m21_202303

Joanna Morris

2023-03-17

Compute PCA

This script computes one ANOVA for all non-words,

Following Andrews and Lo (2013) this script computes a PCA for our spelling and vocabulary measures. Because the standardised spelling and vocabulary scores were correlated, to facilitate interpretation, two orthogonal measures of individual differences were derived from a principal components analysis. Analysis based on this tutorial

```
library(readr)
library(dplyr)
library(datawizard)
sv_202303 <- read_csv("m21_spell_vocab_raw.csv")
sv_202303.na <- na.omit(sv_202303)
sv_202303.na <- mutate(sv_202303.na,
                       z_ART = standardise(ART_correct),
                       z_vocab = standardise(vocab_correct),
                       z_spell = standardise(spell_correct))

cor.test(sv_202303.na$z_vocab, sv_202303.na$z_spell)
```

```

    Pearson's product-moment correlation

data:  sv_202303.na$z_vocab and sv_202303.na$z_spell
t = 1.9352, df = 61, p-value = 0.05761
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.007724833  0.460807138
sample estimates:
      cor
0.2405005
```

By default, the function `PCA()` in `FactoMineR`, standardizes the data automatically during the PCA; so you don't need do this transformation before the PCA.

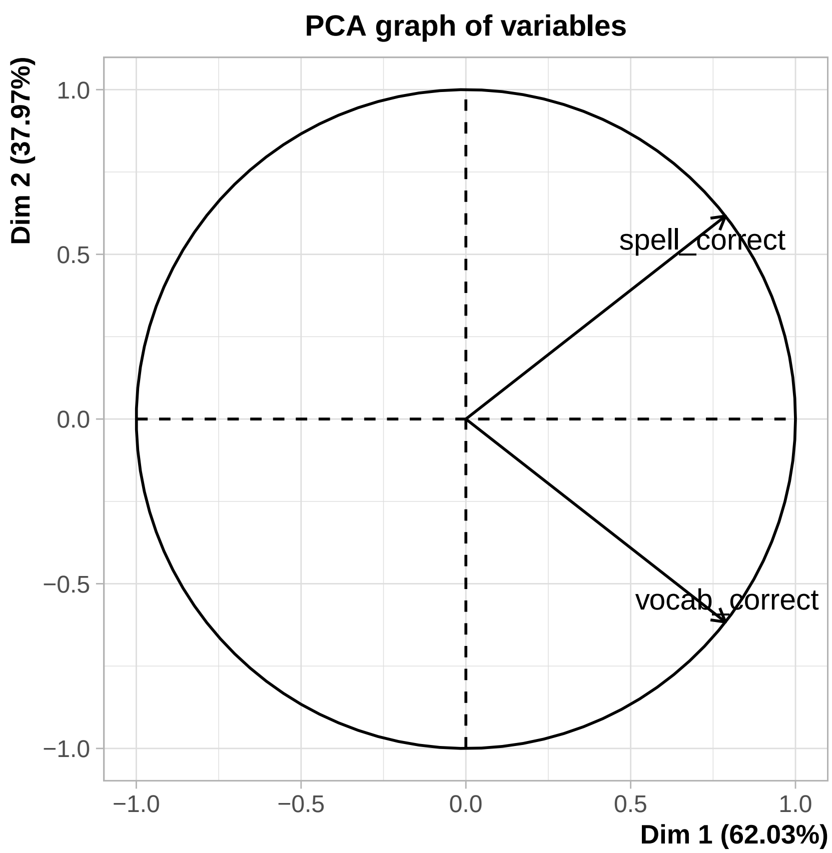
- `X`: a data frame. Rows are individuals and columns are numeric variables
- `scale.unit`: a logical value. If TRUE, the data are scaled to unit variance before the analysis. This standardization to the same scale avoids some variables to become dominant just because of their large measurement units. It makes variable comparable.

- `ncp`: number of dimensions kept in the final results.
- `graph`: a logical value. If TRUE a graph is displayed.

The plot shows the relationships between all variables. It can be interpreted as follow:

- Positively correlated variables are grouped together.
- Negatively correlated variables are positioned on opposite sides of the plot origin (opposed quadrants).
- The distance between variables and the origin measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map.

```
library(FactoMineR)
library(factoextra)

res.pca <- PCA(sv_202303.na[,3:4], scale.unit = TRUE, ncp = 2, graph = FALSE)
plot(res.pca, choix = "varcor", graph.type = c("ggplot"))
```



The eigenvalues measure the amount of variation retained by each principal component. Eigenvalues are large for the first PCs and small for the subsequent PCs. That is, the first PCs corresponds to the directions with the maximum amount of variation in the data set.

We examine the eigenvalues to determine the number of principal components to be considered

```
(eig.val <- get_eigenvalue(res.pca))
```

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1  1.2405005         62.02503                    62.02503
Dim.2  0.7594995         37.97497                   100.00000
```

The quality of representation of the variables on factor map is called cos2 (square cosine, squared coordinates). A high cos2 indicates a good representation of the variable on the principal component. In this case the variable is positioned close to the circumference of the correlation circle. A low cos2 indicates that the variable is not perfectly represented by the PCs. In this case the variable is close to the center of the circle. For a given variable, the sum of the cos2 on all the principal components is equal to one. If a variable is perfectly represented by only two principal components (Dim.1 & Dim.2), the sum of the cos2 on these two PCs is equal to one. In this case the variables will be positioned on the circle of correlations.

```r
res.pca$var$cos2
```

```
                  Dim.1     Dim.2
vocab_correct 0.6202503 0.3797497
spell_correct 0.6202503 0.3797497
```

The contributions of variables in accounting for the variability in a given principal component are expressed in percentage. Variables that are correlated with PC1 (i.e., Dim.1) and PC2 (i.e., Dim.2) are the most important in explaining the variability in the data set. Variables that do not correlated with any PC or correlated with the last dimensions are variables with low contribution and might be removed to simplify the overall analysis.

```r
res.pca$var$contrib
```

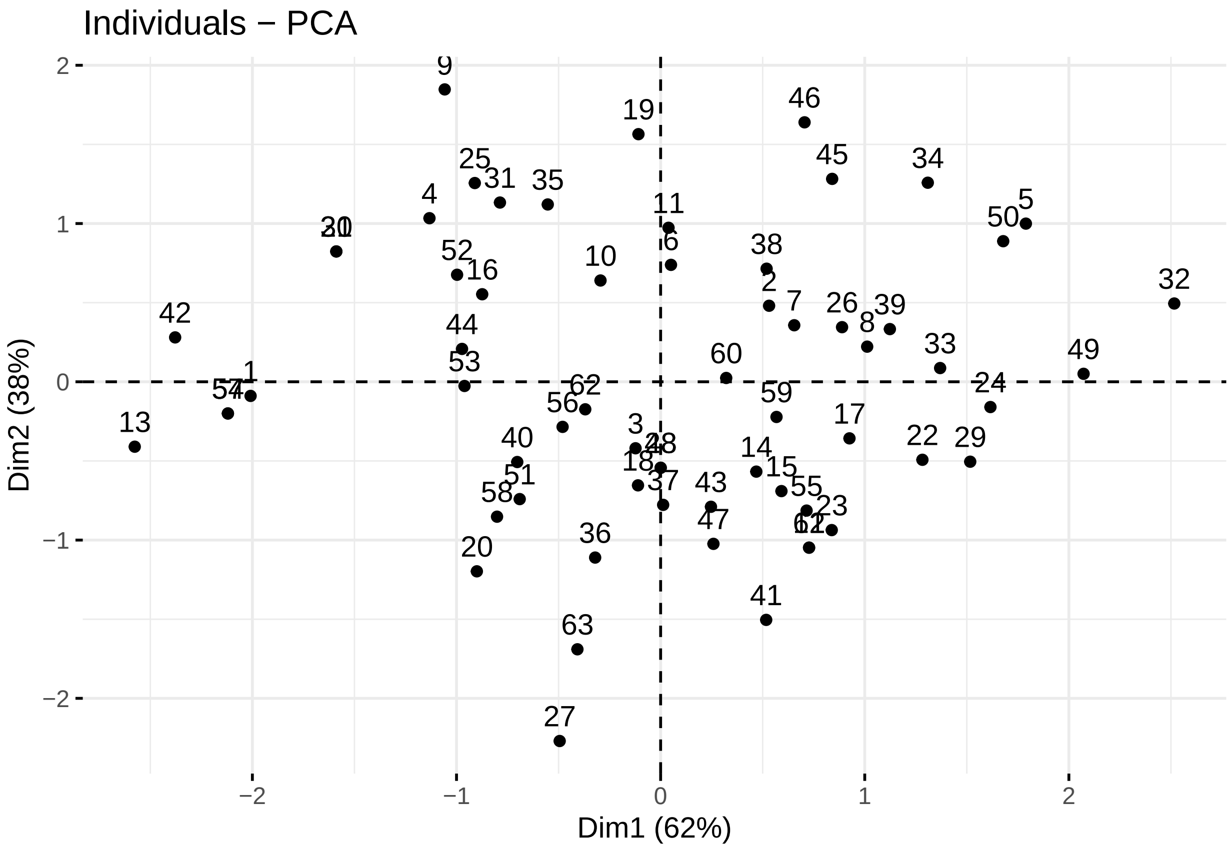
```
              Dim.1 Dim.2
vocab_correct    50    50
spell_correct    50    50
```

```r
(res.desc <- dimdesc(res.pca, axes = c(1,2), proba = 0.05))
```

```
$Dim.1

Link between the variable and the continuous variables (R-square)
=================================================================================
              correlation      p.value
vocab_correct   0.7875597 1.913196e-14
spell_correct   0.7875597 1.913196e-14

$Dim.2

Link between the variable and the continuous variables (R-square)
=================================================================================
              correlation      p.value
spell_correct   0.6162384 7.592937e-08
vocab_correct  -0.6162384 7.592937e-08
```

The fviz_pca_ind() is used to produce the graph of individuals.

```
fviz_pca_ind(res.pca)
```




```
sv_202303.na<-bind_cols(sv_202303.na,res.pca$ind$coord)
```

Loads RT data and join to PCA dataset

```
cw_frq <- read_csv("CW_frq.csv")
nw_frq <- read_csv("NW_frq.csv")

CW_rt <- read_csv("CW_rt_2.csv")
CW_rt$cw_target <- NULL
CW_rt <- rename(CW_rt, cw_target = target_lower)

NW_rt <- read_csv("NW_rt_2.csv")
NW_rt$nw_target <- NULL
NW_rt <- rename(NW_rt, nw_target = target_lower)

cw_rt_pca <- inner_join(sv_202303.na, CW_rt, by = "SubjID")  #join subject PCA data
nw_rt_pca <- inner_join(sv_202303.na, NW_rt, by = "SubjID")

cw <- left_join(cw_rt_pca, cw_frq, by = c("cw_target")) #join word frequency data
nw <- left_join(nw_rt_pca, nw_frq, by = c("nw_target"))
```

Divide participants based on median split of Dim2. Higher values on this factor indicate that spelling scores were relatively higher than vocabulary,

```
cw.median <- median(cw$Dim.2)
cw <- cw |>
    mutate(lang_type = case_when(
      Dim.2 <= cw.median ~ "Semantic",
      Dim.2 > cw.median ~ "Orthographic"
    ))

nw.median <- median(nw$Dim.2)
nw <- nw |>
    mutate(lang_type = case_when(
      Dim.2 <= nw.median ~ "Semantic",
      Dim.2 > nw.median ~ "Orthographic"
    ))
```

```
cols <- c( "cw_famsize", "lang_type")  # recode ind variable columns as factors
cw <- cw |> mutate_at(cols, factor)
cw$cw_famsize <- recode_factor(cw$cw_famsize, S = "Small", L = "Large")


cols <- c( "cw_famsize", "lang_type")
cw <- cw |> mutate_at(cols, factor)
cw$cw_famsize <- recode_factor(cw$cw_famsize, S = "Small", L = "Large")


cols <- c( "nw_famsize", "lang_type", "complexity")
nw <- nw |> mutate_at(cols, factor)
nw$nw_famsize <- recode_factor(nw$nw_famsize, S = "Small", L = "Large")
nw$complexity <- recode_factor(nw$complexity, SIMP = "Simple", COMP = "Complex")


rm(CW_rt)  #remove original rt file after joining neuropsych data
rm(NW_rt)
```

Removes rts for errors (column rt.err) and then imputes missing values with the mean for the dataset (column "rt.err.imp") then creates a new column with inverse RTs

```
library(tidyr)
cw <- cw |> mutate(rt.err = response_time * correct)  # convert error rts to 0
cw <- cw |> mutate(rt.err = na_if(rt.err, 0))         # convert 0 rts to NA
cw.mean <- mean(cw$rt.err, na.rm = TRUE)  # get mean rt excluding errors
cw <- cw |> mutate(rt.err.imp = ifelse(is.na(rt.err),
                                       cw.mean,
                                       rt.err))  # replace missing values with mean
cw <- cw |> mutate(inv.rt = 1/rt.err.imp)  # creates new column with inverse RTs


nw <- nw |> mutate(rt.err = response_time * correct)  # convert error rts to 0
nw <- nw |> mutate(rt.err = na_if(rt.err, 0))         # convert 0 rts to NA
nw.mean <- mean(nw$rt.err, na.rm = TRUE)  # get mean rt excluding errors
nw <- nw |> mutate(rt.err.imp = ifelse(is.na(rt.err),
                                       nw.mean,
                                       rt.err))  # replace missing values with mean
nw <- nw |> mutate(inv.rt = 1/rt.err.imp)  # creates new column with inverst RTs
```

Determines how much missing data there is. Creates new dataframe with just the non-missing data

```
cw_missing_data<- filter(cw, is.na(cw$rt.err))
cw_na.omit <- filter(cw, !is.na(cw$rt.err))
(xtab.missing.data <- xtabs(~cw_famsize+lang_type, data=cw_missing_data))
```

```
          lang_type
cw_famsize Orthographic Semantic
     Small          206      176
     Large          224      164
```

```
nw_missing_data<- filter(nw, is.na(nw$rt.err))
nw_na.omit <- filter(nw, !is.na(nw$rt.err))
(xtab.missing.data <- xtabs(~nw_famsize+lang_type+complexity, data=nw_missing_data))
```

```
, , complexity = Simple

          lang_type
nw_famsize Orthographic Semantic
     Small          194      225
     Large          195      277

, , complexity = Complex

          lang_type
nw_famsize Orthographic Semantic
     Small          215      237
     Large          233      264
```

With RT as dependent variable

```
library(ez)
library(car)
(m.cw <- ezANOVA(cw,
        dv = rt.err.imp,
        wid = SubjID,
        within = cw_famsize,
        between = lang_type))
```

```
Warning: Converting "SubjID" to factor for ANOVA.

Warning: Data is unbalanced (unequal N per group). Make sure you specified a
well-considered value for the type argument to ezANOVA().

Warning: Collapsing data to cell means. *IF* the requested effects are a subset
of the full design, you must use the "within_full" argument, else results may be
inaccurate.

$ANOVA
               Effect DFn DFd         F            p p<.05          ges
2           lang_type   1  58  3.020685 8.751568e-02       0.0480026898
3          cw_famsize   1  58 29.284656 1.242722e-06     * 0.0158162335
4 lang_type:cw_famsize   1  58  0.811432 3.714225e-01       0.0004450876
```

```r
(m.nw <- ezANOVA(nw,
        dv = rt.err.imp,
        wid = SubjID,
        within = .(nw_famsize, complexity),
        between = lang_type))
```

```
Warning: Converting "SubjID" to factor for ANOVA.

Warning: Data is unbalanced (unequal N per group). Make sure you specified a
well-considered value for the type argument to ezANOVA().

Warning: Collapsing data to cell means. *IF* the requested effects are a subset
of the full design, you must use the "within_full" argument, else results may be
inaccurate.

$ANOVA
                           Effect DFn DFd         F          p p<.05
2                       lang_type   1  58 2.5860422 0.11323999
3                      nw_famsize   1  58 2.4798655 0.12075159
5                      complexity   1  58 1.0997591 0.29866934
4            lang_type:nw_famsize   1  58 0.5865729 0.44685318
6            lang_type:complexity   1  58 5.0489637 0.02846111     *
7           nw_famsize:complexity   1  58 0.3480002 0.55753865
8 lang_type:nw_famsize:complexity   1  58 0.3797920 0.54012589
           ges
2 0.0359716096
3 0.0014164494
5 0.0017879856
4 0.0003354013
6 0.0081562251
7 0.0002128401
8 0.0002322797
```

Get condition means

```r
#Define standard error of the mean function

sem <- function(x) sd(x)/sqrt(length(x))

(cw.cond.means <- cw |>
   group_by(cw_famsize, lang_type) |>
   summarise(mean = mean(rt.err.imp),
             se = sem(rt.err.imp),
             num_stim = n()))
```

```
# A tibble: 4 x 5
# Groups:   cw_famsize [2]
  cw_famsize lang_type     mean    se num_stim
  <fct>      <fct>        <dbl> <dbl>    <int>
1 Small      Orthographic  597.  3.15     1450
2 Small      Semantic      628.  3.28     1506
3 Large      Orthographic  584.  3.17     1450
4 Large      Semantic      609.  3.14     1515
```

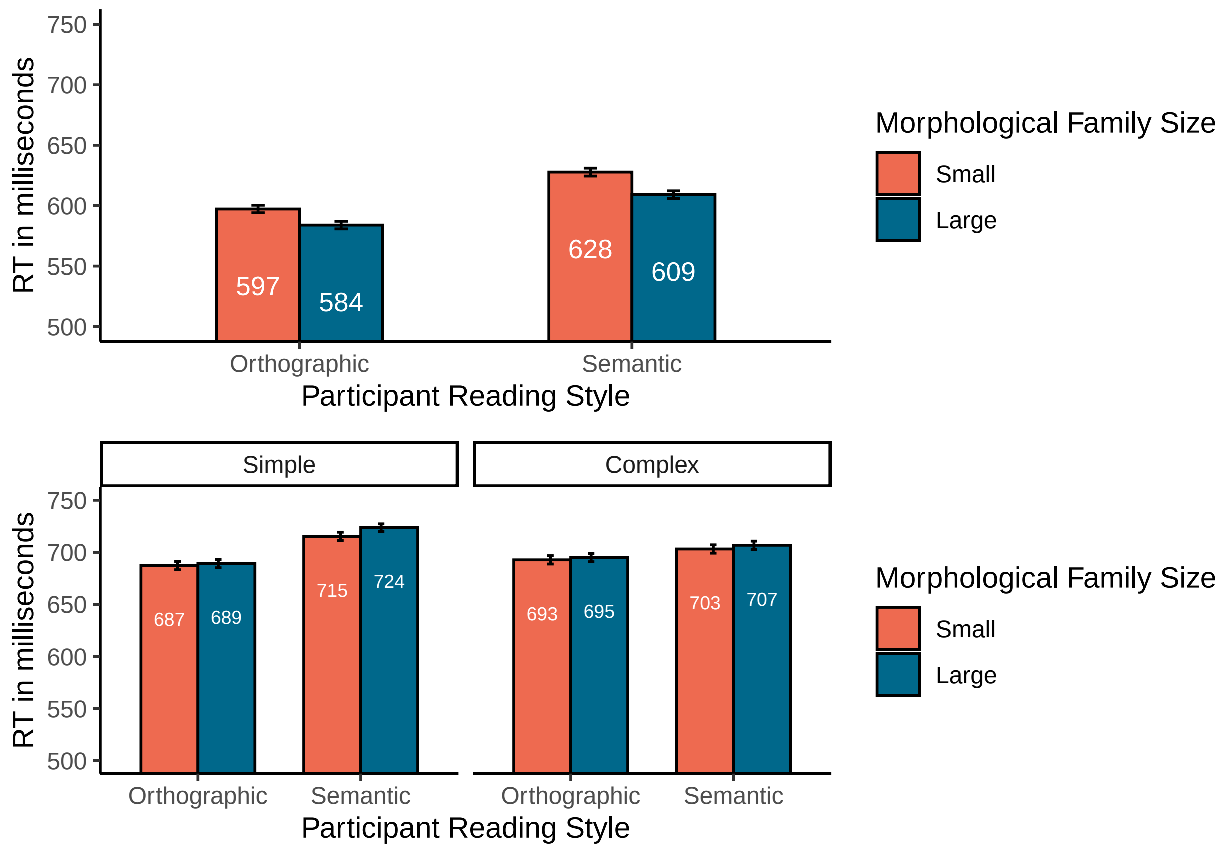
```
(nw.cond.means <- nw |>
    group_by(nw_famsize, complexity, lang_type) |>
    summarise(mean = mean(rt.err.imp),
              se = sem(rt.err.imp),
              num_stim = n()))
```

```
# A tibble: 8 x 6
# Groups:   nw_famsize, complexity [4]
  nw_famsize complexity lang_type     mean    se num_stim
  <fct>      <fct>      <fct>        <dbl> <dbl>    <int>
1 Small      Simple     Orthographic  687.  4.05      725
2 Small      Simple     Semantic      715.  4.06      755
3 Small      Complex    Orthographic  693.  3.98      725
4 Small      Complex    Semantic      703.  3.97      761
5 Large      Simple     Orthographic  689.  4.05      725
6 Large      Simple     Semantic      724.  3.59      757
7 Large      Complex    Orthographic  695.  3.96      725
8 Large      Complex    Semantic      707.  3.97      762
```

Barplots

```
library(gridExtra)
p1 <-  cw.cond.means %>% ggplot(aes(x=lang_type,
                                    y=mean,
                                    fill = cw_famsize,
                                    ymin = mean - se,
                                    ymax = mean + se)) +
  coord_cartesian(xlim = NULL,
                  ylim = c(500, 750),
                  expand = TRUE,
                  default = FALSE,
                  clip = "on") +
  geom_col(position = "dodge",
           width = 0.5,
           color = "black")  +
  ylab("RT in milliseconds")  +
  xlab("Participant Reading Style")  +
  scale_fill_manual(values = c("coral2", "deepskyblue4"))+
  geom_errorbar(width = .08,
                position = position_dodge(0.5)) +
  theme_classic() +
   geom_text(aes(label = round(mean, digits = 0)),
             colour = "white",
             size = 3.5,
             vjust = 4.5,
             position = position_dodge(.5))+
  guides(fill=guide_legend(title="Morphological Family Size"))

p2 <-  nw.cond.means %>% ggplot(aes(x=lang_type,
                                    y=mean,
                                    fill = nw_famsize,
                                    ymin = mean - se,
                                    ymax = mean + se)) +
```

```
  coord_cartesian(xlim = NULL, ylim = c(500, 750),
                  expand = TRUE,
                  default = FALSE,
                  clip = "on") +
  geom_col(position = "dodge",
           width = .7,
           color = "black")  +
  xlab("Participant Reading Style")  +
  ylab("RT in milliseconds")  +
  scale_fill_manual(values = c("coral2", "deepskyblue4"))+
  geom_errorbar(width = .08,
                position = position_dodge(0.5)) +
  theme_classic() +
   geom_text(aes(label = round(mean, digits = 0)),
             colour = "white",
             size = 2.5,
             vjust = 4.5,
             position = position_dodge(.7)) +
  facet_grid(.~complexity) +
  guides(fill=guide_legend(title="Morphological Family Size"))

grid.arrange(p1, p2)
```



LME

Models

```r
library(lme4)

cw_null.model = lmer(rt.err.imp ~ 1 + (1|SubjID) + (1|cw_target), data= cw, REML=FALSE)
summary(cw_null.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: rt.err.imp ~ 1 + (1 | SubjID) + (1 | cw_target)
   Data: cw

     AIC      BIC   logLik deviance df.resid
 72112.9  72139.7 -36052.5  72104.9     5917

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2785 -0.6813 -0.1465  0.4764  4.9695

Random effects:
 Groups    Name        Variance Std.Dev.
 cw_target (Intercept)   576.8   24.02
 SubjID    (Intercept)  3982.7   63.11
 Residual              10713.6  103.51
Number of obs: 5921, groups:  cw_target, 100; SubjID, 60

Fixed effects:
            Estimate Std. Error t value
(Intercept)  605.232      8.601   70.37
```

```r
# Main effects models with random intercepts
cw_main.model = lmer(rt.err.imp ~ lang_type + cw_famsize + (1|SubjID) + (1|cw_target),
                             data= cw, REML=FALSE)
summary(cw_main.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: rt.err.imp ~ lang_type + cw_famsize + (1 | SubjID) + (1 | cw_target)
   Data: cw

     AIC      BIC   logLik deviance df.resid
 72104.9  72145.1 -36046.5  72092.9     5915

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2878 -0.6784 -0.1493  0.4726  4.9400

Random effects:
 Groups    Name        Variance Std.Dev.
 cw_target (Intercept)   511.3   22.61
 SubjID    (Intercept)  3781.6   61.50
```

```
 Residual              10713.5  103.51
Number of obs: 5921, groups:  cw_target, 100; SubjID, 60

Fixed effects:
                  Estimate Std. Error t value
(Intercept)        598.646     12.089  49.522
lang_typeSemantic   28.388     16.116   1.762
cw_famsizeLarge    -16.129      5.262  -3.065

Correlation of Fixed Effects:
            (Intr) lng_tS
lng_typSmnt -0.688
cw_famszLrg -0.218  0.000
```

```
# Interaction effects models with random intercepts
cw_inter.model = lmer(rt.err.imp ~ lang_type * cw_famsize + (1|SubjID) + (1|cw_target),
                              data= cw, REML=FALSE)
summary(cw_inter.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: rt.err.imp ~ lang_type * cw_famsize + (1 | SubjID) + (1 | cw_target)
   Data: cw

     AIC      BIC   logLik deviance df.resid
 72105.9  72152.7 -36045.9  72091.9     5914

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3011 -0.6791 -0.1469  0.4727  4.9275

Random effects:
 Groups    Name        Variance Std.Dev.
 cw_target (Intercept)   511.4   22.61
 SubjID    (Intercept)  3781.8   61.50
 Residual              10711.6  103.50
Number of obs: 5921, groups:  cw_target, 100; SubjID, 60

Fixed effects:
                                  Estimate Std. Error t value
(Intercept)                        597.234     12.166  49.089
lang_typeSemantic                   31.164     16.340   1.907
cw_famsizeLarge                    -13.306      5.936  -2.242
lang_typeSemantic:cw_famsizeLarge   -5.535      5.382  -1.028

Correlation of Fixed Effects:
            (Intr) lng_tS cw_fmL
lng_typSmnt -0.693
cw_famszLrg -0.244  0.076
lng_typS:_L  0.113 -0.165 -0.463
```

```
nw_null.model = lmer(rt.err.imp ~ 1 + (1|SubjID) + (1|nw_target), data= nw, REML=FALSE)
summary(nw_null.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
```

```
Formula: rt.err.imp ~ 1 + (1 | SubjID) + (1 | nw_target)
   Data: nw

     AIC      BIC   logLik deviance df.resid
 71128.8  71155.6 -35560.4  71120.8     5931

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4052 -0.6290 -0.1343  0.5657  4.0441

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  527.9   22.98
 SubjID    (Intercept) 2471.5   49.71
 Residual              8754.2   93.56
Number of obs: 5935, groups:  nw_target, 199; SubjID, 60

Fixed effects:
            Estimate Std. Error t value
(Intercept)  701.748      6.733   104.2
```

```
# Main effects models with random intercepts
nw_main.model = lmer(rt.err.imp ~ lang_type + nw_famsize + complexity +(1|SubjID) + (1|nw_target),
                             data= nw, REML=FALSE)
summary(nw_main.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: rt.err.imp ~ lang_type + nw_famsize + complexity + (1 | SubjID) +
    (1 | nw_target)
   Data: nw

     AIC      BIC   logLik deviance df.resid
 71129.4  71176.2 -35557.7  71115.4     5928

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4021 -0.6280 -0.1327  0.5621  4.0286

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  517.5   22.75
 SubjID    (Intercept) 2360.4   48.58
 Residual              8753.8   93.56
Number of obs: 5935, groups:  nw_target, 199; SubjID, 60

Fixed effects:
                  Estimate Std. Error t value
(Intercept)        691.433      9.756  70.875
lang_typeSemantic   20.977     12.787   1.640
nw_famsizeLarge      4.256      4.040   1.053
complexityComplex   -5.324      4.040  -1.318

Correlation of Fixed Effects:
            (Intr) lng_tS nw_fmL
```

```
lng_typSmnt -0.677
nw_famszLrg -0.207  0.000
cmplxtyCmpl -0.205  0.000 -0.004
```

```r
# Interaction effects models with random intercepts
nw_inter.model = lmer(rt.err.imp ~ lang_type * nw_famsize * complexity + (1|SubjID) + (1|nw_target),
                              data= nw, REML=FALSE)
summary(nw_inter.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: rt.err.imp ~ lang_type * nw_famsize * complexity + (1 | SubjID) +
    (1 | nw_target)
   Data: nw

     AIC      BIC   logLik deviance df.resid
 71122.0  71195.6 -35550.0  71100.0     5924

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3716 -0.6277 -0.1277  0.5711  4.0984

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  512.4   22.64
 SubjID    (Intercept) 2359.8   48.58
 Residual              8732.3   93.45
Number of obs: 5935, groups:  nw_target, 199; SubjID, 60

Fixed effects:
                                             Estimate Std. Error t value
(Intercept)                                  687.9439    10.1843  67.549
lang_typeSemantic                             26.4937    13.4632   1.968
nw_famsizeLarge                                1.8841     6.6831   0.282
complexityComplex                              3.9001     6.7021   0.582
lang_typeSemantic:nw_famsizeLarge              7.2719     6.8776   1.057
lang_typeSemantic:complexityComplex          -15.3532     6.8744  -2.233
nw_famsizeLarge:complexityComplex              0.2628     9.4648   0.028
lang_typeSemantic:nw_famsizeLarge:complexityComplex  -5.8078     9.7186  -0.598

Correlation of Fixed Effects:
            (Intr) lng_tS nw_fmL cmplxC ln_S:_L ln_S:C nw_L:C
lng_typSmnt -0.682
nw_famszLrg -0.328  0.134
cmplxtyCmpl -0.327  0.134  0.499
lng_typS:_L  0.173 -0.256 -0.526 -0.262
lng_typSm:C  0.173 -0.256 -0.263 -0.525  0.501
nw_fmszLr:C  0.232 -0.095 -0.706 -0.708  0.371   0.372
lng_tS:_L:C -0.122  0.181  0.372  0.371 -0.708  -0.707 -0.526
```

Model Comparisons

```r
anova(cw_null.model,cw_main.model)
```

```
Data: cw
Models:
cw_null.model: rt.err.imp ~ 1 + (1 | SubjID) + (1 | cw_target)
cw_main.model: rt.err.imp ~ lang_type + cw_famsize + (1 | SubjID) + (1 | cw_target)
              npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
cw_null.model    4 72113 72140 -36052    72105
cw_main.model    6 72105 72145 -36046    72093 11.999  2   0.002479 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(cw_main.model,cw_inter.model)
```

```
Data: cw
Models:
cw_main.model: rt.err.imp ~ lang_type + cw_famsize + (1 | SubjID) + (1 | cw_target)
cw_inter.model: rt.err.imp ~ lang_type * cw_famsize + (1 | SubjID) + (1 | cw_target)
               npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
cw_main.model     6 72105 72145 -36046    72093
cw_inter.model    7 72106 72153 -36046    72092 1.0575  1     0.3038
```

```r
anova(nw_null.model,nw_main.model)
```

```
Data: nw
Models:
nw_null.model: rt.err.imp ~ 1 + (1 | SubjID) + (1 | nw_target)
nw_main.model: rt.err.imp ~ lang_type + nw_famsize + complexity + (1 | SubjID) + (1 | nw_target)
              npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw_null.model    4 71129 71156 -35560    71121
nw_main.model    7 71129 71176 -35558    71115 5.4494  3     0.1417
```

```r
anova(nw_main.model,nw_inter.model)
```

```
Data: nw
Models:
nw_main.model: rt.err.imp ~ lang_type + nw_famsize + complexity + (1 | SubjID) + (1 | nw_target)
nw_inter.model: rt.err.imp ~ lang_type * nw_famsize * complexity + (1 | SubjID) + (1 | nw_target)
               npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw_main.model     7 71129 71176 -35558    71115
nw_inter.model   11 71122 71196 -35550    71100 15.376  4   0.003982 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```