Advanced mixed-models workshop: Session 2

Dale Barr

University of Glasgow

Bremen March 2015

Determining Maximal Random Effects

Barr, Levy, Scheepers, & Tily (2013); Barr (2013)

- Random intercept is needed whenever there are multiple observations per unit
- Any within-unit factor gets a random slope, *unless there is only one observation per level per unit*
- Between-unit factors do not get a random slope
- For each interaction, include a slope for the highest order combination of within-subject factors subsumed by the interaction
- For time-series data, include random slopes for time predictors if you have more than one time series per unit

Keysar, B., Barr, D. J., Balin, J. A., & Brauner, J. S. (2000). Taking perspective in conversation: The role of mutual knowledge in comprehension. *Psychological Science*, 11, 32–38.

- When interpreting expressions e.g. *the small candle*, do listeners experience interference?

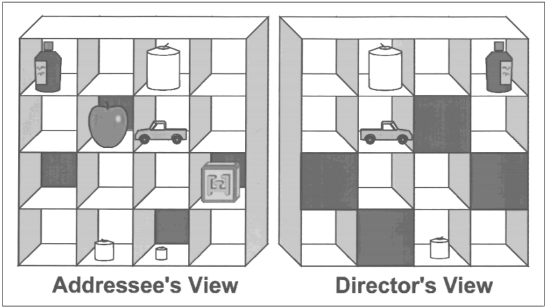


Fig. 1. The 16 slots with a typical set of objects. The addressee's and director's views are distinct because of the occluded slots. The critical instruction (referring to "the small candle") picks out a different candle from the director's perspective (shared candle) than from the addressee's perspective (occluded candle).

- 20 subjects, 12 items for each subject (N=240)
- one factor: condition (competitor vs. noncompetitor)
- data recorded using a 60 Hz eyetracker
- DV: latency to fixate the target object, measured from onset of the critical word

Field	Description
`SubjID`	Subject identifier (N=20)
`cond`	Condition (E=competitor, C=noncompetitor)
`crit`	Moment of onset of critical word (frames)
`targfix`	Moment the hand touched the target (frames)
`object`	Name of the experimental item

Keysar, Barr, Balin, & Brauner (2000)

Analysis Tasks

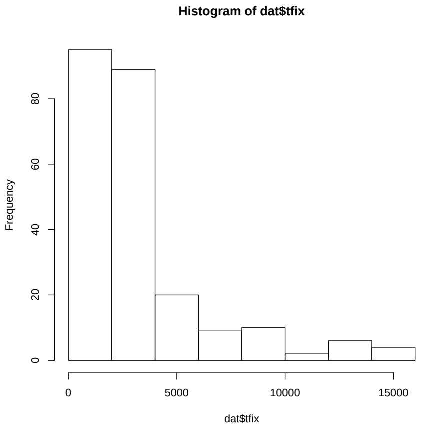
1. load the data from `KeysarEtAl2000.rds` into dataframe `dat`
2. calculate `tfix`, the latency for touching the target in milliseconds, store this in the dataframe `dat`
3. make histogram of `tfix`
4. create "truncated" versions of `tfix`, `tfixTrunc`, truncating values higher than the 97.5th percentile
5. calculate means in each condition

Linear mixed-model analysis

1. Now do the analysis using model comparison in a linear mixed effects model, with maximal random effects
   - tip: use deviation coding for condition
2. Derive *p*-values using:
   - Wald *z* statistic ("t-as-z")
   - Likelihood ratio tests
3. Would you say that subjects or items introduce more overall variation in grand mean target latencies?
4. Do subjects or items vary more in terms of the effect of condition (competitor)?
5. Look at the BLUPS.
   - Which items show the effect most strongly?
   - Which subjects?
   - Do all subjects show the effect?
   - Do all items show the effect?

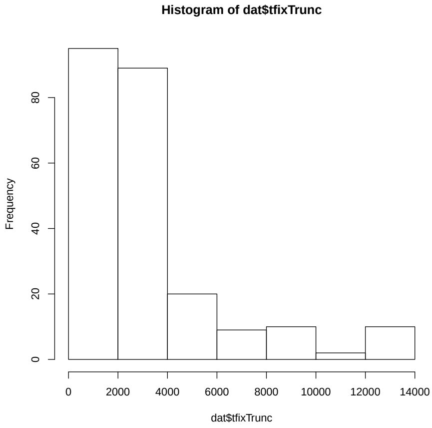
Load and preprocess

```
dat <- readRDS("kbbb.rds")
# calculate latencies
dat$tfix <- 1000*((dat$targfix - dat$crit) / 60)

hist(dat$tfix)
```



Clean up the latency data

```
# truncate outliers at 97.5th percentile of distribution
cutoff.tfix <- quantile(dat$tfix, probs=.975, na.rm=TRUE)
dat$tfixTrunc <- ifelse(dat$tfix>cutoff.tfix, cutoff.tfix, dat$tfix)

hist(dat$tfixTrunc)
```



Descriptive stats

```
# aggregate
aggregate(tfixTrunc ~ cond, dat, mean)
```

```
  cond tfixTrunc
1    C  2589.641
2    E  4036.625
```

Run t-test (aggregating first by subject)

```r
# re-create data: t-test
dat.subj <- aggregate(tfixTrunc ~ SubjID + cond, dat, mean)
dat.subj <- dat.subj[order(dat.subj$SubjID, dat.subj$cond), ]

dat.t <- t.test(subset(dat.subj, cond=="C")$tfixTrunc,
                subset(dat.subj, cond=="E")$tfixTrunc, paired=TRUE)
print(dat.t)
```

```
Paired t-test

data:  subset(dat.subj, cond == "C")$tfixTrunc and subset(dat.subj, cond == "E")$tf
t = -4.3608, df = 19, p-value = 0.0003364
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2129.4142  -748.2524
sample estimates:
mean of the differences
              -1438.833
```

Run linear mixed model

```r
# linear mixed model
# create deviation-coded predictor
dat$D <- dat$cond == "E"
dat$C2 <- dat$D - mean(dat$D)

library("lme4")
mod1 <- lmer(tfixTrunc ~ C2 +
                 (1 + C2 | SubjID) +
                 (1 + C2 | object),
             data=dat,
             subset = complete.cases(dat),
             REML=FALSE)
```

```
Loading required package: Matrix
Loading required package: Rcpp
```

View results

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: tfixTrunc ~ C2 + (1 + C2 | SubjID) + (1 + C2 | object)
   Data: dat
 Subset: complete.cases(dat)

     AIC      BIC   logLik deviance df.resid
  4421.9   4453.0  -2201.9   4403.9      226

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 SubjID   (Intercept)  417282   645.97
          C2           758341   870.83  1.00
 object   (Intercept)  616982   785.48
          C2             6765    82.25  1.00
 Residual             7236631  2690.10
Number of obs: 235, groups:  SubjID, 20; object, 12

Fixed effects:
            Estimate Std. Error t value
(Intercept)   3306.4      321.1  10.296
C2            1439.6      402.2   3.579
```

Wald z statistics

a.k.a. "t-as-z" method

```r
paramest <- fixef(mod1)
stderrs <- sqrt(diag(vcov(mod1)))
tstats <- paramest / stderrs
pvals <- 2 * (1 - pnorm(abs(tstats)))

data.frame(b = paramest, se = stderrs, t = tstats, p = pvals)
```

```
                   b       se         t            p
(Intercept) 3306.446 321.1306 10.296267 0.0000000000
C2          1439.572 402.2187  3.579078 0.0003448088
```

Likelihood ratio tests

```r
mod2 <- update(mod1, . ~ . -C2)

anova(mod1, mod2)

chi2 <- deviance(mod2) - deviance(mod1)
pchi <- pchisq(chi2, 1, lower.tail = FALSE)

c(chisq = chi2, p = pchi)
```

```
Data: dat
Subset: complete.cases(dat)
Models:
mod2: tfixTrunc ~ (1 + C2 | SubjID) + (1 + C2 | object)
mod1: tfixTrunc ~ C2 + (1 + C2 | SubjID) + (1 + C2 | object)
     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mod2  8 4430.4 4458.1 -2207.2   4414.4
mod1  9 4421.9 4453.0 -2201.9   4403.9 10.539      1   0.001169 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
       chisq           p
10.538911764  0.001168872
```

Blups

```r
blups <- ranef(mod1)
blups$SubjID$C2 + fixef(mod1)[2] # every subject shows effect
blups$object$C2 + fixef(mod1)[2] # every item shows effect
```

```
 [1] 1480.1108  781.4079  650.9276 1738.6800 1221.5055 2844.7430 1442.4837
 [8] 1182.7265 2373.9572 1758.3663 1279.2353 1600.2375  773.5742  689.4678
[15] 1279.2739 1720.9333 2675.1397 1351.8773 1152.1319  794.6589

[1] 1404.950 1377.372 1375.472 1403.316 1504.187 1478.570 1486.604 1397.842
 [9] 1417.357 1551.797 1347.852 1529.544
```

Additional stats

```
library("pbkrtest")

mod_kr <- KRmodcomp(mod1, mod2)

summary(mod_kr)
```

```
F-test with Kenward-Roger approximation; computing time: 1.14 sec.
large : tfixTrunc ~ C2 + (1 + C2 | SubjID) + (1 + C2 | object)
small : tfixTrunc ~ (1 + C2 | SubjID) + (1 + C2 | object)
         stat     ndf     ddf F.scaling  p.value
Ftest  12.4550  1.0000  8.9771         1 0.006448 **
FtestU 12.4550  1.0000  8.9771           0.006448 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Parametric bootstrap

```
mod_pb <- PBmodcomp(mod1, mod2)

summary(mod_pb)
```

```
There were 50 or more warnings (use warnings() to see the first 50)
Parametric bootstrap test; time: 156.28 sec; samples: 1000 extremes: 0;
Requested samples: 1000 Used samples: 996 Extremes: 0
large : tfixTrunc ~ C2 + (1 + C2 | SubjID) + (1 + C2 | object)
small : tfixTrunc ~ (1 + C2 | SubjID) + (1 + C2 | object)
           stat     df    ddf   p.value
PBtest   10.539                0.0010030 **
Gamma    10.539                0.0005716 ***
Bartlett 11.161  1.000         0.0008356 ***
F        10.539  1.000 -33.912
LRT      10.539  1.000         0.0011689 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Warning message:
In pf(Fobs, df1 = ndf, df2 = ddf) : NaNs produced
```