Morph 21

Joanna Morris

2023-03-18

This R script contains the code for analysing the morph 21 erp data for the 200-300 ms time window.

1. First we load the libraries we need

```
library(readr)
library(psych)
library(dplyr)
library(tidyr)
```

Compute PCA

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_corre

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
```

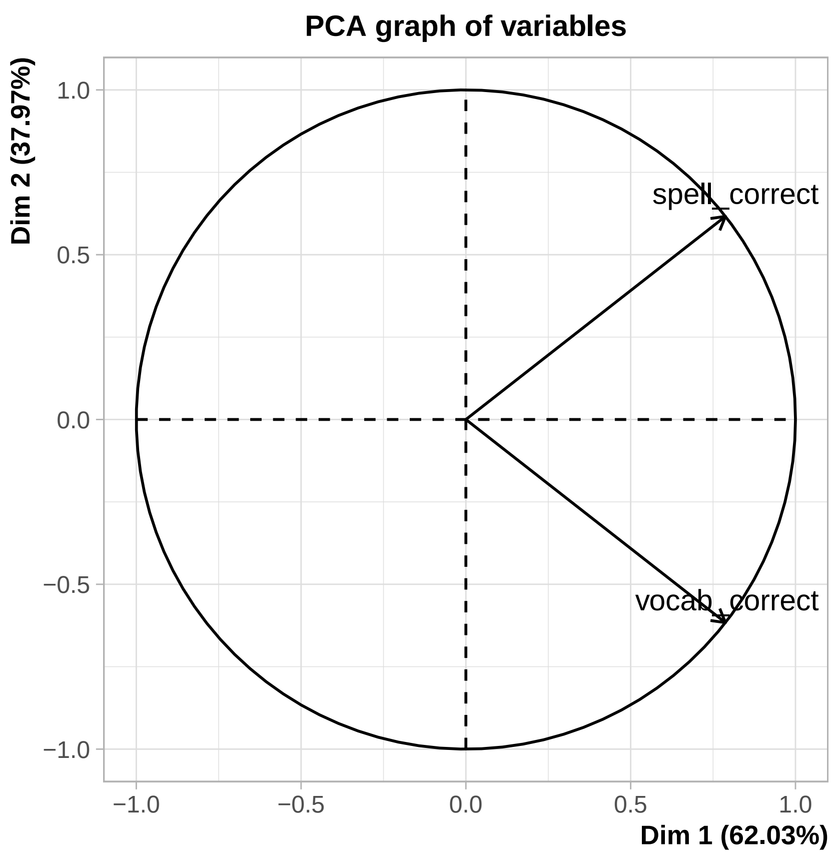
By default, the function `PCA()` in `FactoMineR`, standardizes the data automatically during the PCA; so you don't need do this transformation before the PCA.

- `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

```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.

```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
```

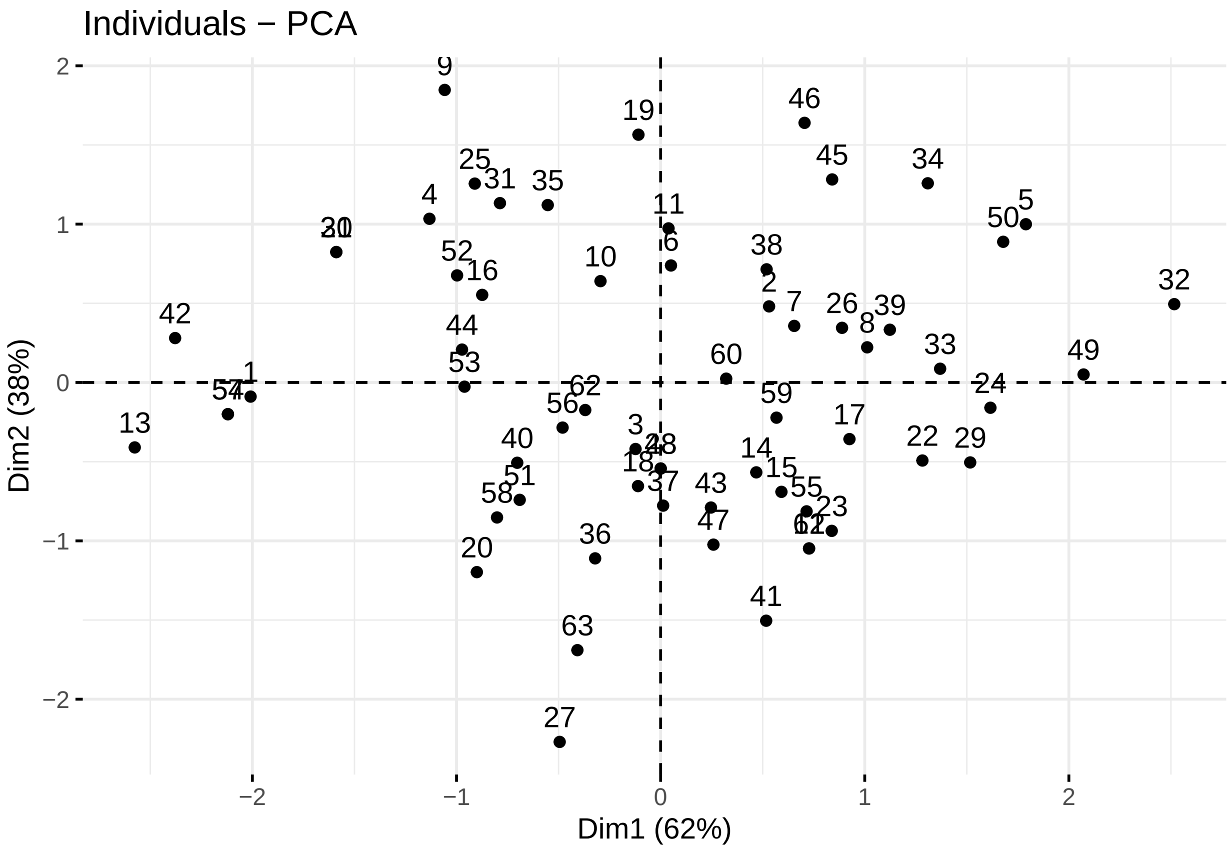
```
              Dim.1 Dim.2
vocab_correct    50    50
spell_correct    50    50
```

```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)
```

We load the N250 erp data file and the word and non-word base frequency data

```
n250 <- read_csv("S101-177_n250.csv")
```

Then we join the demographic and erp data files. We will use the `inner_join` rather than the `full_join` function in order to eliminate rows with missing data.

```
n250 <- inner_join(sv_202303.na,n250, by = "SubjID")  #join subject PCA data
```

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

```
n250.median <- median(n250$Dim.2)
n250 <- n250 |>
    mutate(lang_type = case_when(
      Dim.2 <= n250.median ~ "Semantic",
      Dim.2 > n250.median ~ "Orthographic"
    ))
```

5. Let's save a `.csv` file with the data from the combined dataset

```
write_csv(n250, "202303_sv_n250_rmna.csv")
```

6. For each dataset, we will create a subset with only the electrode sites we will be analysing—F3, Fz, F4, C3, Cz, C4, P3, Pz, P4

```
sites = c(3,2, 25, 7, 20, 21, 12, 11, 16)
n250_9 <- dplyr::filter(n250, chindex %in% sites)
```

7. We then create separate columns, one for each independent variable (anteriority, laterality, morphological family size). To do this we have to use the`mutate` function from the dplyr package along with the `case_when` function. The `case_when` function is a sequence of two-sided formulas. The left hand side determines which values match this case. The right hand side provides the replacement value.

```
n250_9 <- dplyr::mutate(n250_9,
                        anteriority = case_when(grepl("F", chlabel) ~ "Frontal",
                                                grepl("C", chlabel) ~ "Central",
                                                grepl("P", chlabel) ~ "Parietal"))

n250_9 <- dplyr::mutate(n250_9,
                        laterality = case_when(grepl("3", chlabel) ~ "Left",
                                               grepl("z", chlabel) ~ "Midline",
                                               grepl("Z", chlabel) ~ "Midline",
                                               grepl("4", chlabel) ~ "Right"))



n250_9 <- dplyr::mutate(n250_9,
                        fam_size = case_when(grepl("small", binlabel) ~ "Small",
                                             grepl("large", binlabel) ~ "Large"))
```

8. We then create a smaller dataset with only the columns we need

```
n250_9b <- dplyr::select(n250_9,
                             SubjID,
                             lang_type,
                             anteriority,
                             laterality,
                             fam_size,
                             value,
                             chlabel,
                             binlabel)
```

9. We then divide dataset into 3 separate ones—for "words", "simple nonwords" and "complex nonwords"

```
n250_words <- dplyr::filter(n250_9b, grepl("Critical_word",binlabel))
n250_nwsmpl <- dplyr::filter(n250_9b, grepl("simple",binlabel))
n250_nwcplx <- dplyr::filter(n250_9b, grepl("complex",binlabel))
```

10. Now we can compute the ANOVA for each of the three datasets.

We use the `aov` function to calculate the source table and F. But before we do that, we want to tell R to change the way it is comparing the groups. We are going to use `Helmert` contrast, and we only need to set it once for all ANOVAs we run during a single R session.

```
options(contrasts = c("contr.helmert", "contr.poly"))
options("contrasts")
```

```
$contrasts
[1] "contr.helmert" "contr.poly"
```

After we tell R we want to use a Helmert contrast (`contr. helmert`) for categorical variables (and polynomial contrasts for ordered variables (`contr.poly`)), we can now run our ANOVA. For more on contrasts see this explanation

```
m1 <- aov(value ~ lang_type * fam_size * anteriority * laterality, data = n250_words)
m2 <- aov(value ~ lang_type * fam_size * anteriority * laterality, data = n250_nwsmpl)
m3 <- aov(value ~ lang_type * fam_size * anteriority * laterality, data = n250_nwcplx)
summary(m1)
```

```
                                           Df Sum Sq Mean Sq F value   Pr(>F)
lang_type                                   1    224  224.16  21.149 4.81e-06
fam_size                                    1     14   13.56   1.279   0.2584
anteriority                                 2     88   43.76   4.129   0.0164
laterality                                  2      8    4.04   0.381   0.6830
lang_type:fam_size                          1      1    1.23   0.116   0.7335
lang_type:anteriority                       2      4    1.83   0.173   0.8413
fam_size:anteriority                        2      5    2.60   0.245   0.7826
lang_type:laterality                        2      0    0.06   0.006   0.9945
fam_size:laterality                         2      0    0.03   0.003   0.9974
anteriority:laterality                      4      5    1.23   0.116   0.9769
lang_type:fam_size:anteriority              2      6    2.85   0.269   0.7643
lang_type:fam_size:laterality               2      4    2.23   0.210   0.8105
lang_type:anteriority:laterality            4      5    1.29   0.122   0.9747
fam_size:anteriority:laterality             4      1    0.22   0.021   0.9992
lang_type:fam_size:anteriority:laterality   4      1    0.37   0.035   0.9976
Residuals                                 972  10303   10.60

lang_type                                 ***
fam_size
anteriority                               *
laterality
lang_type:fam_size
lang_type:anteriority
fam_size:anteriority
lang_type:laterality
fam_size:laterality
anteriority:laterality
lang_type:fam_size:anteriority
lang_type:fam_size:laterality
lang_type:anteriority:laterality
fam_size:anteriority:laterality
lang_type:fam_size:anteriority:laterality
Residuals
```

```
---
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    598   598.0  37.453 1.36e-09
fam_size                                  1     84    84.3   5.276   0.0218
anteriority                               2    108    54.0   3.379   0.0345
laterality                                2      6     3.0   0.186   0.8299
lang_type:fam_size                        1     19    19.0   1.192   0.2751
lang_type:anteriority                     2     23    11.4   0.717   0.4885
fam_size:anteriority                      2      3     1.6   0.099   0.9062
lang_type:laterality                      2      0     0.0   0.003   0.9973
fam_size:laterality                       2      4     2.2   0.135   0.8740
anteriority:laterality                    4      8     1.9   0.119   0.9756
lang_type:fam_size:anteriority            2      0     0.1   0.009   0.9915
lang_type:fam_size:laterality             2      4     2.0   0.128   0.8798
lang_type:anteriority:laterality          4      3     0.7   0.043   0.9964
fam_size:anteriority:laterality           4      1     0.3   0.021   0.9992
lang_type:fam_size:anteriority:laterality 4      8     2.1   0.130   0.9713
Residuals                               972  15521    16.0

lang_type                               ***
fam_size                                *
anteriority                             *
laterality
lang_type:fam_size
lang_type:anteriority
fam_size:anteriority
lang_type:laterality
fam_size:laterality
anteriority:laterality
lang_type:fam_size:anteriority
lang_type:fam_size:laterality
lang_type:anteriority:laterality
fam_size:anteriority:laterality
lang_type:fam_size:anteriority:laterality
Residuals
---
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    487   486.8  41.272 2.07e-10
fam_size                                  1    196   196.1  16.623 4.93e-05
anteriority                               2    141    70.5   5.975  0.00264
laterality                                2     10     5.1   0.428  0.65173
lang_type:fam_size                        1     47    46.8   3.965  0.04675
lang_type:anteriority                     2     27    13.3   1.126  0.32467
fam_size:anteriority                      2     26    12.9   1.096  0.33456
lang_type:laterality                      2      3     1.3   0.109  0.89669
```

```
fam_size:laterality                          2      3     1.3   0.112  0.89422
anteriority:laterality                       4     13     3.2   0.272  0.89617
lang_type:fam_size:anteriority               2     13     6.4   0.540  0.58318
lang_type:fam_size:laterality                2      1     0.7   0.058  0.94347
lang_type:anteriority:laterality             4     10     2.5   0.211  0.93214
fam_size:anteriority:laterality              4      6     1.4   0.122  0.97470
lang_type:fam_size:anteriority:laterality    4      3     0.7   0.063  0.99269
Residuals                                  972  11464    11.8

lang_type                                  ***
fam_size                                   ***
anteriority                                **
laterality
lang_type:fam_size                         *
lang_type:anteriority
fam_size:anteriority
lang_type:laterality
fam_size:laterality
anteriority:laterality
lang_type:fam_size:anteriority
lang_type:fam_size:laterality
lang_type:anteriority:laterality
fam_size:anteriority:laterality
lang_type:fam_size:anteriority:laterality
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also use the function `Anova()` from the `car` package with the argument `type = III` to get the source table. For our design with no factors with more than 2 levels, both methods should give similar results. For more on types of sums of square see this explanation

```r
library(car)
Anova(m1, type = "III")
```

```
Anova Table (Type III tests)

Response: value
                                          Sum Sq  Df F value    Pr(>F)
(Intercept)                                 12.9   1  1.2137   0.27086
lang_type                                  224.2   1 21.1486 4.811e-06 ***
fam_size                                    13.6   1  1.2789   0.25838
anteriority                                 87.5   2  4.1288   0.01638 *
laterality                                   8.1   2  0.3814   0.68301
lang_type:fam_size                           1.2   1  0.1160   0.73353
lang_type:anteriority                        3.7   2  0.1728   0.84129
fam_size:anteriority                         5.2   2  0.2452   0.78262
lang_type:laterality                         0.1   2  0.0055   0.99450
fam_size:laterality                          0.1   2  0.0026   0.99736
anteriority:laterality                       4.9   4  0.1159   0.97693
lang_type:fam_size:anteriority               5.7   2  0.2688   0.76432
lang_type:fam_size:laterality                4.5   2  0.2101   0.81053
lang_type:anteriority:laterality             5.2   4  0.1218   0.97470
```

```
fam_size:anteriority:laterality               0.9   4  0.0208   0.99915
lang_type:fam_size:anteriority:laterality     1.5   4  0.0353   0.99762
Residuals                                 10302.7 972
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(m2, type = "III")
```

```
Anova Table (Type III tests)

Response: value
                                           Sum Sq  Df F value    Pr(>F)
(Intercept)                                 165.5   1 10.3630  0.001328 **
lang_type                                   598.0   1 37.4531 1.356e-09 ***
fam_size                                     84.3   1  5.2764  0.021828 *
anteriority                                 107.9   2  3.3792  0.034474 *
laterality                                    6.0   2  0.1865  0.829911
lang_type:fam_size                           19.0   1  1.1923  0.275140
lang_type:anteriority                        22.9   2  0.7169  0.488518
fam_size:anteriority                          3.1   2  0.0985  0.906164
lang_type:laterality                          0.1   2  0.0027  0.997337
fam_size:laterality                           4.3   2  0.1347  0.874001
anteriority:laterality                        7.6   4  0.1194  0.975606
lang_type:fam_size:anteriority                0.3   2  0.0086  0.991466
lang_type:fam_size:laterality                 4.1   2  0.1281  0.879753
lang_type:anteriority:laterality              2.8   4  0.0433  0.996450
fam_size:anteriority:laterality               1.3   4  0.0207  0.999166
lang_type:fam_size:anteriority:laterality     8.3   4  0.1304  0.971306
Residuals                                 15520.6 972
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(m3, type = "III")
```

```
Anova Table (Type III tests)

Response: value
                                           Sum Sq  Df F value    Pr(>F)
(Intercept)                                  30.1   1  2.5561  0.110191
lang_type                                   486.8   1 41.2715 2.071e-10 ***
fam_size                                    196.1   1 16.6233 4.931e-05 ***
anteriority                                 140.9   2  5.9748  0.002636 **
laterality                                   10.1   2  0.4283  0.651734
lang_type:fam_size                           46.8   1  3.9645  0.046748 *
lang_type:anteriority                        26.6   2  1.1263  0.324669
fam_size:anteriority                         25.9   2  1.0962  0.334561
lang_type:laterality                          2.6   2  0.1091  0.896694
fam_size:laterality                           2.6   2  0.1118  0.894220
anteriority:laterality                       12.8   4  0.2719  0.896173
lang_type:fam_size:anteriority               12.7   2  0.5396  0.583183
lang_type:fam_size:laterality                 1.4   2  0.0582  0.943474
lang_type:anteriority:laterality             10.0   4  0.2114  0.932136
fam_size:anteriority:laterality               5.7   4  0.1218  0.974695
```

```
lang_type:fam_size:anteriority:laterality     3.0   4  0.0630  0.992695
Residuals                                 11463.8 972
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#Plot Means

Get condition means

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

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

(cw.cond.means <- n250_words |>
   group_by(fam_size, lang_type) |>
   summarise(mean = mean(value),
             se = sem(value),
             num_stim = n()))
```

```
# A tibble: 4 x 5
# Groups:   fam_size [2]
  fam_size lang_type      mean    se num_stim
  <chr>    <chr>         <dbl> <dbl>    <int>
1 Large    Orthographic  0.208 0.224      252
2 Large    Semantic     -0.666 0.167      252
3 Small    Orthographic  0.509 0.216      252
4 Small    Semantic     -0.504 0.201      252
```

```r
(nw_smp.cond.means <- n250_nwsmpl |>
    group_by(fam_size, lang_type) |>
    summarise(mean = mean(value),
              se = sem(value),
              num_stim = n()))
```

```
# A tibble: 4 x 5
# Groups:   fam_size [2]
  fam_size lang_type       mean    se num_stim
  <chr>    <chr>          <dbl> <dbl>    <int>
1 Large    Orthographic -0.0614 0.254      252
2 Large    Semantic     -1.33   0.207      252
3 Small    Orthographic  0.792  0.279      252
4 Small    Semantic     -1.02   0.250      252
```

```r
(nw_cpx.cond.means <- n250_nwcplx |>
    group_by(fam_size, lang_type) |>
    summarise(mean = mean(value),
              se = sem(value),
              num_stim = n()))
```

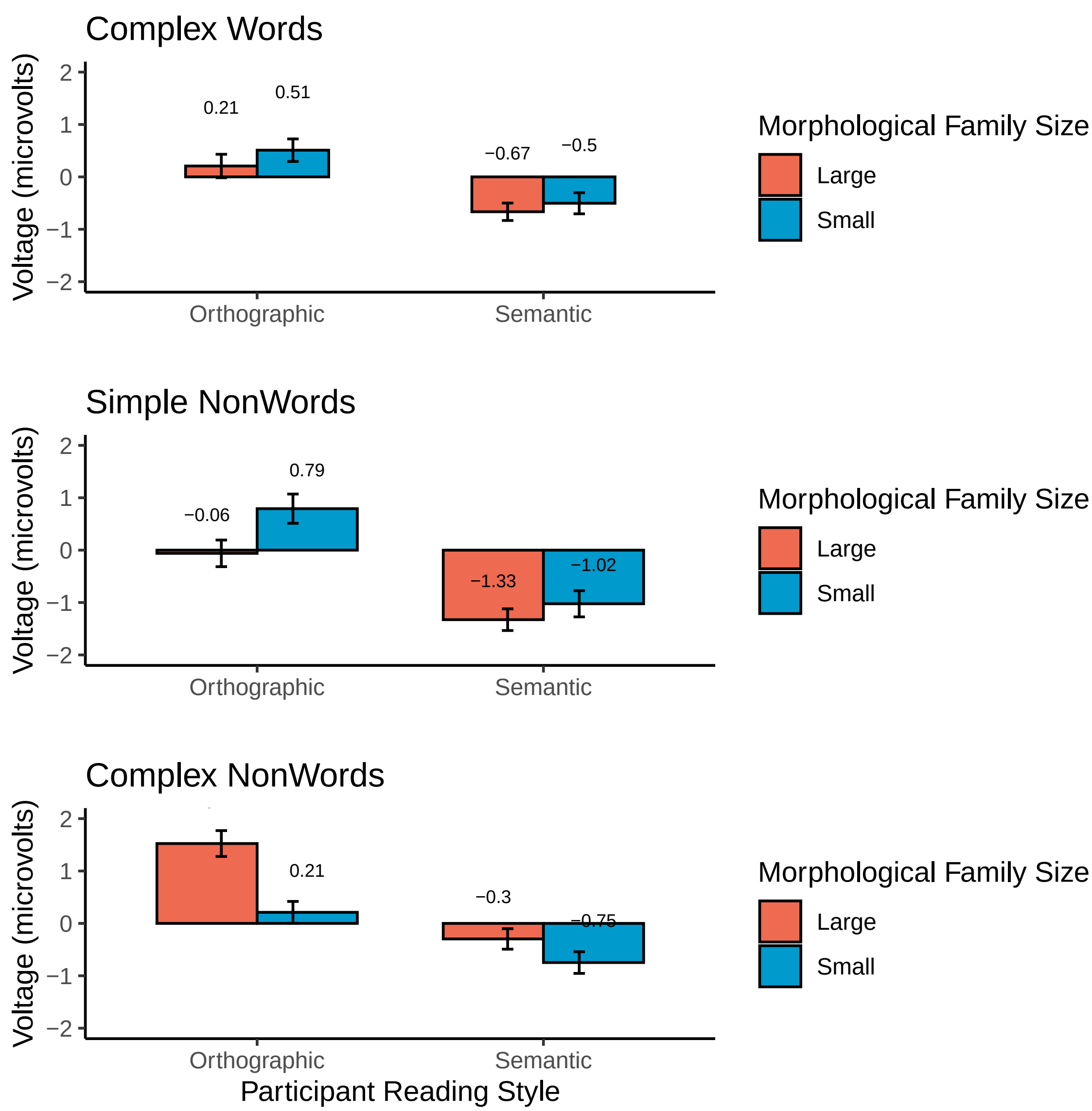
```
# A tibble: 4 x 5
# Groups:   fam_size [2]
  fam_size lang_type      mean    se num_stim
```

```
  <chr>    <chr>          <dbl> <dbl>    <int>
1 Large    Orthographic  1.52  0.247      252
2 Large    Semantic     -0.296 0.196      252
3 Small    Orthographic  0.211 0.208      252
4 Small    Semantic     -0.748 0.206      252
```

Barplots

```
library(gridExtra)
p1 <-  cw.cond.means |> ggplot(aes(x=lang_type,
                                   y=mean,
                                   fill = fam_size,
                                   ymin = mean - se,
                                   ymax = mean + se)) +
  coord_cartesian(xlim = NULL,
                  ylim = c(-2, 2),
                  expand = TRUE,
                  default = FALSE,
                  clip = "on") +
  geom_col(position = "dodge", width = 0.5, color = "black")  +
  ylab("Voltage (microvolts)")  +
  xlab("")  +
  ggtitle("Complex Words") +
  scale_fill_manual(values = c("coral2", "deepskyblue3"))+
  geom_errorbar(width = .08, position = position_dodge(0.5)) +
  theme_classic() +
   geom_text(aes(label = round(mean, digits = 2)),
             colour = "black",
             size = 2.5,
             vjust = -4,
             position = position_dodge(.5))+
  guides(fill=guide_legend(title="Morphological Family Size"))

p2 <-  nw_smp.cond.means |> ggplot(aes(x=lang_type,
                                       y=mean, fill = fam_size,
                                       ymin = mean - se,
                                       ymax = mean + se)) +
  coord_cartesian(xlim = NULL,
                  ylim = c(-2, 2),
                  expand = TRUE,
                  default = FALSE,
                  clip = "on") +
  geom_col(position = "dodge", width = .7, color = "black")  +
  xlab("")  +
  ylab("Voltage (microvolts)")  +
  ggtitle("Simple NonWords") +
  scale_fill_manual(values = c("coral2", "deepskyblue3"))+
  geom_errorbar(width = .08, position = position_dodge(0.5)) +
  theme_classic() +
   geom_text(aes(label = round(mean, digits = 2)),
             colour = "black",
             size = 2.5,
             vjust = -2.5,
             position = position_dodge(.7)) +
```

```r
  guides(fill=guide_legend(title="Morphological Family Size"))

p3 <-  nw_cpx.cond.means |> ggplot(aes(x=lang_type,
                                      y=mean,
                                      fill = fam_size,
                                      ymin = mean - se,
                                      ymax = mean + se)) +
  coord_cartesian(xlim = NULL,
                  ylim = c(-2, 2),
                  expand = TRUE,
                  default = FALSE,
                  clip = "on") +
  geom_col(position = "dodge", width = .7, color = "black")  +
  xlab("Participant Reading Style")  +
  ylab("Voltage (microvolts)")  +
  ggtitle("Complex NonWords") +
  scale_fill_manual(values = c("coral2", "deepskyblue3"))+
  geom_errorbar(width = .08, position = position_dodge(0.5)) +
  theme_classic() +
   geom_text(aes(label = round(mean, digits = 2)),
             colour = "black",
             size = 2.5,
             vjust = -2.75,
             position = position_dodge(.7)) +
  guides(fill=guide_legend(title="Morphological Family Size"))

grid.arrange(p1, p2, p3)
```



LME

```
library(lme4)

# COMPLEX WORDS

cw_null.model = lmer(value ~ 1 + (1|SubjID) ,
                     data= n250_words, REML=FALSE)
summary(cw_null.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ 1 + (1 | SubjID)
   Data: n250_words
```

```
     AIC      BIC   logLik deviance df.resid
  4555.6   4570.4  -2274.8   4549.6     1005

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.2085 -0.6434 -0.0517  0.5876  3.5597

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.713    2.390
 Residual             4.490    2.119
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
            Estimate Std. Error t value
(Intercept)  -0.2065     0.3292  -0.627
```

```
# Main effects models with random intercepts
cw_main.model = lmer(value ~ lang_type + fam_size + (1|SubjID) ,
                     data= n250_words, REML=FALSE)
summary(cw_main.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type + fam_size + (1 | SubjID)
   Data: n250_words

     AIC      BIC   logLik deviance df.resid
  4555.2   4579.8  -2272.6   4545.2     1003

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1700 -0.6478 -0.0455  0.6075  3.5007

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.565    2.359
 Residual             4.476    2.116
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
            Estimate Std. Error t value
(Intercept) -0.19960    0.32510  -0.614
lang_type1  -0.38495    0.32510  -1.184
fam_size1    0.11597    0.06663   1.740

Correlation of Fixed Effects:
           (Intr) lng_t1
lang_type1 -0.018
fam_size1   0.000  0.000
```

```
# Interaction effects models with random intercepts
cw_inter.model = lmer(value ~ lang_type * fam_size + (1|SubjID) ,
                      data= n250_words, REML=FALSE)
summary(cw_inter.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type * fam_size + (1 | SubjID)
   Data: n250_words

     AIC      BIC   logLik deviance df.resid
  4557.0   4586.5  -2272.5   4545.0     1002

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1541 -0.6385 -0.0437  0.6065  3.4848

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.565    2.359
 Residual             4.474    2.115
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
                    Estimate Std. Error t value
(Intercept)         -0.19960    0.32510  -0.614
lang_type1          -0.38495    0.32510  -1.184
fam_size1            0.11597    0.06662   1.741
lang_type1:fam_size1 -0.03492    0.06662  -0.524

Correlation of Fixed Effects:
            (Intr) lng_t1 fm_sz1
lang_type1  -0.018
fam_size1    0.000  0.000
lng_typ1:_1  0.000  0.000  0.000
```

```
# SIMPLE NONWORDS

nw.smpl_null.model = lmer(value ~ 1 + (1|SubjID) ,
                          data= n250_nwsmpl, REML=FALSE)
summary(nw.smpl_null.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ 1 + (1 | SubjID)
   Data: n250_nwsmpl

     AIC      BIC   logLik deviance df.resid
  5144.8   5159.6  -2569.4   5138.8     1005

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.6813 -0.5297  0.0084  0.5082  5.1466

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 8.104    2.847
 Residual             8.159    2.856
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
```

```
            Estimate Std. Error t value
(Intercept)  -0.4552     0.3944  -1.154
```

```r
# Main effects models with random intercepts
nw.smpl_main.model = lmer(value ~ lang_type + fam_size + (1|SubjID) ,
                          data= n250_nwsmpl, REML=FALSE)
summary(nw.smpl_main.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type + fam_size + (1 | SubjID)
   Data: n250_nwsmpl

     AIC      BIC   logLik deviance df.resid
  5134.9   5159.4  -2562.4   5124.9     1003

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.6110 -0.5387 -0.0202  0.4916  5.0669

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 7.570    2.751
 Residual             8.071    2.841
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
            Estimate Std. Error t value
(Intercept) -0.44216    0.38178  -1.158
lang_type1  -0.73326    0.38178  -1.921
fam_size1    0.28911    0.08948   3.231

Correlation of Fixed Effects:
           (Intr) lng_t1
lang_type1 -0.018
fam_size1   0.000  0.000
```

```r
# Interaction effects models with random intercepts
nw.smpl_inter.model = lmer(value ~ lang_type * fam_size + (1|SubjID) ,
                           data= n250_nwsmpl, REML=FALSE)
summary(nw.smpl_inter.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type * fam_size + (1 | SubjID)
   Data: n250_nwsmpl

     AIC      BIC   logLik deviance df.resid
  5134.5   5164.0  -2561.2   5122.5     1002

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.5684 -0.5299 -0.0071  0.4871  5.0247

Random effects:
```

```
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 7.571    2.752
 Residual             8.051    2.837
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
                     Estimate Std. Error t value
(Intercept)          -0.44216    0.38178  -1.158
lang_type1           -0.73326    0.38178  -1.921
fam_size1             0.28911    0.08937   3.235
lang_type1:fam_size1 -0.13743    0.08937  -1.538

Correlation of Fixed Effects:
            (Intr) lng_t1 fm_sz1
lang_type1  -0.018
fam_size1    0.000  0.000
lng_typ1:_1  0.000  0.000  0.000
```

```
# COMPLEX NONWORDS

nw.cplx_null.model = lmer(value ~ 1 + (1|SubjID) ,
                          data= n250_nwcplx, REML=FALSE)
summary(nw.cplx_null.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ 1 + (1 | SubjID)
   Data: n250_nwcplx

     AIC      BIC   logLik deviance df.resid
  4850.0   4864.8  -2422.0   4844.0     1005

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.7570 -0.6195 -0.0018  0.5511  4.5243

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.766    2.401
 Residual             6.106    2.471
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.07054    0.33308   0.212
```

```
# Main effects models with random intercepts
nw.cplx_main.model = lmer(value ~ lang_type + fam_size + (1|SubjID) ,
                          data= n250_nwcplx, REML=FALSE)
summary(nw.cplx_main.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type + fam_size + (1 | SubjID)
   Data: n250_nwcplx
```

```
     AIC      BIC   logLik deviance df.resid 
  4818.0   4842.5  -2404.0   4808.0     1003 

Scaled residuals: 
    Min      1Q  Median      3Q     Max 
-3.6556 -0.6335 -0.0374  0.5874  4.4057 

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.411    2.326   
 Residual             5.901    2.429   
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.08127    0.32298   0.252
lang_type1  -0.60324    0.32298  -1.868
fam_size1   -0.44102    0.07651  -5.764

Correlation of Fixed Effects:
           (Intr) lng_t1
lang_type1 -0.018       
fam_size1   0.000  0.000
```

```r
# Interaction effects models with random intercepts
nw.cplx_inter.model = lmer(value ~ lang_type * fam_size  + (1|SubjID),
                           data= n250_nwcplx, REML=FALSE)
summary(nw.cplx_inter.model)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: value ~ lang_type * fam_size + (1 | SubjID)
   Data: n250_nwcplx

     AIC      BIC   logLik deviance df.resid 
  4812.0   4841.5  -2400.0   4800.0     1002 

Scaled residuals: 
    Min      1Q  Median      3Q     Max 
-3.5815 -0.6299 -0.0492  0.5936  4.3354 

Random effects:
 Groups   Name        Variance Std.Dev.
 SubjID   (Intercept) 5.413    2.327   
 Residual             5.852    2.419   
Number of obs: 1008, groups:  SubjID, 55

Fixed effects:
                     Estimate Std. Error t value
(Intercept)           0.08125    0.32297   0.252
lang_type1           -0.60321    0.32297  -1.868
fam_size1            -0.44102    0.07619  -5.788
lang_type1:fam_size1  0.21538    0.07619   2.827
```

```
Correlation of Fixed Effects:
            (Intr) lng_t1 fm_sz1
lang_type1  -0.018
fam_size1    0.000  0.000
lng_typ1:_1  0.000  0.000  0.000
```

Model Comparisons

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

```
Data: n250_words
Models:
cw_null.model: value ~ 1 + (1 | SubjID)
cw_main.model: value ~ lang_type + fam_size + (1 | SubjID)
              npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
cw_null.model    3 4555.6 4570.4 -2274.8   4549.6
cw_main.model    5 4555.2 4579.8 -2272.6   4545.2 4.4082  2     0.1104
```

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

```
Data: n250_words
Models:
cw_main.model: value ~ lang_type + fam_size + (1 | SubjID)
cw_inter.model: value ~ lang_type * fam_size + (1 | SubjID)
               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
cw_main.model     5 4555.2 4579.8 -2272.6   4545.2
cw_inter.model    6 4557.0 4586.5 -2272.5   4545.0 0.2747  1     0.6002
```

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

```
Data: n250_nwsmpl
Models:
nw.smpl_null.model: value ~ 1 + (1 | SubjID)
nw.smpl_main.model: value ~ lang_type + fam_size + (1 | SubjID)
                   npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
nw.smpl_null.model    3 5144.8 5159.6 -2569.4   5138.8
nw.smpl_main.model    5 5134.9 5159.4 -2562.4   5124.9 13.952  2  0.0009342 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
Data: n250_nwsmpl
Models:
nw.smpl_main.model: value ~ lang_type + fam_size + (1 | SubjID)
nw.smpl_inter.model: value ~ lang_type * fam_size + (1 | SubjID)
                    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
nw.smpl_main.model     5 5134.9 5159.4 -2562.4   5124.9
nw.smpl_inter.model    6 5134.5 5164.0 -2561.2   5122.5 2.3617  1     0.1243
```

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

```
Data: n250_nwcplx
Models:
nw.cplx_null.model: value ~ 1 + (1 | SubjID)
nw.cplx_main.model: value ~ lang_type + fam_size + (1 | SubjID)
                   npar  AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.cplx_null.model    3 4850 4864.8  -2422     4844
nw.cplx_main.model    5 4818 4842.5  -2404     4808 36.041  2  1.492e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
Data: n250_nwcplx
Models:
nw.cplx_main.model: value ~ lang_type + fam_size + (1 | SubjID)
nw.cplx_inter.model: value ~ lang_type * fam_size + (1 | SubjID)
                    npar  AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
nw.cplx_main.model     5 4818 4842.5  -2404     4808
nw.cplx_inter.model    6 4812 4841.5  -2400     4800 7.9572  1    0.00479 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```