SMiP 2024 MLM Example1

Lesa Hoffman

May 2024

Table of Contents


```
knitr::opts_chunk$set(echo=TRUE)

# Working directory for all chunks below
  knitr::opts_knit$set(root.dir="C:/Dropbox/Papers and Data/SMiP/SMiP_2024_MLM_Part1")
```

Multilevel Models for Subjects Crossed with Items Predicting Response Time (RT)

This example will illustrate the estimation and interpretation of multilevel models with crossed random effects for an observed continuous outcome. The data are from the supplemental materials of Locker, Hoffman, and Bovaird (2007), as included in the Example 1 .zip folder. Response time (RT) outcomes for a lexical decision task (in which subjects decide as quickly as they can whether each item is a word or a non-word) were collected for 39 items from 38 subjects (total possible observations = 1482; total actual observations = 1392 after removing RTs for inaccurate responses). Items are words that varied systematically in two features: semantic frequency (freq01: 0=low, 1=high) and neighborhood size (size01: 0=small, 1=large). Given the small samples of subjects and items, we will use REML estimation and Satterthwaite denominator degrees of freedom.

Loading Packages

First (below), we set global options to my preferred versions, and then we install and load the R packages to be used.

```
# Set width of output and number of significant digits printed,
# number of digits before using scientific notation, shut off significance stars
options(width=120, digits=8, scipen=9, show.signif.stars=FALSE)

#####  Check to see if packages are downloaded, install if not, then load  #####

# To import xls or xlsx data as table
if (!require("readxl")) install.packages("readxl"); library(readxl)
## Loading required package: readxl
## Warning: package 'readxl' was built under R version 4.3.1

# To get compact data description
if (!require("psych")) install.packages("psych"); library(psych)
## Loading required package: psych

# To estimate MLMs using gls or lme
if (!require("nlme")) install.packages("nlme"); library(nlme)
## Loading required package: nlme

# To estimate MLMs using lmer
# Re-install to fix problem of matrix incompatibility
#oo <- options(repos = "https://cran.r-project.org/")
#utils::install.packages("Matrix")
#utils::install.packages("lme4")
#options(oo)
library(Matrix); library(lme4)
## Warning: package 'Matrix' was built under R version 4.3.3
## Warning: package 'lme4' was built under R version 4.3.3
##
## Attaching package: 'lme4'
## The following object is masked from 'package:nlme':
##
##     lmList

# To get Satterthwaite DDF in lmer
if (!require("lmerTest")) install.packages("lmerTest"); library(lmerTest)
## Loading required package: lmerTest
##
## Attaching package: 'lmerTest'
## The following object is masked from 'package:lme4':
##
##     lmer
## The following object is masked from 'package:stats':
##
##     step

# To get ICC in lmer
if (!require("performance")) install.packages("performance"); library(performance)
## Loading required package: performance
## Warning: package 'performance' was built under R version 4.3.3
```

```
# Clear environment (re-run as needed for troubleshooting purposes)
#rm(list=ls())
```

Data Import, Manipulation, and Description

Next, we import the excel data file for this example and examine descriptive statistics. The data is in "long" (or "stacked") format in which each row contains one trial (per subject per item).

```
# Define variables for working directory and data name -- CHANGE THESE
filesave = "C:\\Dropbox/Papers and Data/SMiP/SMiP_2024_MLM_Part1/"
filename = "Example1.xlsx"
setwd(dir=filesave)

# Import Example 1 excel trial-level dataset
Example1 = read_excel(paste0(filesave,filename))
# Convert to data frame to use in analysis
Example1 = as.data.frame(Example1)

# Filter to only cases complete on all variables to be used below
Example1 = Example1[complete.cases(Example1[ , c("RT","freq01","size01")]),]

print("Descriptives for RT across all trials")

## [1] "Descriptives for RT across all trials"

describe(x=Example1[,c("RT")], fast=TRUE)

##    vars    n   mean     sd min  max range   se
## X1    1 1392 632.38 146.08 352 1806  1454 3.92

print("Variance of RT across all trials")

## [1] "Variance of RT across all trials"

var(x=Example1[,c("RT")])

## [1] 21340.288

# Add mean subject RT and number of responses per subject to dataset
Example1$SubjectMeanRT = ave(x=Example1$RT, Example1$SubjectID, FUN=mean)
Example1$SubjectN = ave(x=Example1$RT, Example1$SubjectID, FUN=length)
# Create subject-level dataset
Example1Subjects = unique(Example1[,c("SubjectID","SubjectMeanRT","SubjectN")])
print("Descriptives for SubjectMeanRT and SubjectN across subjects")

## [1] "Descriptives for SubjectMeanRT and SubjectN across subjects"

describe(x=Example1Subjects[,c("SubjectMeanRT","SubjectN")], fast=TRUE)

##               vars  n   mean    sd    min    max  range    se
## SubjectMeanRT    1 38 631.33 74.92 501.69 783.83 282.13 12.15
## SubjectN         2 38  36.63  1.84  30.00  39.00   9.00  0.30

# Add mean item RT and number of responses per item to dataset
Example1$ItemMeanRT = ave(x=Example1$RT, Example1$ItemID, FUN=mean)
Example1$ItemN = ave(x=Example1$RT, Example1$ItemID, FUN=length)
```

```
# Create item-level dataset to see how many items there are of each kind
Example1Items = unique(Example1[,c("ItemID","freq01","size01","ItemMeanRT","ItemN")])
print("Cross-tabulation of item predictors")
```

```
## [1] "Cross-tabulation of item predictors"
```

```
table(Example1Items$freq01,Example1Items$size01)
```

```
##
##      0  1
##   0 10 10
##   1 10  9
```

```
print("Descriptives for ItemMeanRT and ItemN across items")
```

```
## [1] "Descriptives for ItemMeanRT and ItemN across items"
```

```
describe(x=Example1Items[,c("ItemMeanRT","ItemN")], fast=TRUE)
```

```
##            vars  n   mean    sd    min max  range   se
## ItemMeanRT    1 39 637.18 56.16 566.97 833 266.03 8.99
## ItemN         2 39  35.69  3.78  18.00  38  20.00 0.61
```

As shown in the descriptive tables above, there is a considerable range in mean RT across subjects as well as across items. Thus, our models will likely need to represent RT variability across both crossed level-2 dimensions (an empirical question to be answered below).

Empty Models for Partitioning RT Variance

Next, we will estimate and compare three "empty means" (i.e., no-predictor) models for RT, in which the fixed effects in each model include only an intercept, denoted in each as γ_{000}. In the notation below, the subscripts are t = level-1 trial, i = level-2 item, and s = level-2 subject.

Empty Means Model 1: Single-level

The first empty means (single-level) model contains only a level-1 e_{tis} trial-specific residual, whose estimated variance across all trials is denoted as the level-1 residual variance σ_e^2:

$$\text{Empty Model 1: } RT_{tis} = \gamma_{000} + e_{tis}$$
$$e_{tis} \sim N(0, \sigma_e^2)$$

```
print("Empty Model 1: Single-level ignoring dependency of subjects and items")
```

```
## [1] "Empty Model 1: Single-level ignoring dependency of subjects and items"
```

```
Empty1 = gls(data=Example1, method="REML", model=RT~1)
print("Show default results"); summary(Empty1)
```

```
## [1] "Show default results"
```

```
## Generalized least squares fit by REML
##   Model: RT ~ 1
##   Data: Example1
##         AIC       BIC     logLik
##   17824.703 17835.179 -8910.3516
##
```

```
## Coefficients:
##                Value Std.Error   t-value p-value
## (Intercept) 632.3829 3.9154395 161.51007       0
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -1.91933767 -0.67347192 -0.19429278  0.50393967  8.03389752
##
## Residual standard error: 146.08316
## Degrees of freedom: 1392 total; 1391 residual
```

```
print("Model -2LL = "); -2*logLik(Empty1)
```

```
## [1] "Model -2LL = "
```

```
## 'log Lik.' 17820.703 (df=2)
```

```
print("Level-1 residual variance = "); summary(Empty1)$sigma^2
```

```
## [1] "Level-1 residual variance = "
```

```
## [1] 21340.288
```

As shown above, the single-level empty model perfectly reproduces the original RT mean = 632.4 as the fixed intercept γ_{000}. The REML estimate of RT SD = 146.1 (with total variance $\sigma_e^2 = 21340.3$) also matches that of the original RT outcome as expected.

Btw, we are using the function `gls` from the package `nlme` (instead of the function `lm` from base R that would be equivalent in this case) in order to get a $-2LL$ value by which to demonstrate a model comparison using a likelihood ratio test (LRT) below.

Empty Model 2: Two-level for trials nested within subjects

From here onwards, we are switching to the function `lmer` from the package `lme4` (as called by the `lmerTest` package, more specifically) to estimate the two-level model. In each model below, the `REML=TRUE` argument specifies that we want to use restricted maximum likelihood estimation. The `ddf="Satterthwaite"` argument specifies that we want to use Satterthwaite denominator degrees of freedom for the t-tests of fixed effects. The function `lme` from the package `nlme` could also have been used to estimate the two-level model, but it does not provide Satterthwaite denominator degrees of freedom.

The second empty means (two-level nested) model below adds a random intercept for each subject, U_{00s}, whose estimated variance across subjects is denoted as the level-2 subject random intercept variance $\tau^2_{U_{00s}}$:

$$\text{Empty Model 2: } RT_{tis} = \gamma_{000} + U_{00s} + e_{tis}$$
$$e_{tis} \sim N(0, \sigma_e^2)$$
$$U_{00s} \sim N\left(0, \tau^2_{U_{00s}}\right)$$

```
print("Empty Model 2: Two-level nested for trials nested in subjects only")
```

```
## [1] "Empty Model 2: Two-level nested for trials nested in subjects only"
```

```
Empty2 = lmer(data=Example1, REML=TRUE, formula=RT~1+(1|SubjectID))
print("Show results using Sattherwaite DDF")
```

```
## [1] "Show results using Sattherwaite DDF"
```

```
summary(Empty2, ddf="Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: RT ~ 1 + (1 | SubjectID)
##    Data: Example1
##
## REML criterion at convergence: 17540.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.33983 -0.62036 -0.18420  0.36356  9.38355
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  SubjectID (Intercept)  5167.1   71.883
##  Residual              16307.2  127.700
## Number of obs: 1392, groups:  SubjectID, 38
##
## Fixed effects:
##             Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)  631.417     12.154  37.004  51.951 < 2.2e-16
```

As shown above, the two-level nested model returns a fixed intercept $\gamma_{000} = 631.4$ that is nearly identical to the original RT mean = 632.4, but it now represents the sample mean of the subject means (and is thus a weighted mean). The sum of the two estimated variances = 21474.3 is not the same as the model-estimated variance from the single-level model = 21340.3 (but it will be closer in completely balanced data, in which we have the same number of responses for each subject).

```
print("Show intraclass correlation"); icc(Empty2)
```

```
## [1] "Show intraclass correlation"
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.241
##   Unadjusted ICC: 0.241
```

The two-level nested model partitions the observed variance in RT into between-subject mean differences (24.1% as given by the intraclass correlation, $ICC_S = 0.241$) and within-subject deviations from their subject means (the remaining 75.9%). The ICC was computed using `icc` from the `performance` package as follows:

$$ICC_S = \frac{\tau^2_{U_{00S}}}{\tau^2_{U_{00S}} + \sigma^2_e} = \frac{5167.1}{5167.1 + 16307.2} = .241$$

```
print("Print stored variance components table for reference")
```

```
## [1] "Print stored variance components table for reference"
```

```
as.data.frame(VarCorr(Empty2))
```

```
##         grp        var1 var2       vcov      sdcor
## 1 SubjectID (Intercept) <NA>  5167.0959  71.882515
## 2  Residual        <NA> <NA> 16307.2062 127.699672

# Compute model-implied total variance
Empty2TotVar = as.data.frame(VarCorr(Empty2))[1,4] +
               as.data.frame(VarCorr(Empty2))[2,4]
print("Total model-implied variance = "); Empty2TotVar

## [1] "Total model-implied variance = "

## [1] 21474.302

# Save subject random intercept variance as object to show computation of ICC
Empty2SubVar = as.data.frame(VarCorr(Empty2))[1,4]
# Manual computation of subject ICC
print("Subject ICC = "); Empty2ICC_S = Empty2SubVar/Empty2TotVar; Empty2ICC_S

## [1] "Subject ICC = "

## [1] 0.24061764
```

The ranova command then conducts a likelihood ratio test comparing the log-likelihood (LL) from the empty means models with vs without the level-2 subject random intercept variance $\tau^2_{U_{0oS}}$:

```
print("Show intraclass correlation LRT"); ranova(Empty2)

## [1] "Show intraclass correlation LRT"

## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## RT ~ (1 | SubjectID)
##                  npar   logLik     AIC     LRT Df Pr(>Chisq)
## <none>              3 -8770.13 17546.3
## (1 | SubjectID)     2 -8910.35 17824.7 280.439  1 < 2.22e-16
```

Here's how to compute an LRT using either LL or $-2LL$, in which the simpler model goes first either way:

$$LL \text{ version: } \left(LL_{\text{Empty1}}\right) - \left(LL_{\text{Empty2}}\right) = (-8910.4) - (-8770.1) = 280.4$$

$$-2LL \text{ version: } \left(-2LL_{\text{Empty1}}\right) - \left(-2LL_{\text{Empty2}}\right) = (17820.7) - (17540.3) = 280.4$$

The $-2LL$ difference, denoted as $-2\Delta LL$, is then treated as a χ^2 statistic with 1 degree of freedom (df) to test whether the level-2 subject random intercept variance is significantly > 0. As expected, the $ICC = 0.241$ is significantly > 0, $-2\Delta LL(1) = 280.4$, $p < .001$.

However, because variances are bounded at 0 (i.e., the null hypothesis of 0 is on its boundary), the more correct way to conduct this LRT is to compare the $-2\Delta LL$ to a mixture of χ^2 distributions: with $df = 0$ for when the variance would have been negative in a two-sided symmetric sampling distribution around 0 (which is not allowed), and $df = 1$ for when the variance would have been positive. For this simple case, this amounts to cutting the p-value in half, which is still $< .001$. Btw, the critical value for a mixture of $df = 0{,}1$ is 2.71 at $p < .05$ (instead of 3.84 for $df = 1$).

Empty Model 3: Two-level for trials nested in subjects crossed with items

The third empty means (two-level crossed) model below adds a random intercept for each item, U_{0i0}, whose estimated variance across items is then denoted as the level-2 item random intercept variance $\tau^2_{U_{0I0}}$:

$$\text{Empty Model 3: } RT_{tis} = \gamma_{000} + U_{00s} + U_{0i0} + e_{tis}$$
$$e_{tis} \sim N(0, \sigma^2_e)$$
$$U_{00s} \sim N\left(0, \tau^2_{U_{00S}}\right)$$
$$U_{0i0} \sim N\left(0, \tau^2_{U_{0I0}}\right)$$

```
print("Empty Model 3: Two-level crossed for trials nested in subjects and in items")
```

```
## [1] "Empty Model 3: Two-level crossed for trials nested in subjects and in items"
```

```
Empty3 = lmer(data=Example1, REML=TRUE, formula=RT~1+(1|SubjectID)+(1|ItemID))
print("Show results using Sattherwaite DDF")
```

```
## [1] "Show results using Sattherwaite DDF"
```

```
summary(Empty3, ddf="Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: RT ~ 1 + (1 | SubjectID) + (1 | ItemID)
##    Data: Example1
##
## REML criterion at convergence: 17439.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.27264 -0.60106 -0.14868  0.37684  9.74505
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  ItemID    (Intercept)  2414.1   49.134
##  SubjectID (Intercept)  5166.7   71.880
##  Residual              14343.0  119.762
## Number of obs: 1392, groups:  ItemID, 39; SubjectID, 38
##
## Fixed effects:
##             Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)  635.328     14.434  59.435  44.016 < 2.2e-16
```

As shown above, the two-level crossed model returns a fixed intercept $\gamma_{000} = 635.3$ that is very close to the original RT mean = 632.4, but it now represents the sample mean of the subject means AND the item means (and is thus a weighted mean). We now have three variance components (or "piles" of variance, as I like to call them).

```
print("Show intraclass correlations for proportion of variance due to each sampling
dimension")
```

```
## [1] "Show intraclass correlations for proportion of variance due to each sampling
dimension"
```

```
icc(Empty3, by_group=TRUE)
```

```
## # ICC by Group
##
## Group     |   ICC
## ----------------
## ItemID    | 0.110
## SubjectID | 0.236
```

Here is how the ICCs above were computed:

$$ICC_I = \frac{\tau^2_{U_{0I0}}}{\tau^2_{U_{00S}} + \tau^2_{U_{0I0}} + \sigma^2_e} = \frac{2414.1}{5166.7 + 2414.1 + 14343.0} = .110$$

$$ICC_S = \frac{\tau^2_{U_{00S}}}{\tau^2_{U_{00S}} + \tau^2_{U_{0I0}} + \sigma^2_e} = \frac{5166.7}{5166.7 + 2414.1 + 14343.0} = .236$$

As shown above, the two-level crossed model partitions the observed variance in RT into between-subject mean differences (23.6% as given by $ICC_S = 0.236$, between-item mean differences (11% as given by $ICC_I = 0.11$, and trial-specific deviations from their subject and item means (the remaining 65.4%). Note that the previous two-level nested model $ICC_S = 0.241$ is very similar to that obtained from the crossed model, as it was the level-1 residual variance that was partitioned into a new level-2 item random intercept variance.

Here is how these ICCs can be computed manually using saved output:

```
print("Print stored variance components table for reference")
```

```
## [1] "Print stored variance components table for reference"
```

```
as.data.frame(VarCorr(Empty3))
```

```
##         grp        var1 var2       vcov      sdcor
## 1    ItemID (Intercept) <NA>  2414.1132  49.133626
## 2 SubjectID (Intercept) <NA>  5166.7217  71.879912
## 3  Residual        <NA> <NA> 14343.0430 119.762444
```

```
# Compute model-implied total variance
Empty3TotVar = as.data.frame(VarCorr(Empty3))[1,4] +
               as.data.frame(VarCorr(Empty3))[2,4] +
               as.data.frame(VarCorr(Empty3))[3,4]
print("Total model-implied variance = "); Empty3TotVar
```

```
## [1] "Total model-implied variance = "
```

```
## [1] 21923.878
```

```
# Save each variance as an object
Empty3ItemIntVar = as.data.frame(VarCorr(Empty3))[1,4]
Empty3SubIntVar  = as.data.frame(VarCorr(Empty3))[2,4]
Empty3ResVar     = as.data.frame(VarCorr(Empty3))[3,4]
# Manual computation of subject and item ICCs
print("Subject ICC = "); Empty3ICC_S = Empty3SubIntVar /Empty3TotVar; Empty3ICC_S
```

```
## [1] "Subject ICC = "
```

```
## [1] 0.23566641

print("Item ICC = ");    Empty3ICC_I = Empty3ItemIntVar/Empty3TotVar; Empty3ICC_I

## [1] "Item ICC = "

## [1] 0.11011342
```

The ranova command then conducts a likelihood ratio test comparing the log-likelihood from the empty models with vs without each level-2 random intercept variance:

```
print("Show intraclass correlation LRTs"); ranova(Empty3)

## [1] "Show intraclass correlation LRTs"

## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## RT ~ (1 | SubjectID) + (1 | ItemID)
##                  npar   logLik     AIC     LRT Df Pr(>Chisq)
## <none>              4 -8719.93 17447.9
## (1 | SubjectID)     3 -8878.90 17763.8 317.925  1 < 2.22e-16
## (1 | ItemID)        3 -8770.13 17546.3 100.399  1 < 2.22e-16
```

The LRT from ranova indicates that both random intercept variances are significantly > 0, items: $-2\Delta LL(1) = 100.4, p < .001$, and subjects: $-2\Delta LL(1) = 317.9, p < .001$. The conclusion would be the same using a mixture of χ^2 distributions with $df = 0{,}1$ as was found in the previous model.

To help convey effect size of these piles of variance in a more meaningful metric, we can use the results of our model to compute 95% random effects confidence intervals for the subject and item random intercepts. These indicate the expected range of the subject intercepts and item intercepts for 95% of our sample as follows:

$$\text{Subject Random Intercept 95\% CI} = \gamma_{000} \pm 1.96 \times \sqrt{\tau^2_{U_{00S}}}$$

$$\text{Subject Random Intercept 95\% CI} = 635.3 \pm 1.96 \times \sqrt{5166.7} = 494.4 \text{ to } 776.2$$

$$\text{Item Random Intercept 95\% CI} = \gamma_{000} \pm 1.96 \times \sqrt{\tau^2_{U_{0I0}}}$$

$$\text{Item Random Intercept 95\% CI} = 635.3 \pm 1.96 \times \sqrt{2414.1} = 539.0 \text{ to } 731.6$$

Here is how the random effect confidence intervals can be computed using saved output:

```
# Save fixed intercept for use below
Empty3FixInt = fixef(Empty3)
print("95% random intercept confidence interval for subjects")

## [1] "95% random intercept confidence interval for subjects"

SubInt_LCI = Empty3FixInt - 1.96*sqrt(Empty3SubIntVar)
SubInt_UCI = Empty3FixInt + 1.96*sqrt(Empty3SubIntVar)
print("Subject intercept lower CI = "); SubInt_LCI

## [1] "Subject intercept lower CI = "
```

```
## (Intercept)
##    494.4431

print("Subject intercept upper CI = "); SubInt_UCI

## [1] "Subject intercept upper CI = "

## (Intercept)
##   776.21235

print("95% random intercept confidence interval for items")

## [1] "95% random intercept confidence interval for items"

ItemInt_LCI = Empty3FixInt - 1.96*sqrt(Empty3ItemIntVar)
ItemInt_UCI = Empty3FixInt + 1.96*sqrt(Empty3ItemIntVar)
print("Item intercept lower CI = "); ItemInt_LCI

## [1] "Item intercept lower CI = "

## (Intercept)
##   539.02582

print("Item intercept upper CI = "); ItemInt_UCI

## [1] "Item intercept upper CI = "

## (Intercept)
##   731.62963
```

Conditional Models Including Item Predictors of RT

Conditional Model 1: Add Fixed Slopes of Item Predictors

Next, we add fixed slopes for the item predictors (freq01 and size01) to the two-level crossed model to predict RT (in which the index in the second subject keeps track of which item predictor each is). The fixed effects in each model include the intercept γ_{000}, the fixed slopes γ_{010} and γ_{020} for the main effects of freq01 and size01, respectively, and a fixed slope for their interaction term, γ_{030}:

Conditional Model 1: $RT_{tis} = \gamma_{000} + \gamma_{010}(\text{freq01}_i) + \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i) + U_{00s} + U_{0i0} + e_{tis}$

$e_{tis} \sim N(0, \sigma_e^2)$

$U_{00s} \sim N(0, \tau^2_{U_{00S}})$

$U_{0i0} \sim N(0, \tau^2_{U_{0I0}})$

```
print("Conditional Model 1: Add fixed slopes for item predictors")

## [1] "Conditional Model 1: Add fixed slopes for item predictors"

Cond1 = lmer(data=Example1, REML=TRUE, formula=
             RT~1+freq01+size01+freq01:size01+(1|SubjectID)+(1|ItemID))
print("Show results using Sattherwaite DDF")

## [1] "Show results using Sattherwaite DDF"

summary(Cond1, ddf="Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: RT ~ 1 + freq01 + size01 + freq01:size01 + (1 | SubjectID) +      (1 |
ItemID)
##    Data: Example1
##
## REML criterion at convergence: 17402.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.29868 -0.59565 -0.15486  0.37923  9.68272
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  ItemID    (Intercept)  1692.4   41.139
##  SubjectID (Intercept)  5168.5   71.892
##  Residual              14341.0  119.754
## Number of obs: 1392, groups:  ItemID, 39; SubjectID, 38
##
## Fixed effects:
##               Estimate Std. Error       df t value  Pr(>|t|)
## (Intercept)   615.7825    18.5748  60.6612 33.1514 < 2.2e-16
## freq01         70.0208    20.5969  32.4008  3.3996  0.001807
## size01          4.4349    20.4219  31.3924  0.2172  0.829484
## freq01:size01 -72.0306    29.3780  31.7807 -2.4519  0.019894
##
## Correlation of Fixed Effects:
##             (Intr) freq01 size01
## freq01      -0.546
## size01      -0.551  0.497
## freq01:sz01  0.383 -0.701 -0.695
```

```
# Btw, here is shorter code to include interactions with all lower-order main effects:
#Cond1 = lmer(data=Example1, REML=TRUE,
#             RT~1+freq01*size01+(1|SubjectID)+(1|ItemID))
```

Let's interpret the results for the fixed effects in the model above:

fixed intercept γ_{000} =

fixed slope for freq01 γ_{010} =

fixed slope for size01 γ_{020} =

fixed slope for interaction γ_{030} =

As we can see, only some of the possible predicted RT cell means and simple slopes are given directly as model parameters (and thus are shown in the default output). To get the others, we can use the`contest1D` function from the `lmerTest` package to generate linear combinations of our fixed effects (and their corresponding standard errors and hypothesis tests using Satterthwaite denominator degrees of freedom here). In the `c=(_,_,_,_)` calls below, the blanks provide multipliers for each fixed effect in the order they appear in the output. Thus, the first number is for the intercept, the second is for the freq01 slope, the third is for the size01 slope, and the fourth is for the interaction slope.

First, to get a predicted mean for any item, we can use the model fixed effects:

$$\text{Predicted RT } = \gamma_{000} + \gamma_{010}(\text{freq01}_i) + \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i)$$

```
# Cell means (the manual way)
print("RT for low  freq, small size"); contest1D(Cond1, L=c(1,0,0,0))

## [1] "RT for low  freq, small size"

##    Estimate Std. Error        df   t value     Pr(>|t|)
## 1 615.78251   18.57485 60.661209 33.151412 1.424135e-40

print("RT for low  freq, large size"); contest1D(Cond1, L=c(1,0,1,0))

## [1] "RT for low  freq, large size"

##    Estimate Std. Error        df  t value      Pr(>|t|)
## 1 620.21746  18.549125 60.333446 33.43648 1.2145796e-40

print("RT for high freq, small size"); contest1D(Cond1, L=c(1,1,0,0))

## [1] "RT for high freq, small size"

##    Estimate Std. Error        df   t value      Pr(>|t|)
## 1 685.80335  18.742481 62.663015 36.590852 5.3255942e-44

print("RT for high freq, large size"); contest1D(Cond1, L=c(1,1,1,1))

## [1] "RT for high freq, large size"

##    Estimate Std. Error        df   t value      Pr(>|t|)
## 1 618.20773  19.151359 58.825634 32.280097 4.0663877e-39
```

Second, to get predicted slopes for any item, we use only the fixed effects that involve the predictor the slope is for, then factor that predictor out of the equation, as follows:

$$\text{Predicted RT } = \gamma_{000} + \gamma_{010}(\text{freq01}_i) + \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i)$$

$$\text{Predicted freq01 slope } = \gamma_{010}(\text{freq01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i)$$
$$\text{Predicted freq01 slope } = \gamma_{010} + \gamma_{030}(\text{size01}_i)$$

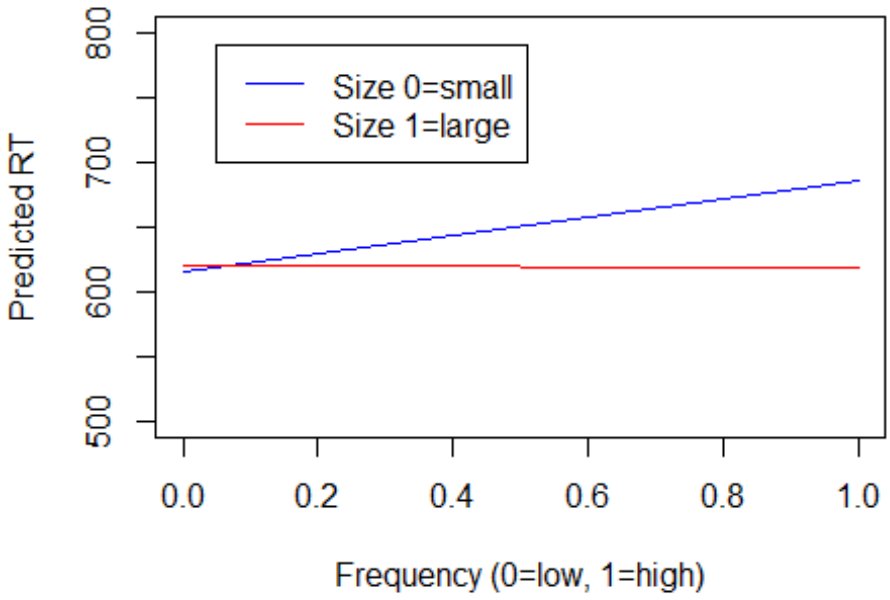
$$\text{Predicted size01 slope } = \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size}_i)$$
$$\text{Predicted size01 slope } = \gamma_{020} + \gamma_{030}(\text{freq01}_i)$$

```
# Simple effects of freq per size
print("Freq slope for small size"); contest1D(Cond1, L=c(0,1,0,0))

## [1] "Freq slope for small size"

##    Estimate Std. Error        df   t value     Pr(>|t|)
## 1 70.020836  20.596884 32.400846 3.3995839 0.0018065107

print("Freq slope for large size"); contest1D(Cond1, L=c(0,1,0,1))

## [1] "Freq slope for large size"
```

```
##     Estimate Std. Error        df      t value   Pr(>|t|)
## 1 -2.0097334  20.947767 31.194007 -0.095940223 0.92418176

# Simple effects of size per freq
print("Size slope for low freq");  contest1D(Cond1, L=c(0,0,1,0))

## [1] "Size slope for low freq"

##   Estimate Std. Error        df   t value   Pr(>|t|)
## 1 4.434948   20.42187 31.392391 0.2171666 0.82948379

print("Size slope for high freq"); contest1D(Cond1, L=c(0,0,1,1))

## [1] "Size slope for high freq"

##     Estimate Std. Error        df    t value     Pr(>|t|)
## 1 -67.595621  21.119301 32.151039 -3.2006561 0.0030813267

# Interaction (repeated for convenience, two possible interpretations)
print("Size slope diff for low vs high freq");    contest1D(Cond1, L=c(0,0,0,1))

## [1] "Size slope diff for low vs high freq"

##     Estimate Std. Error        df    t value    Pr(>|t|)
## 1 -72.030569  29.378023 31.780715 -2.4518521 0.019893842

print("Freq slope diff for small vs large size"); contest1D(Cond1, L=c(0,0,0,1))

## [1] "Freq slope diff for small vs large size"

##     Estimate Std. Error        df    t value    Pr(>|t|)
## 1 -72.030569  29.378023 31.780715 -2.4518521 0.019893842
```

We can visualize the interaction by creating “fake items” to plot predicted outcomes from the fixed effects:

```
# Make a plot of predicted outcomes
# Create columns of values to be changed across fake items (FI)
FakeFreq = c(0,1,0,1)
FakeSize = c(0,0,1,1)
# Create dataset using just-created columns and constants for other model variables
FakeItems = data.frame(SubjectID=-99, ItemID=-99, freq01=FakeFreq, size01=FakeSize)

# Merge predicted values from main-effects-only model into FI data
FakeItems = data.frame(FakeItems, yhat=predict(object=Cond1, newdata=FakeItems,
re.form=NA))

# Make plot
plot(y=FakeItems$yhat, x=FakeItems$freq01, type="n", ylim=c(500,800), xlim=c(0,1),
     xlab="Frequency (0=low, 1=high)", ylab="Predicted RT")
lines(x=FakeItems$freq01[1:2], y=FakeItems$yhat[1:2], type="l", col="blue1")
lines(x=FakeItems$freq01[3:4], y=FakeItems$yhat[3:4], type="l", col="red1")
legend(x=.05, y=790, legend=c("Size 0=small","Size 1=large"),
       col=c("blue1","red1"), lty=1) #lty=linetype
```



Third, to get an effect size for the contribution of our item predictors, we can compute a pseudo-R^2 value for the proportion reduction in item random intercept variance relative to the empty means model:

$$\text{Pseudo-R}_I^2 = \frac{\text{Empty3 Variance}_I - \text{Cond1 Variance}_I}{\text{Empty3 Variance}_I} = \frac{2414.1 - 1692.4}{2414.1} = .299$$

Here is how to compute pseudo-R^2 values using saved output:

```
print("Print stored variance components table for reference")

## [1] "Print stored variance components table for reference"

as.data.frame(VarCorr(Cond1))

##         grp        var1 var2       vcov      sdcor
## 1    ItemID (Intercept) <NA>  1692.4114  41.138928
## 2 SubjectID (Intercept) <NA>  5168.4803  71.892143
## 3  Residual        <NA> <NA> 14341.0390 119.754077

# Save each variance as an object
Cond1ItemIntVar = as.data.frame(VarCorr(Cond1))[1,4]
Cond1SubIntVar  = as.data.frame(VarCorr(Cond1))[2,4]
Cond1ResVar     = as.data.frame(VarCorr(Cond1))[3,4]
# Compute pseudo-R2 for each variance
Cond1ItemIntR2 = (Empty3ItemIntVar-Cond1ItemIntVar)/Empty3ItemIntVar; Cond1ItemIntR2

## [1] 0.2989511

Cond1SubIntR2  = (Empty3SubIntVar- Cond1SubIntVar) /Empty3SubIntVar;  Cond1SubIntR2

## [1] -0.00034036623
```

```
Cond1ResR2     = (Empty3ResVar-    Cond1ResVar)    /Empty3ResVar;     Cond1ResR2
```

```
## [1] 0.00013971802
```

We can get a significance test for that item random intercept pseudo-R^2 value using the function `contestMD` from the `lmerTest` package, which allows us to obtain a joint hypothesis test for multiple slopes at once:

```
print("Omnibus F-test for model prediction")
```

```
## [1] "Omnibus F-test for model prediction"
```

```
contestMD(Cond1, ddf="Satterthwaite", L=rbind(c(0,1,0,0),c(0,0,1,0),c(0,0,0,1)))
```

```
##      Sum Sq   Mean Sq NumDF     DenDF   F value       Pr(>F)
## 1 229892.42 76630.807     3 31.781956 5.3434627 0.0042752274
```

The two item features and their interaction significantly predicted RT, $F(3,\ 37.8) = 5.34, p = .004$, and accounted for 29.9% of the item random intercept variance. The other variances remained unchanged relative to the empty means model, as expected given that no subject-level or trial-level predictors were added.

We can also see whether the item random intercept variance that remains is significantly > 0 using an LRT against a model without it via `ranova`:

```
print("LRT for remaining item random intercept variance"); ranova(Cond1)
```

```
## [1] "LRT for remaining item random intercept variance"
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## RT ~ freq01 + size01 + (1 | SubjectID) + (1 | ItemID) + freq01:size01
##                  npar   logLik     AIC     LRT Df Pr(>Chisq)
## <none>              7 -8701.22 17416.5
## (1 | SubjectID)     6 -8860.32 17732.7 318.202  1 < 2.22e-16
## (1 | ItemID)        6 -8733.22 17478.4  63.991  1 1.2496e-15
```

Significant item random intercept variance remained, $-2\Delta LL(1) = 64.0, p < .001$, so we will retain the level-2 item random intercept variance in our model.

Conditional Model 2: Add Random Slope over Subjects for the Freqency Item Predictor

Next, we add to the two-level crossed model a subject random slope for the item predictor freq01, U_{01s}, whose variance across subjects is then estimated to form the level-2 subject random freq01 slope variance $\tau^2_{U_{01S}}$:

$$\text{Conditional Model 2: } RT_{tis} = \gamma_{000} + \gamma_{010}(\text{freq01}_i) + \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i)$$
$$+U_{00s} + U_{010s}(\text{freq01}_i) + U_{0i0} + e_{tis}$$
$$e_{tis} \sim N(0, \sigma^2_e)$$
$$\begin{bmatrix} U_{00s} \\ U_{01s} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2_{U_{00S}} & \tau_{U_{00S,01S}} \\ \tau_{U_{00S,01S}} & \tau^2_{U_{01S}} \end{bmatrix}\right)$$
$$U_{0i0} \sim N(0, \tau^2_{U_{0I0}})$$

This new random slope allow individual differences in the extent of the RT difference between low and high frequency words (equivalently for both small and large neighborhoods). Also added is the covariance between the subject random intercepts and the subject random freq01 slopes (which is provided as a correlation in the output below).

```
print("Conditional Model 2: Add random slope across subjects for item freq")
```

```
## [1] "Conditional Model 2: Add random slope across subjects for item freq"
```

```
Cond2 = lmer(data=Example1, REML=TRUE, formula=
             RT~1+freq01+size01+freq01:size01+(1+freq01|SubjectID)+(1|ItemID))
print("Show results using Sattherwaite DDF")
```

```
## [1] "Show results using Sattherwaite DDF"
```

```
summary(Cond2, ddf="Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: RT ~ 1 + freq01 + size01 + freq01:size01 + (1 + freq01 | SubjectID) +      (1
| ItemID)
##    Data: Example1
##
## REML criterion at convergence: 17397.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.40130 -0.59429 -0.15206  0.37306  9.70966
##
## Random effects:
##  Groups    Name        Variance Std.Dev. Corr
##  ItemID    (Intercept)  1700.40  41.236
##  SubjectID (Intercept)  4266.15  65.316
##            freq01        371.71  19.280  0.693
##  Residual              14244.40 119.350
## Number of obs: 1392, groups:  ItemID, 39; SubjectID, 38
##
## Fixed effects:
##               Estimate Std. Error       df t value  Pr(>|t|)
## (Intercept)   615.8495    17.9390  55.8230 34.3303 < 2.2e-16
## freq01         69.8452    20.8595  33.5319  3.3484  0.002019
## size01          4.4433    20.4479  31.3928  0.2173  0.829381
## freq01:size01 -72.0688    29.4162  31.7842 -2.4500  0.019980
##
## Correlation of Fixed Effects:
##             (Intr) freq01 size01
## freq01      -0.499
## size01      -0.571  0.491
## freq01:sz01  0.397 -0.693 -0.695
```

```
print("LRT for subject random slope of item freq"); ranova(Cond2)
```

```
## [1] "LRT for subject random slope of item freq"
```

```
## ANOVA-like table for random-effects: Single term deletions
##
```

```
## Model:
## RT ~ freq01 + size01 + (1 + freq01 | SubjectID) + (1 | ItemID) + freq01:size01
##                                     npar   logLik     AIC     LRT Df Pr(>Chisq)
## <none>                                 9 -8698.79 17415.6
## freq01 in (1 + freq01 | SubjectID)     7 -8701.22 17416.5  4.8744  2   0.087405
## (1 | ItemID)                           8 -8731.19 17478.4 64.8086  1 8.2537e-16
```

As shown above, the fixed freq01 slope is now $\gamma_{010} = 69.8$, whereas it was previously $\gamma_{010} = 70$. It, too, has changed interpretation: It is now the mean of the subject-specific freq01 slopes. The new subject random slope variance $\tau^2_{U_{01S}} = 371.7$. The subject random slopes were positively correlated $r = 0.69$ with the subject random intercepts, indicating that subjects who had slower response times to low-frequency words (i.e., higher intercepts at freq01=0) tended to have more of a difference between low- and high-frequency words (i.e., steeper freq01 slopes).

However, the LRT generated by `ranova` indicates that the model fit did not improve from adding the new random slope variance and correlation, $-2\Delta LL(2) = 4.87, p = .087$. To use the more correct mixture χ^2 distribution instead, we would use $df = 1$ for when the random slope variance would have become negative and $df = 2$ for when it would have been positive; the covariance does not have a boundary at 0 so we keep its $df = 1$ regardless. The mixture critical value at $p < .05$ for $df = 1{,}2$ is 5.14, and we can compute an exact p-value for the mixture by weighting each p-value by 0.5 and then summing them, as shown below:

```
print("LRT for random slope variance using a mixture-chi-square test")

## [1] "LRT for random slope variance using a mixture-chi-square test"

Cond2Diff2LL = -2*(logLik(Cond1)-logLik(Cond2)); Cond2Diff2LL

## 'log Lik.' 4.8744161 (df=7)

Cond2DiffP1 = pchisq(Cond2Diff2LL, df=1, lower.tail=FALSE); Cond2DiffP1

## 'log Lik.' 0.027257662 (df=7)

Cond2DiffP2 = pchisq(Cond2Diff2LL, df=2, lower.tail=FALSE); Cond2DiffP2

## 'log Lik.' 0.087404542 (df=7)

Cond2DiffP12 = (.5*Cond2DiffP1) + (.5*Cond2DiffP2)
print("Test statistic and mixture p-values for df=1,2")

## [1] "Test statistic and mixture p-values for df=1,2"

Cond2Diff2LL; Cond2DiffP12

## 'log Lik.' 4.8744161 (df=7)

## 'log Lik.' 0.057331102 (df=7)
```

Either way, we would not retain the two new parameters (and so they are removed in the model that follows). To get a sense of the degree of slope differences (to help convey effect size), however, we can still compute a 95% random slope confidence interval as follows:

$$\text{Subject Random Freq01 Slope 95\% CI} = \gamma_{010} \pm 1.96 \times \sqrt{\tau^2_{U_{01S}}}$$

$$\text{Subject Random Freq01 Slope 95\% CI} = 69.8 \pm 1.96 \times \sqrt{371.7} = 32.1 \text{ to } 107.6$$

This indicates the expected variability in how high-frequency words differ from low-frequency words, in which 95% of the sample subjects would be expected to have a positive item predictor slope.

Below is how the random effect confidence intervals can be computed using saved output:

```
# Print stored fixed effects table for reference
as.data.frame(fixef(Cond2))
```

```
##                fixef(Cond2)
## (Intercept)    615.8495237
## freq01          69.8452105
## size01           4.4433239
## freq01:size01  -72.0687755
```

```
# Save fixed freq01 slope for use below
Cond2FreqSlp = as.data.frame(fixef(Cond2))[2,1]
# Print stored variance components table for reference
as.data.frame(VarCorr(Cond2))
```

```
##         grp        var1   var2        vcov       sdcor
## 1    ItemID (Intercept)   <NA>  1700.40425  41.2359583
## 2 SubjectID (Intercept)   <NA>  4266.14952  65.3157678
## 3 SubjectID      freq01   <NA>   371.70671  19.2796969
## 4 SubjectID (Intercept) freq01   872.17244   0.6926026
## 5  Residual        <NA>   <NA> 14244.39883 119.3499008
```

```
# Save subject random freq01 slope variance for use below
Cond2SubFreqVar = as.data.frame(VarCorr(Cond2))[3,4]
print("95% random freq01 slope confidence interval for subjects")
```

```
## [1] "95% random freq01 slope confidence interval for subjects"
```

```
SubFreqSlp_LCI = Cond2FreqSlp - 1.96*sqrt(Cond2SubFreqVar)
SubFreqSlp_UCI = Cond2FreqSlp + 1.96*sqrt(Cond2SubFreqVar)
print("Subject freq01 slope lower CI = "); SubFreqSlp_LCI
```

```
## [1] "Subject freq01 slope lower CI = "
```

```
## [1] 32.057005
```

```
print("Subject freq01 slope upper CI = "); SubFreqSlp_UCI
```

```
## [1] "Subject freq01 slope upper CI = "
```

```
## [1] 107.63342
```

Conditional Model 3: Add Random Slope over Subjects for the Neighborhood Item Predictor

Lastly, we add to the two-level crossed model a subject random slope for the item predictor size01, U_{02S}, whose variance across subjects is then estimated to form the level-2 subject random size01 slope variance $\tau^2_{U_{02S}}$:

Conditional Model 3: $RT_{tis} = \gamma_{000} + \gamma_{010}(\text{freq01}_i) + \gamma_{020}(\text{size01}_i) + \gamma_{030}(\text{freq01}_i)(\text{size01}_i)$
$+U_{00s} + U_{020s}(\text{size01}_i) + U_{0i0} + e_{tis}$
$e_{tis} \sim N(0, \sigma_e^2)$
$\begin{bmatrix} U_{00s} \\ U_{02s} \end{bmatrix} \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2_{U_{00S}} & \tau_{U_{00S,02S}} \\ \tau_{U_{00S,02S}} & \tau^2_{U_{02S}} \end{bmatrix}\right)$
$U_{0i0} \sim N(0, \tau^2_{U_{0I0}})$

This new random slope allow individual differences in the extent of the RT difference between small and large neighborhood-size words (equivalently for both low and high frequency words). Also added is the covariance between the subject random intercepts and the subject random size01 slopes (which is provided as a correlation in the output below).

```
print("Conditional Model 3: Add random slope across subjects for item size")
```

```
## [1] "Conditional Model 3: Add random slope across subjects for item size"
```

```
Cond3 = lmer(data=Example1, REML=TRUE, formula=
             RT~1+freq01+size01+freq01:size01+(1+size01|SubjectID)+(1|ItemID))
print("Show results using Sattherwaite DDF")
```

```
## [1] "Show results using Sattherwaite DDF"
```

```
summary(Cond3, ddf="Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
## Formula: RT ~ 1 + freq01 + size01 + freq01:size01 + (1 + size01 | SubjectID) +      (1
| ItemID)
##    Data: Example1
##
## REML criterion at convergence: 17402.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.27766 -0.59622 -0.15598  0.37985  9.68269
##
## Random effects:
##  Groups    Name        Variance  Std.Dev. Corr
##  ItemID    (Intercept)  1692.994  41.1460
##  SubjectID (Intercept)  5103.813  71.4410
##            size01         84.446   9.1895 0.067
##  Residual               14319.258 119.6631
## Number of obs: 1392, groups:  ItemID, 39; SubjectID, 38
##
## Fixed effects:
##               Estimate Std. Error       df t value  Pr(>|t|)
## (Intercept)   615.7996    18.5290  57.9654 33.2344 < 2.2e-16
## freq01         70.0007    20.5968  32.4030  3.3986  0.001811
## size01          4.4155    20.4763  31.4448  0.2156  0.830662
## freq01:size01 -72.0237    29.3780  31.7830 -2.4516  0.019904
##
## Correlation of Fixed Effects:
##             (Intr) freq01 size01
## freq01      -0.548
```

```
## size01      -0.548  0.496
## freq01:sz01  0.384 -0.701 -0.693
```

```
print("LRT for subject random slope of item freq"); ranova(Cond3)
```

```
## [1] "LRT for subject random slope of item freq"
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## RT ~ freq01 + size01 + (1 + size01 | SubjectID) + (1 | ItemID) + freq01:size01
##                                    npar   logLik     AIC     LRT Df Pr(>Chisq)
## <none>                                9 -8701.18 17420.4
## size01 in (1 + size01 | SubjectID)    7 -8701.22 17416.5  0.0817  2    0.95999
## (1 | ItemID)                          8 -8733.21 17482.4 64.0637  1 1.2046e-15
```

As shown above, the fixed size01 slope is now $\gamma_{020} = 4.42$, whereas it was previously $\gamma_{020} = 4.43$. It, too, has changed interpretation: It is now the mean of the subject-specific size01 slopes. The new subject random slope variance $\tau^2_{U_{02S}}$ was estimated as 84.45. The subject random slopes were positively correlated $r = 0.07$ with the subject random intercepts, indicating that subjects who had slower response times to small-neighborhood-size words (i.e., higher intercepts) tended to have slightly more of a difference between small- and large-neighborhood-size words (i.e., steeper slopes).

However, the LRT generated by `ranova` indicates that the model fit did not improve from adding the new random slope variance and covariance, $-2\Delta LL(2) = 0.08, p = .960$, so we will not retain them in our model. Even if using a mixture chi-square distribution instead (whose critical value at $p < .05$ for $df = 1,2$ would be 5.14 instead of 5.99 for $df = 2$), we would not retain the two new parameters, as shown below:

```
print("LRT for random slope variance using a mixture-chi-square test")
```

```
## [1] "LRT for random slope variance using a mixture-chi-square test"
```

```
Cond3Diff2LL = -2*(logLik(Cond1)-logLik(Cond3)); Cond3Diff2LL
```

```
## 'log Lik.' 0.081670482 (df=7)
```

```
Cond3DiffP1 = pchisq(Cond3Diff2LL, df=1, lower.tail=FALSE); Cond3DiffP1
```

```
## 'log Lik.' 0.77504626 (df=7)
```

```
Cond3DiffP2 = pchisq(Cond3Diff2LL, df=2, lower.tail=FALSE); Cond3DiffP2
```

```
## 'log Lik.' 0.95998728 (df=7)
```

```
Cond3DiffP12 = (.5*Cond3DiffP1) + (.5*Cond3DiffP2)
print("Test statistic and mixture p-values for df=1,2")
```

```
## [1] "Test statistic and mixture p-values for df=1,2"
```

```
Cond3Diff2LL; Cond3DiffP12
```

```
## 'log Lik.' 0.081670482 (df=7)
```

```
## 'log Lik.' 0.86751677 (df=7)
```

To get a sense of the degree of slope differences, however, we can still compute a 95% random slope confidence interval as follows:

$$\text{Subject Random Size Slope 95\% CI} = \gamma_{020} \pm 1.96 \times \sqrt{\tau^2_{U_{02S}}}$$

$$\text{Subject Random Size Slope 95\% CI} = 4.42 \pm 1.96 \times \sqrt{84.45} = -13.6 \text{ to } 22.4$$

This indicates the expected variability in how small-neighborhood words differ from large-neighborhood words, in which 95% of the sample would be expected to have slopes ranging from negative to positive slope.

Below is how the random effect confidence intervals can be computed using saved output:

```
# Print stored fixed effects table for reference
as.data.frame(fixef(Cond3))
```

```
##                fixef(Cond3)
## (Intercept)    615.7995689
## freq01          70.0007436
## size01           4.4154652
## freq01:size01  -72.0237185
```

```
# Save fixed freq01 slope for use below
Cond3SizeSlp = as.data.frame(fixef(Cond3))[3,1]
# Print stored variance components table for reference
as.data.frame(VarCorr(Cond3))
```

```
##         grp        var1   var2         vcov        sdcor
## 1    ItemID (Intercept)   <NA>  1692.993504  41.14600228
## 2 SubjectID (Intercept)   <NA>  5103.812955  71.44097532
## 3 SubjectID      size01   <NA>    84.446198   9.18946126
## 4 SubjectID (Intercept) size01    44.183177   0.06730069
## 5  Residual        <NA>   <NA> 14319.257502 119.66310000
```

```
# Save subject random size01 slope variance for use below
Cond3SubSizeVar = as.data.frame(VarCorr(Cond3))[3,4]
print("95% random size01 slope confidence interval for subjects")
```

```
## [1] "95% random size01 slope confidence interval for subjects"
```

```
SubSizeSlp_LCI = Cond3SizeSlp - 1.96*sqrt(Cond3SubSizeVar)
SubSizeSlp_UCI = Cond3SizeSlp + 1.96*sqrt(Cond3SubSizeVar)
print("Subject size01 slope lower CI = "); SubSizeSlp_LCI
```

```
## [1] "Subject size01 slope lower CI = "
```

```
## [1] -13.595879
```

```
print("Subject size01 slope upper CI = "); SubSizeSlp_UCI
```

```
## [1] "Subject size01 slope upper CI = "
```

```
## [1] 22.426809
```

Sample Results Section

The extent to which semantic frequency (coded low = 0, high = 1) and phonological neighborhood size (coded small = 0, large = 1) could predict response time (RT) in milliseconds in a lexical decision task was examined for 39 items administered to 38 subjects. Because RTs for incorrect responses were not included, the data were unbalanced, such that each subject had a different number of trials included for each condition. Accordingly, rather than aggregating the individual trial RTs into potentially biased item condition means (that would assume items are fixed) and conducting an analysis of variance, all possible RTs were examined instead in a multilevel model with crossed random effects for subjects and items, in which level-1 trials (i.e., the unique combination of each subject with each item) were nested within level-2 subjects and within level-2 items (as crossed random factors). Restricted maximum likelihood within the R function `lmer` from the `lme4` was used to estimate all models; denominator degrees of freedom were estimated with the Satterthwaite method using the package `lmerTest`. Likelihood ratio tests (i.e., the difference in model $-2LL$ values) were used to evaluate new random effect variances and covariances, in which a mixture of χ^2 distributions (with the two mixture degrees of freedom given in parentheses below) was used to determine the significance of the new random effect variances bounded at 0.

The extent to which systematic variability in mean RT existed for each dimension of sampling was first examined in a series of empty means models (i.e., only a fixed intercept and no predictors). Relative to a model with only a residual variance, the addition of a random intercept variance for subjects significantly improved model fit, $-2\Delta LL(0,1) = 280.4, p < .001$, indicating significant differences between subjects in mean RT, and that trials from the same subject were positively correlated. The addition of a random intercept for items also significantly improved model fit, $-2\Delta LL(0,1) = 100.4, p < .001$, indicating significant differences between items in mean RT as well, and that trials for the same item were also positively correlated. Of the total estimated RT variance, 24% was due to between-subject differences in mean RT (given by the subject random intercept), 11% was due to between-item differences in mean RT (given by the item random intercept), and the remaining 65% was due to the subject by item interaction (i.e., residual variance). Construction of 95% random intercept confidence intervals as described in Snijders and Bosker (2012) revealed that 95% of subject mean RTs are expected to fall between 494 and 776 ms, whereas 95% of the item mean RTs are expected to fall between 539 and 732 ms. Thus, there was relatively more variability across subjects than across items. The extent to which the main and interaction effects of semantic frequency and neighborhood size could account for between-item differences in mean RT was then examined in a conditional model; results are provided in Table 1.

ANOVA-like description of the results: There was a significant semantic frequency by neighborhood size interaction, $t(31.8) = -2.45, p = .0199$; the pattern of the interaction is shown in Figure 1 and was interpreted by examining simple main effects of each predictor. First, with respect to the effect of frequency, for small neighborhood words, responses were significantly faster to words of low than high frequency ($M = 615.8, M = 685.8$), $t(32.4) = 3.40, p = .002$, whereas for large neighborhood words, there was no significant difference between words of low or high frequency ($M = 620.2, M = 618.2$), $t(31.2) = -0.096, p = .924$. Second, with respect to the effect of neighborhood size, for low frequency words, there was no significant difference between words with small or large neighborhood size ($M = 615.8, M = 620.2$), $t(31.4) = 0.22, p = .829$, whereas for high frequency words, responses were significantly slower to words with smaller than larger neighborhoods ($M = 685.8, M = 618.2$), $t(32.2) = -3.20, p = .003$.

Regression-like description of the same results, which refer to the linear model equation: The fixed intercept for the predicted RT for a word of low frequency and small size was $\gamma_{000} = 615.8$. There was a significant simple main effect for the mean difference between low and high frequency words of small size of $\gamma_{010} = 70.0$ ($p = .002$). There was a nonsignificant simple main effect for the mean difference

between small and large size words of low frequency of $\gamma_{020} = 4.4$ ($p = .829$). However, there was a significant frequency by size interaction of $\gamma_{030} = -72.0$ ($p = .020$), such that relative to the frequency effect for small words of $\gamma_{010} = 70.0$, the frequency effect for large words was significantly less positive by $\gamma_{030} = -72.0$ (yielding a nonsignificant simple effect of frequency for large words of $\gamma_{010} + \gamma_{030} = -2.0$, $p = .924$). Similarly, relative to the size effect for low frequency words of $\gamma_{010} = 4.4$, the size effect for high frequency words was significantly more negative by $\gamma_{030} = -72.0$ (yielding a significant simple effect of size for high frequency words of $\gamma_{020} + \gamma_{030} = -67.56$, $p = .003$). Thus, as shown in Figure 1, a positive frequency effect was found only for words of small size, and a negative size effect was found only for high frequency words.

The effects of frequency and size explained approximately 30% of the item intercept variance. Given that 11% of the total RT variance was due to mean differences between items, this translates into a total reduction in all RT variance of 3.28%. The extent to which these effects were sufficient to describe all between-item differences in mean RT was then examined by removing the item random intercept variance from the conditional model. The resulting significant decrease in model fit, $-2\Delta LL(1) = 64.4$, $p < .001$, suggesting that significant differences remain between items after controlling for their primary design features, or that items should not be treated as fixed effects.

Finally, the potential for individual subject differences in the frequency slope was examined by adding a random subject frequency slope (and its covariance with the subject random intercept) to the model. Model fit did not significantly improve, $-2\Delta LL(1,2) = 4.87$, $p = .057$, indicating that each subject does not need their own random deviation from the fixed effect of frequency. Likewise, the potential for individual subject differences in the neighborhood size slope was examined by adding a random subject size slope (and its covariance with the subject random intercept) to the model. Model fit did not significantly improve, $-2\Delta LL(1,2) = 0.08$, $p = .867$, indicating that each subject does not need their own random deviation from the fixed effect of size, either.