixed_within,decomp_fixed_between,decomp_varslopes,decomp_varmeans,decomp_sigma,
                                  decomp_fixed_within_w,0,decomp_varslopes_w,0,decomp_sigma_w,
                                  0,decomp_fixed_between_b,0,decomp_varmeans_b,0),5,3)
  colnames(contributions_stacked) <- c("total","within","between")
  rownames(contributions_stacked) <- c("fixed slopes (within)",
                                     "fixed slopes (between)",
                                     "slope variation (within)",
                                     "intercept variation (between)",
                                     "residual (within)")
  barplot(contributions_stacked, main="Decomposition", horiz=FALSE,
          ylim=c(0,1),col=c("darkred","steelblue","darkred","midnightblue","white"),ylab="proportion of variance",
          density=c(NA,NA,30,40,NA),angle=c(0,45,0,135,0),xlim=c(0,1),width=c(.3,.3))
  legend(.30,-.1,legend=rownames(contributions_stacked),fill=c("darkred","steelblue","darkred","midnightblue","white"),
         cex=.7, pt.cex = 1,xpd=T,density=c(NA,NA,30,40,NA),angle=c(0,45,0,135,0))
 }
  if(clustermeancentered==FALSE){
    decomp_table <- matrix(c(decomp_fixed_notdecomp,decomp_varslopes,decomp_varmeans,decomp_sigma),ncol=1)
    rownames(decomp_table) <- c("fixed","slope variation","mean variation","sigma2")
    colnames(decomp_table) <- c("total")
    R2_table <- matrix(c(R2_f,R2_v,R2_m,R2_fv,R2_fvm),ncol=1)
    rownames(R2_table) <- c("f","v","m","fv","fvm")
    colnames(R2_table) <- c("total")
    ##barchar
    contributions_stacked <- matrix(c(decomp_fixed_notdecomp,decomp_varslopes,decomp_varmeans,decomp_sigma),4,1)
    colnames(contributions_stacked) <- c("total")
    rownames(contributions_stacked) <- c("fixed slopes",
                                       "slope variation",
                                       "intercept variation",
                                       "residual")
    barplot(contributions_stacked, main="Decomposition", horiz=FALSE,
            ylim=c(0,1),col=c("darkblue","darkblue","darkblue","white"),ylab="proportion of variance",
            density=c(NA,30,40,NA),angle=c(0,0,135,0),xlim=c(0,1),width=c(.6))
    legend(.30,-.1,legend=rownames(contributions_stacked),fill=c("darkblue","darkblue","darkblue","white"),
           cex=.7, pt.cex = 1,xpd=TRUE,density=c(NA,30,40,NA),angle=c(0,0,135,0))
 }
  Output <- list(noquote(decomp_table),noquote(R2_table))
  names(Output) <- c("Decompositions","R2s")
  return(Output)
}
```

r2MLM R function Example Input:

```
#NOTE: estimates in the input represent hypothetical results for a random slope model with two level-1 predictors and two level-2 predictors
#in practice a user would have previously obtained these input estimates by fitting their model in MLM software
#additionally, the input consists of hypothetical predictor data, whereas in practice a user would read-in their actual data

data <- matrix(NA,100,4)
xs <- mvrnorm(n=100,mu=c(0,0),Sigma=matrix(c(2,.75,.75,1.5),2,2))
ws <- mvrnorm(n=10,mu=c(0,2),Sigma=matrix(c(1,.5,.5,2),2,2))
data[,1:2] <- xs
for (i in seq(10)){
  data[(10*(i-1)+1):(i*10),3] <- ws[i,1]
  data[(10*(i-1)+1):(i*10),4] <- ws[i,2]
  data[(10*(i-1)+1):(i*10),1] <- data[(10*(i-1)+1):(i*10),1] - mean(data[(10*(i-1)+1):(i*10),1])
  data[(10*(i-1)+1):(i*10),2] <- data[(10*(i-1)+1):(i*10),2] - mean(data[(10*(i-1)+1):(i*10),2])
}
r2MLM(data,within_covs=c(1,2),between_covs=c(3,4),random_covs=c(1,2),
      gamma_w=c(2.5,-1),gamma_b=c(1,.25,1.5),Tau=matrix(c(4,1,.75,1,1,.25,.75,.25,.5),3,3),sigma2=10)
```

Online Appendix B: Empirical example results

Online Appendix B Table 1. Parameter estimates and standards errors from Empirical Example 1

Fixed effects			
	Est	**SE**	***t***
intercept	51.19	0.86	59.71*
school-mean-centered homework	1.88	0.84	2.22*
school-mean-centered parental education	1.81	0.39	4.62**
school-mean homework	0.70	1.53	0.45
school-mean parental education	3.18	1.01	3.15**
Variance components			
	Est	**SE**	***z* †**
variance of intercept	13.61	5.27	2.58**
variance of school-mean-centered homework	14.36	5.25	2.74**
variance of school-mean-centered parental education	1.17	1.43	1.43
covariance of intercept with school-mean-centered homework	-0.14	3.75	-0.04
covariance of intercept with school-mean-centered parental education	-2.88	1.92	-1.50
covariance of school-mean-centered homework with school-mean-centered parental education	-0.16	1.72	-0.09
variance of level-1 residual	49.32	3.26	15.13**

Notes: Results obtained from SAS Proc Mixed. *significant, $p<.05$; **significant, $p<.01$
† *z*-tests of variance components are conservative; one way to address this is to employ the alpha-correction approach of Fitzmaurice et al. (2011, p. 209) which uses α=.10 instead of α=.05. For a discussion of other alternatives, see Rights and Sterba (2016).

Online Appendix B Table 2. Parameter estimates and standards errors from Empirical Example 2

Fixed effects			
	Est	**SE**	***t***
Intercept	41.04	0.31	134.29**
school-mean-centered verbal IQ	2.24	0.09	26.14**
school-mean-centered SES	0.17	0.02	10.77**
school-mean-centered verbal IQ * SES	-0.01	0.01	-1.63
school-mean verbal IQ	3.36	0.37	9.13**
school-mean SES	0.07	0.05	1.34
school-mean verbal IQ * SES	-0.09	0.04	-2.37*
Variance components			
	Est	**SE**	***z* †**
variance of intercept	8.03	1.35	5.95**
variance of school-mean-centered verbal IQ	0.23	0.11	2.09**
covariance of intercept with school-mean-centered verbal IQ	-0.92	0.30	-3.06*
variance of level-1 residual	39.18	1.22	32.04**

Notes: Results obtained from SAS Proc Mixed. *significant, p<.05; **significant, p<.01
† z-tests of variance components are conservative; one way to address this is to employ the alpha-correction approach of Fitzmaurice et al. (2011, p. 209) which uses α=.10 instead of α=.05. For a discussion of other alternatives, see Rights and Sterba (2016).

Online Appendix B Table 3. Parameter estimates and standards errors from Empirical Example 3

Fixed effects			
	Est	**SE**	***t***
intercept	5.08	0.08	65.29**
class-mean-centered extraversion	0.45	0.02	25.67**
class-mean-centered sex	1.23	0.04	33.68**
teacher experience	0.06	0.01	5.22**
class-mean-centered extraversion * teacher experience	-0.03	0.01	-9.50**
Variance components			
	Est	**SE**	***z*** †
variance of intercept	0.58	0.09	6.68**
variance of class-mean-centered extraversion	0.01	0.01	1.26
covariance of intercept with class-mean-centered extraversion	-0.01	0.01	-0.91
variance of level-1 residual	39.18	0.02	29.97**

Notes: Results obtained from SAS Proc Mixed. *significant, $p<.05$; **significant, $p<.01$
† z-tests of variance components are conservative; one way to address this is to employ the alpha-correction approach of Fitzmaurice et al. (2011, p. 209) which uses $\alpha=.10$ instead of $\alpha=.05$. For a discussion of other alternatives, see Rights and Sterba (2016).