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)

<table>
<thead>
<tr>
<th>subj</th>
<th>Condition</th>
<th>Object</th>
<th>Time</th>
<th>timeBin</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>Function:4113</td>
<td>Target :2733</td>
<td>Min. : -1000</td>
<td>Min. : 0</td>
</tr>
<tr>
<td>281</td>
<td>Thematic:4086</td>
<td>Competitor:2733</td>
<td>1st Qu.: 0</td>
<td>1st Qu.:20</td>
</tr>
<tr>
<td>419</td>
<td>486</td>
<td>Unrelated :2733</td>
<td>Median : 1000</td>
<td>Median :40</td>
</tr>
<tr>
<td>1088</td>
<td>486</td>
<td></td>
<td>Mean : 1000</td>
<td>Mean :40</td>
</tr>
<tr>
<td>1238</td>
<td>486</td>
<td></td>
<td>3rd Qu.: 2000</td>
<td>3rd Qu.:60</td>
</tr>
<tr>
<td>1392</td>
<td>486</td>
<td></td>
<td>Max. : 3000</td>
<td>Max. :80</td>
</tr>
<tr>
<td>(Other):5283</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>meanFix</th>
<th>sumFix</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. : 0.00000</td>
<td>Min. : 0.000</td>
<td>Min. :12.00</td>
</tr>
<tr>
<td>1st Qu.: 0.03125</td>
<td>1st Qu.: 1.000</td>
<td>1st Qu.:15.00</td>
</tr>
<tr>
<td>Median : 0.12500</td>
<td>Median : 2.000</td>
<td>Median :16.00</td>
</tr>
<tr>
<td>Mean : 0.17768</td>
<td>Mean : 3.261</td>
<td>Mean :15.41</td>
</tr>
<tr>
<td>3rd Qu.: 0.25000</td>
<td>3rd Qu.: 5.000</td>
<td>3rd Qu.:16.00</td>
</tr>
<tr>
<td>Max. : 1.00000</td>
<td>Max. :16.000</td>
<td>Max. :16.00</td>
</tr>
</tbody>
</table>
Plot the data

Fixation Proportion

Object

Target

Competitor

Unrelated

Function

Thematic

Time since word onset (ms)

-1000 0 1000 2000 3000

0 1000 2000 3000

0.0

0.2

0.4

0.6

0.8

1.0
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`:

```r
> 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:

```r
> 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

```r
> 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"
```

```r
> 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 -0.0092934643 -0.15294028 0.056835287
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:

```r
> 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:

```r
> 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:

```r
> 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:

```r
> 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:

```r
> head(ES)

       Subject Function_Intercept Function_Linear Thematic_Intercept Thematic_Linear
1          206          -0.058133899     -0.038754805        0.030962905   -0.152874960
2          281           0.026681025     0.011349340        0.015367348       0.003994856
3          419           0.008654998      0.399613330      -0.001865295       0.003994856
4         1088          -0.003282983     -0.156276060      -0.084598501      -0.061914859
5         1238          -0.013349166     -0.139858710      -0.022051096      -0.015555252
6         1392          -0.003196420      0.191225743       0.061526543       -0.353128827
```

```r
> 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
```

```r
> 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
```