Walkthrough: Correlations between individual effect sizes for different manipulations

Dan Mirman

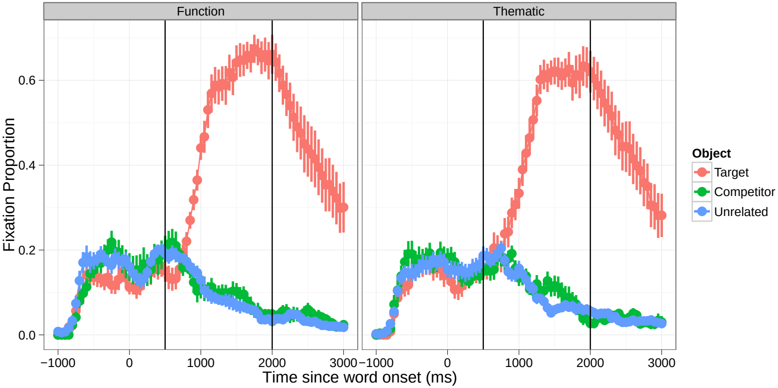
Load the data

Data from experiment that examined the time course of activation of **Function** and **Thematic** relations during spoken word-to-picture matching in 17 participants with left hemisphere stroke (Kalenine, Mirman, & Buxbaum, 2012, *Front. Hum. Neurosci.*, 6:106):

```
> library(lme4)
> load("FunctTheme.RData")
> summary(FunctTheme)
```

```
      subj          Condition            Object           Time           timeBin
 206     : 486   Function:4113   Target     :2733   Min.   :-1000   Min.   : 0
 281     : 486   Thematic:4086   Competitor:2733   1st Qu.:    0   1st Qu.:20
 419     : 486                   Unrelated :2733   Median : 1000   Median :40
 1088    : 486                                     Mean   : 1000   Mean   :40
 1238    : 486                                     3rd Qu.: 2000   3rd Qu.:60
 1392    : 486                                     Max.   : 3000   Max.   :80
 (Other):5283
    meanFix            sumFix             N
 Min.   :0.00000   Min.   : 0.000   Min.   :12.00
 1st Qu.:0.03125   1st Qu.: 1.000   1st Qu.:15.00
 Median :0.12500   Median : 2.000   Median :16.00
 Mean   :0.17768   Mean   : 3.261   Mean   :15.41
 3rd Qu.:0.25000   3rd Qu.: 5.000   3rd Qu.:16.00
 Max.   :1.00000   Max.   :16.000   Max.   :16.00
```

Plot the data



Orthogonal polynomial time

Orthogonal polynomials need to be defined for the specific analysis time window, so it is easier if we start by making a subset of the data that is just that critical time window (and only contains the critical non-target data). Then we can make a fourth-order orthogonal polynomial in the range of `timeBin` and insert it into the data frame aligned by `timeBin`:

```
> data.gca <- subset(FunctTheme,
+                 Time >= 500 & Time <= 2000 & Object != "Target")
> data.gca$timeBin <- data.gca$timeBin - 29
> t <- poly((unique(data.gca$timeBin)), 4)
> data.gca[,paste("ot", 1:4, sep="")] <- t[data.gca$timeBin, 1:4]
```

Fit the models

Fit separate models for the Function and Thematic conditions:

```
> m.funct <- lmer(meanFix ~ (ot1+ot2+ot3+ot4)*Object +
+               (ot1+ot2+ot3+ot4 | subj) + (ot1+ot2 | subj:Object),
+               data=subset(data.gca, Condition == "Function"), REML=F)
> m.theme <- lmer(meanFix ~ (ot1+ot2+ot3+ot4)*Object +
+               (ot1+ot2+ot3+ot4 | subj) + (ot1+ot2 | subj:Object),
+               data=subset(data.gca, Condition == "Thematic"), REML=F)
```

Random effects

The fitted model's random effects can be extracted using the `ranef` function, which has two elements corresponding to the two sets of random effects

```
> str(ranef(m.funct))

List of 2
 $ subj:Object:data.frame:        34 obs. of  3 variables:
  ..$ (Intercept): num [1:34] -0.032746 0.025387 0.00665 -0.020031 -0.000638 ...
  ..$ ot1        : num [1:34] 0.0339 0.0726 -0.0439 -0.0553 0.2467 ...
  ..$ ot2        : num [1:34] -0.15381 0.01431 -0.05961 0.11582 -0.00125 ...
 $ subj       :data.frame:        17 obs. of  5 variables:
  ..$ (Intercept): num [1:17] 0.02495 -0.01339 -0.00786 0.01495 -0.00855 ...
  ..$ ot1        : num [1:17] 0.0924 -0.0454 -0.0149 0.0451 -0.0236 ...
  ..$ ot2        : num [1:17] -0.1915 0.0375 0.0488 -0.105 0.0404 ...
  ..$ ot3        : num [1:17] 0.1124 -0.0677 0.0233 0.0253 -0.0126 ...
  ..$ ot4        : num [1:17] 0.0386 0.0474 -0.0504 0.0491 -0.011 ...
 - attr(*, "class")= chr "ranef.mer"

> head(ranef(m.funct)$subj:Object)

                (Intercept)         ot1          ot2
206:Competitor -0.0327464432  0.03386123 -0.153810857
206:Unrelated   0.0253874558  0.07261604  0.014311448
281:Competitor  0.0066498899 -0.04394898 -0.059606049
281:Unrelated  -0.0200311352 -0.05529833  0.115819282
419:Competitor -0.0006384665  0.24667305 -0.001245722
419:Unrelated  -0.0092934643 -0.15294028  0.056835287
```

Effect sizes

The difference between the **Competitor** and **Unrelated** random effect provides an estimate of each individual participant's competition effect size (relative to the mean effect size). This will require a little data manipulation:

```
> blup.funct <- data.frame(
+                 colsplit(row.names(ranef(m.funct)$subj:Object),
+                   ":", c("Subject", "Object")),
+                 ranef(m.funct)$subj:Object)
```

Effect sizes

The difference between the **Competitor** and **Unrelated** random effect provides an estimate of each individual participant's competition effect size (relative to the mean effect size). This will require a little data manipulation:

```
> blup.funct <- data.frame(
+               colsplit(row.names(ranef(m.funct)$subj:Object),
+                 ":", c("Subject", "Object")),
+               ranef(m.funct)$subj:Object)
> ES.funct <- ddply(blup.funct, .(Subject), summarize,
+               Function_Intercept = X.Intercept.[Object=="Competitor"] -
+                 X.Intercept.[Object=="Unrelated"],
+               Function_Linear = ot1[Object=="Competitor"] -
+                 ot1[Object=="Unrelated"])
```

Effect sizes

The difference between the **Competitor** and **Unrelated** random effect provides an estimate of each individual participant's competition effect size (relative to the mean effect size). This will require a little data manipulation:

```
> blup.funct <- data.frame(
+                 colsplit(row.names(ranef(m.funct)$subj:Object),
+                   ":", c("Subject", "Object")),
+                 ranef(m.funct)$subj:Object)
> ES.funct <- ddply(blup.funct, .(Subject), summarize,
+                 Function_Intercept = X.Intercept.[Object=="Competitor"] -
+                   X.Intercept.[Object=="Unrelated"],
+                 Function_Linear = ot1[Object=="Competitor"] -
+                   ot1[Object=="Unrelated"])
> blup.theme <- data.frame(
+                 colsplit(row.names(ranef(m.theme)$subj:Object),
+                   ":", c("Subject", "Object")),
+                 ranef(m.theme)$subj:Object)
> ES.theme <- ddply(blup.theme, .(Subject), summarize,
+                 Thematic_Intercept = X.Intercept.[Object=="Competitor"] -
+                   X.Intercept.[Object=="Unrelated"],
+                 Thematic_Linear = ot1[Object=="Competitor"] -
+                   ot1[Object=="Unrelated"])
```

Effect sizes

The difference between the **Competitor** and **Unrelated** random effect provides an estimate of each individual participant's competition effect size (relative to the mean effect size). This will require a little data manipulation:

```
> blup.funct <- data.frame(
+                 colsplit(row.names(ranef(m.funct)$subj:Object),
+                   ":", c("Subject", "Object")),
+                 ranef(m.funct)$subj:Object)
> ES.funct <- ddply(blup.funct, .(Subject), summarize,
+                 Function_Intercept = X.Intercept.[Object=="Competitor"] -
+                   X.Intercept.[Object=="Unrelated"],
+                 Function_Linear = ot1[Object=="Competitor"] -
+                   ot1[Object=="Unrelated"])
> blup.theme <- data.frame(
+                 colsplit(row.names(ranef(m.theme)$subj:Object),
+                   ":", c("Subject", "Object")),
+                 ranef(m.theme)$subj:Object)
> ES.theme <- ddply(blup.theme, .(Subject), summarize,
+                 Thematic_Intercept = X.Intercept.[Object=="Competitor"] -
+                   X.Intercept.[Object=="Unrelated"],
+                 Thematic_Linear = ot1[Object=="Competitor"] -
+                   ot1[Object=="Unrelated"])
> ES <- merge(ES.funct, ES.theme, by="Subject")
```

Effect size correlations

Now it is possible to test whether the Function and Thematic effect sizes are correlated:

```
> head(ES)

  Subject Function_Intercept Function_Linear Thematic_Intercept Thematic_Linear
1     206       -0.058133899     -0.03875480        0.030962905    -0.152874960
2     281        0.026681025      0.01134934        0.015367348     0.003994856
3     419        0.008654998      0.39961333       -0.001865295    -0.145994969
4    1088       -0.003282983     -0.15627606       -0.084598501    -0.061914859
5    1238       -0.013349166     -0.13985871       -0.022051096    -0.015555525
6    1392       -0.003196420      0.19122574        0.061526543    -0.353128827

> cor.test(ES$Function_Intercept, ES$Thematic_Intercept)

        Pearsons product-moment correlation

data:  ES$Function_Intercept and ES$Thematic_Intercept
t = -2.3602, df = 15, p-value = 0.03223
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.80074858 -0.05300061
sample estimates:
       cor
-0.5203887

> cor.test(ES$Function_Linear, ES$Thematic_Linear)

        Pearsons product-moment correlation

data:  ES$Function_Linear and ES$Thematic_Linear
t = -3.3571, df = 15, p-value = 0.004322
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8637204 -0.2544506
sample estimates:
       cor
-0.6549899
```

Scatterplots