m21_202303_rt v.2

Joanna Morris

2023-03-20

Compute PCA

This script computes separate ANOVAs for simple and complex 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.

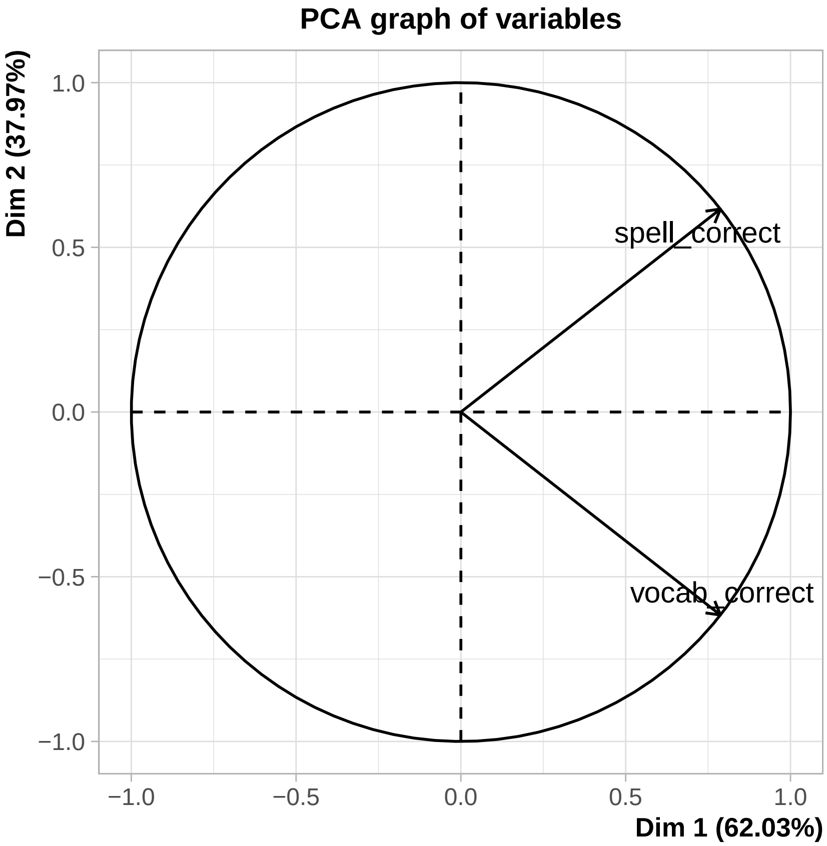
- `X`: a data frame. Rows are individuals and columns are numeric variables
- `scale.unit`: a logical value. If TRUE, the data are scaled to unit variance before the analysis. This standardization to the same scale avoids some variables to become dominant just because of their large measurement units.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

```r
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

```r
(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.

```
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.

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

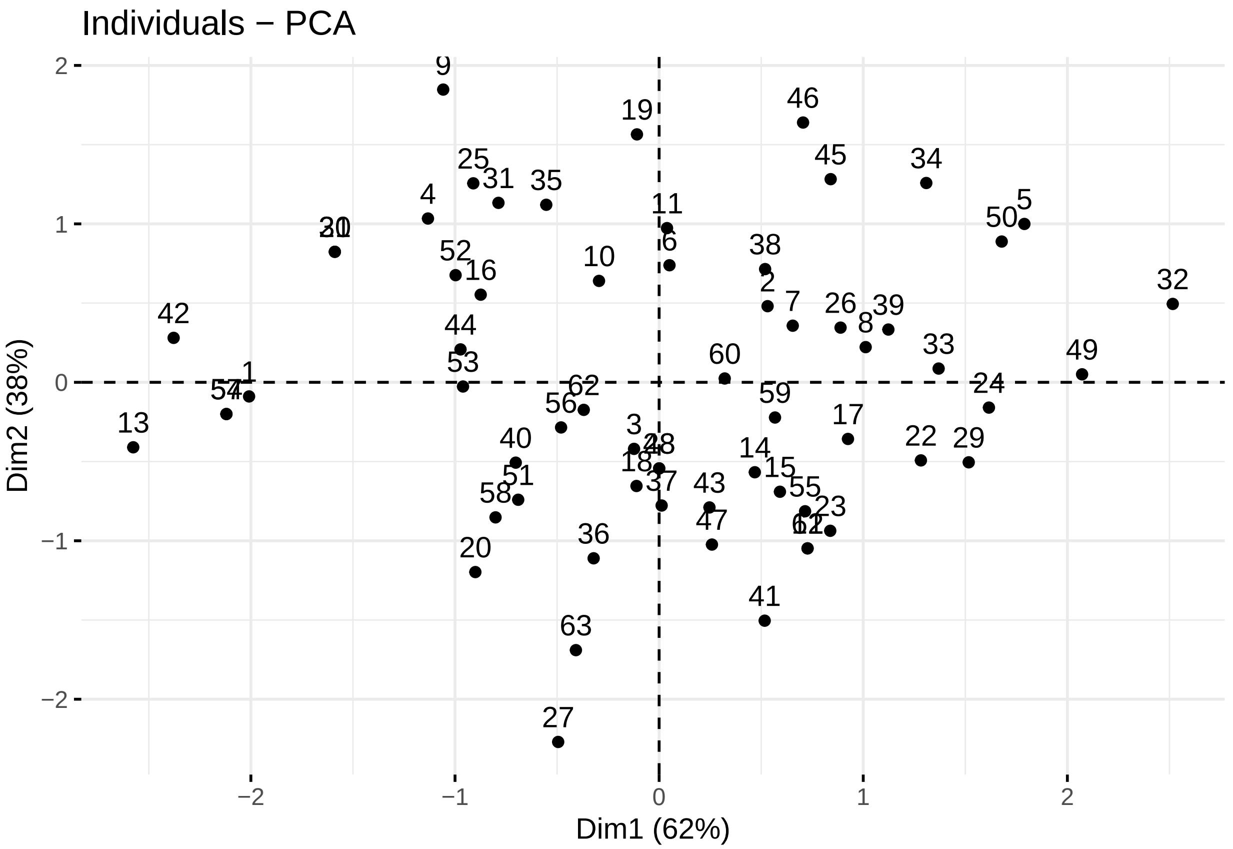
```
              Dim.1 Dim.2
vocab_correct    50    50
spell_correct    50    50
```

```
(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.

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



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

Loads RT data and join to PCA dataset

```r
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"
    ))
```

Export list of subjects with Reading type classification

```
df <- distinct(select(cw, SubjID, lang_type ))
df_o <- filter(df, lang_type == "Orthographic")
df_s <- filter(df, lang_type != "Orthographic")
write_csv(df, "m21_subjlist_reading_type.csv")
write_csv(df_o, "m21_subjlist_ortho.csv")
write_csv(df_s, "m21_subjlist_semant.csv")
```

```
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")


nw_smpl <- filter(nw, complexity == "Simple")
nw_smpl$complexity <- NULL

nw_cplx <- filter(nw, complexity == "Complex")
nw_cplx$complexity <- NULL

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

nw_cplx <- nw_cplx |> mutate(rt.err = response_time * correct)  # convert error rts to 0
nw_cplx <- nw_cplx |> mutate(rt.err = na_if(rt.err, 0))         # convert 0 rts to NA
nw_cplx.mean <- mean(nw_cplx$rt.err, na.rm = TRUE)  # get mean rt excluding errors
nw_cplx <- nw_cplx |> mutate(rt.err.imp = ifelse(is.na(rt.err),
                                                 nw_smpl.mean,
                                                 rt.err))  # replace missing values with mean
nw_cplx <- nw_cplx |> mutate(inv.rt = 1/rt.err.imp)  # creates new column with inverse 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))
(xtab.missing.data <- xtabs(~cw_famsize+lang_type, data=cw_missing_data))
```

```
          lang_type
cw_famsize Orthographic Semantic
     Small          213      169
     Large          225      163
```

```
nw.smpl_missing_data<- filter(nw_smpl, is.na(nw_smpl$rt.err))
(xtab.missing.data <- xtabs(~nw_famsize+lang_type, data=nw.smpl_missing_data))
```

```
          lang_type
nw_famsize Orthographic Semantic
     Small          207      212
     Large          215      257
```

```
nw.cplx_missing_data<- filter(nw_cplx, is.na(nw_cplx$rt.err))
(xtab.missing.data <- xtabs(~nw_famsize+lang_type, data=nw.cplx_missing_data))
```

```
          lang_type
nw_famsize Orthographic Semantic
     Small          223      229
     Large          245      252
```

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: 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  6.8551395 1.125713e-02     * 0.1024825208
3          cw_famsize   1  58 29.1533842 1.299937e-06     * 0.0167579493
4 lang_type:cw_famsize   1  58  0.5478038 4.622020e-01       0.0003201529
```

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

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

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 7.197773 0.009494787     * 0.1018639868
3          nw_famsize   1  58 2.105327 0.152174207       0.0031148459
4 lang_type:nw_famsize   1  58 0.171281 0.680502319       0.0002541385
```

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

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

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 2.14786827 0.1481685       0.0337835603
3          nw_famsize   1  58 0.57142520 0.4527524       0.0005497191
4 lang_type:nw_famsize   1  58 0.07260026 0.7885440       0.0000698760
```

```
m1 <- aov(rt.err.imp ~ lang_type * cw_famsize, data = cw)
m2 <- aov(rt.err.imp ~ lang_type * nw_famsize, data = nw_smpl)
m3 <- aov(rt.err.imp ~ lang_type * nw_famsize, data = nw_cplx)
summary(m1)
```

```
                       Df   Sum Sq Mean Sq F value   Pr(>F)    
lang_type               1  2502145 2502145 169.151  < 2e-16 ***
cw_famsize              1   382343  382343  25.847 3.81e-07 ***
lang_type:cw_famsize    1     7375    7375   0.499     0.48    
Residuals            5917 87526294   14792                     
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(m2)
```

```
                       Df   Sum Sq Mean Sq F value Pr(>F)    
lang_type               1   927772  927772  81.241 <2e-16 ***
nw_famsize              1    21204   21204   1.857  0.173    
lang_type:nw_famsize    1     1128    1128   0.099  0.753    
Residuals            2958 33780406   11420                   
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(m3)
```

```
                       Df   Sum Sq Mean Sq F value   Pr(>F)    
lang_type               1   326190  326190  27.975 1.32e-07 ***
nw_famsize              1     6693    6693   0.574    0.449    
lang_type:nw_famsize    1      218     218   0.019    0.891    
Residuals            2969 34618790   11660                     
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Get condition means

```
#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  591.  3.20     1500
2 Small      Semantic      635.  3.19     1456
3 Large      Orthographic  578.  3.21     1500
4 Large      Semantic      616.  3.04     1465
```

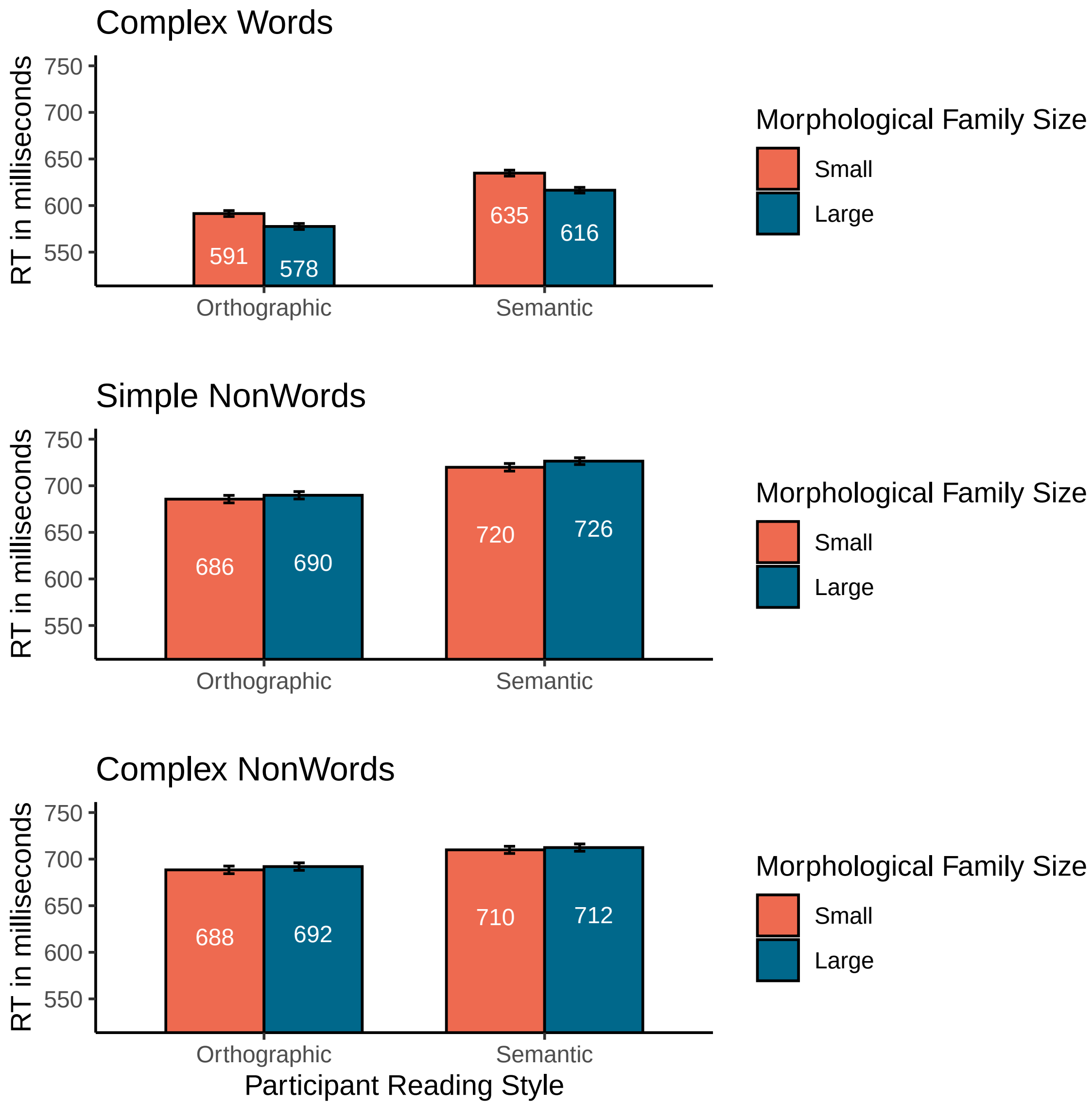
```r
(nw_smpl.cond.means <- nw_smpl |>
    group_by(nw_famsize, lang_type) |>
    summarise(mean = mean(rt.err.imp),
              se = sem(rt.err.imp),
              num_stim = n()))
```

```
# A tibble: 4 x 5
# Groups:   nw_famsize [2]
  nw_famsize lang_type     mean    se num_stim
  <fct>      <fct>        <dbl> <dbl>    <int>
1 Small      Orthographic  686.  4.02      750
2 Small      Semantic      720.  4.06      730
3 Large      Orthographic  690.  3.95      750
4 Large      Semantic      726.  3.66      732
```

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

```
# A tibble: 4 x 5
# Groups:   nw_famsize [2]
  nw_famsize lang_type     mean    se num_stim
  <fct>      <fct>        <dbl> <dbl>    <int>
1 Small      Orthographic  688.  4.09      750
2 Small      Semantic      710.  3.83      736
3 Large      Orthographic  692.  4.01      750
4 Large      Semantic      712.  3.90      737
```

Barplots



LME

Models

```
library(lme4)

# COMPLEX WORDS

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
```

```
# 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
 72101.3  72141.4 -36044.6  72089.3     5915

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2881 -0.6775 -0.1487  0.4741  4.9462

Random effects:
 Groups    Name        Variance Std.Dev.
 cw_target (Intercept)   511.3   22.61
 SubjID    (Intercept)  3552.2   59.60
 Residual              10713.5  103.51
Number of obs: 5921, groups:  cw_target, 100; SubjID, 60

Fixed effects:
                  Estimate Std. Error t value
(Intercept)        592.563     11.576  51.187
lang_typeSemantic   41.510     15.625   2.657
cw_famsizeLarge    -16.131      5.262  -3.065

Correlation of Fixed Effects:
            (Intr) lng_tS
lng_typSmnt -0.674
cw_famszLrg -0.227  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
 72102.6  72149.4 -36044.3  72088.6     5914

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2994 -0.6792 -0.1480  0.4721  4.9576

Random effects:
 Groups    Name        Variance Std.Dev.
 cw_target (Intercept)   511.3   22.61
 SubjID    (Intercept)  3552.3   59.60
 Residual              10712.2  103.50
Number of obs: 5921, groups:  cw_target, 100; SubjID, 60

Fixed effects:
                                  Estimate Std. Error t value
(Intercept)                        591.432     11.652  50.757
lang_typeSemantic                   43.810     15.857   2.763
cw_famsizeLarge                    -13.869      5.894  -2.353
lang_typeSemantic:cw_famsizeLarge   -4.585      5.381  -0.852

Correlation of Fixed Effects:
            (Intr) lng_tS cw_fmL
lng_typSmnt -0.680
cw_famszLrg -0.253  0.076
lng_typS:_L  0.114 -0.170 -0.450
```

```
# SIMPLE NONWORDS

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

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

     AIC      BIC   logLik deviance df.resid
 35537.9  35561.8 -17764.9  35529.9     2958

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3363 -0.6270 -0.0966  0.5739  4.0367
```

```
Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  474.6   21.78
 SubjID    (Intercept) 2575.7   50.75
 Residual              8689.4   93.22
Number of obs: 2962, groups:  nw_target, 100; SubjID, 60

Fixed effects:
            Estimate Std. Error t value
(Intercept)  705.123      7.117   99.08
```

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

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

     AIC      BIC   logLik deviance df.resid
 35533.8  35569.8 -17760.9  35521.8     2956

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3147 -0.6303 -0.0928  0.5752  4.0300

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  465.9   21.59
 SubjID    (Intercept) 2271.4   47.66
 Residual              8689.5   93.22
Number of obs: 2962, groups:  nw_target, 100; SubjID, 60

Fixed effects:
                  Estimate Std. Error t value
(Intercept)        684.885      9.684  70.722
lang_typeSemantic   34.810     12.779   2.724
nw_famsizeLarge      5.726      5.513   1.039

Correlation of Fixed Effects:
            (Intr) lng_tS
lng_typSmnt -0.659
nw_famszLrg -0.285  0.000
```

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

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

```
   Data: nw_smpl

     AIC      BIC   logLik deviance df.resid
 35535.6  35577.5 -17760.8  35521.6     2955

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3235 -0.6306 -0.0923  0.5744  4.0386

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  466     21.59
 SubjID    (Intercept) 2272     47.66
 Residual              8689     93.21
Number of obs: 2962, groups:  nw_target, 100; SubjID, 60

Fixed effects:
                                  Estimate Std. Error t value
(Intercept)                        685.682      9.832  69.742
lang_typeSemantic                   33.191     13.234   2.508
nw_famsizeLarge                      4.133      6.469   0.639
lang_typeSemantic:nw_famsizeLarge    3.229      6.855   0.471

Correlation of Fixed Effects:
            (Intr) lng_tS nw_fmL
lng_typSmnt -0.671
nw_famszLrg -0.329  0.135
lng_typS:_L  0.172 -0.260 -0.523
```

```
# COMPLEX NONWORDS

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

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

     AIC      BIC   logLik deviance df.resid
 35610.7  35634.7 -17801.3  35602.7     2969

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2418 -0.6006 -0.1426  0.5543  3.4087

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  379.9   19.49
 SubjID    (Intercept) 2810.5   53.01
 Residual              8537.8   92.40
Number of obs: 2973, groups:  nw_target, 99; SubjID, 60

Fixed effects:
```

```
            Estimate Std. Error t value
(Intercept)   700.41       7.32   95.69
```

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

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

     AIC      BIC   logLik deviance df.resid
 35612.1  35648.1 -17800.1  35600.1     2967

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2278 -0.6036 -0.1481  0.5568  3.4204

Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  378.3   19.45
 SubjID    (Intercept) 2699.1   51.95
 Residual              8537.7   92.40
Number of obs: 2973, groups:  nw_target, 99; SubjID, 60

Fixed effects:
                  Estimate Std. Error t value
(Intercept)        688.461     10.311  66.770
lang_typeSemantic   20.876     13.838   1.509
nw_famsizeLarge      3.004      5.180   0.580

Correlation of Fixed Effects:
            (Intr) lng_tS
lng_typSmnt -0.671
nw_famszLrg -0.253  0.000
```

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

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

     AIC      BIC   logLik deviance df.resid
 35614.1  35656.1 -17800.0  35600.1     2966

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2241 -0.6031 -0.1465  0.5543  3.4240
```

```
Random effects:
 Groups    Name        Variance Std.Dev.
 nw_target (Intercept)  378.2   19.45
 SubjID    (Intercept) 2699.2   51.95
 Residual              8537.6   92.40
Number of obs: 2973, groups:  nw_target, 99; SubjID, 60

Fixed effects:
                                  Estimate Std. Error t value
(Intercept)                        688.124     10.447  65.868
lang_typeSemantic                   21.556     14.247   1.513
nw_famsizeLarge                      3.678      6.174   0.596
lang_typeSemantic:nw_famsizeLarge   -1.360      6.781  -0.201

Correlation of Fixed Effects:
            (Intr) lng_tS nw_fmL
lng_typSmnt -0.681
nw_famszLrg -0.297  0.129
lng_typS:_L  0.161 -0.238 -0.544
```

Model Comparisons

```
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 72101 72141 -36045    72089 15.647  2  0.0004002 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
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 72101 72141 -36045    72089
cw_inter.model    7 72103 72149 -36044    72089 0.726  1     0.3942
```

```
anova(nw.smpl_null.model,nw.smpl_main.model)
```

```
Data: nw_smpl
Models:
nw.smpl_null.model: rt.err.imp ~ 1 + (1 | SubjID) + (1 | nw_target)
nw.smpl_main.model: rt.err.imp ~ lang_type + nw_famsize + (1 | SubjID) + (1 | nw_target)
```

```
                   npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.smpl_null.model    4 35538 35562 -17765    35530
nw.smpl_main.model    6 35534 35570 -17761    35522 8.0675  2    0.01771 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
anova(nw.smpl_main.model,nw.smpl_inter.model)
```

```

Data: nw_smpl
Models:
nw.smpl_main.model: rt.err.imp ~ lang_type + nw_famsize + (1 | SubjID) + (1 | nw_target)
nw.smpl_inter.model: rt.err.imp ~ lang_type * nw_famsize + (1 | SubjID) + (1 | nw_target)
                    npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.smpl_main.model     6 35534 35570 -17761    35522
nw.smpl_inter.model    7 35536 35578 -17761    35522 0.2219  1     0.6376
```

```
anova(nw.cplx_null.model,nw.cplx_main.model)
```

```

Data: nw_cplx
Models:
nw.cplx_null.model: rt.err.imp ~ 1 + (1 | SubjID) + (1 | nw_target)
nw.cplx_main.model: rt.err.imp ~ lang_type + nw_famsize + (1 | SubjID) + (1 | nw_target)
                   npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.cplx_null.model    4 35611 35635 -17801    35603
nw.cplx_main.model    6 35612 35648 -17800    35600 2.5685  2     0.2769
```

```
anova(nw.cplx_main.model,nw.cplx_inter.model)
```

```

Data: nw_cplx
Models:
nw.cplx_main.model: rt.err.imp ~ lang_type + nw_famsize + (1 | SubjID) + (1 | nw_target)
nw.cplx_inter.model: rt.err.imp ~ lang_type * nw_famsize + (1 | SubjID) + (1 | nw_target)
                    npar   AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.cplx_main.model     6 35612 35648 -17800    35600
nw.cplx_inter.model    7 35614 35656 -17800    35600 0.0402  1      0.841
```