Simple mixed effect recipes

Block 4
Overview

• Give a brief introduction to the linear mixed effects models
  – From the context of experimental data, but hopefully generalisable to other sorts of data.

• Give brief introduction to Growth Curve Models

• Answer questions and play with data.
Traditional analyses in psycholinguistics

• By-participants ANOVA ($F_1$):
  Analysing condition means of all participants with participants as random variable.

• By-items ANOVA ($F_2$):
  Analysing condition means for all items with items as random variable.

We can't simultaneously generalise across both participants and items.

The reason is that we can't get an appropriate error term without averaging over either participants or items.
Mixed effect analyses

• Generalisation of (linear/logit) multiple regression.

• I am not an expert, and don't know much about the underlying rationale/mathematics.

• I just want to use it, and hopefully, I will start understanding the method better while I go along.

• After this presentation, you should hopefully be able to carry out your own, simple mixed-effect analyses.
Why mixed-effect analyses?

• Reviewers may ask you!

• We can simultaneously include participants and items as random variables -> allows us to generalise across both.

• ANOVAs aren't appropriate for dichotomous/binomial data. Logit mixed-effect models are.

• Well-suited for data that include missing responses.

• You can include continuous variables.

• Suitable for analysing unbalanced data sets (observational data/corpora).
How do we carry out mixed-effect analyses?

- We need to use R, a statistical programming language: http://www.r-project.org/
- Very powerful statistical package.
- It is possible to do some analyses in SPSS, but hardly anyone know how to do this, so you can't get any help.
Downloading R

- Go to R website: http://www.r-project.org/
Preparing a data file for R

- Each row should have a single observation/data point.
- Variables should be in the columns.
- Remove cases/trials that you don't want to analyse.
- Code dependent variable (response) as 0 or 1.
- Code independent variables as 0, 1, ...
- You can make the data file in Excel or other spreadsheet and save the file as .csv

My experiment:

Effect of transitivity (0 trans vs. 1 intrans) and repetition (0 rep vs. 1 non-rep) on response type (1 transitive vs. 0 intransitive).

Dependent variable (response) is dichotomous
Reading a data file

- R is case sensitive.
- # Allows you to comment.

library(languageR)
    # Opens LanguageR package, needed for analyses
    talkdata <- read.csv(file="H:\LMEtalk\mydatafile.csv",header=TRUE,sep=",")
    # Opens csv file, assigns the name talkdata for use in R

- Alternatively, you can open a tab-delimited text file:

    talkdata <- read.table("H:\LMEtalk\mydatafile.txt", header=TRUE)
    # Opens tab-delimited text file, assigns the name talkdata for use in R

- To run a command, you highlight it (with mouse or Ctrl-a), press right-mouse button, then "run line or selection".
A few basic commands

head(talkdata, n=10) # Shows rows 1-10

talkdata[20:40,] # Shows rows 20-20

tapply(talkdata$response, list(talkdata$repetition, talkdata$transitivity), mean)
  # Response mean broken down by repetition and transitivity

tapply(talkdata$response, list(talkdata$transitivity), mean)
  # Response mean broken down by transitivity

xtabs(~transitivity + repetition, data=talkdata)
  # Number of observations broken down by transitivity and repetition
> head(talkdata, n=6) # Shows rows 1-6
>  subject item transitivity repetition response grammaticality
> 1 9 1 0 0 0 1 6.6
> 2 17 1 0 0 0 1 6.6
> 3 25 1 0 0 0 1 6.6
> 4 47 1 0 0 0 0 6.6
> 5 14 2 0 0 0 1 6.4
> 6 22 2 0 0 0 0 6.4
> talkdata[20:25,] # Shows rows 20-45
>  subject item transitivity repetition response grammaticality
> 20 5 5 0 0 0 6.7
> 21 5 5 0 0 0 6.7
> 22 5 5 0 0 0 6.7
> 23 5 5 0 0 0 6.7
> 24 5 5 0 0 0 6.7
> 25 2 6 0 0 0 1 5.7
> tapply(talkdata$response, list(talkdata$repetition, talkdata$transitivity), mean)
> 0 1
> 0 0.6109325 0.3798701
> 1 0.6262626 0.5281438
> # Response mean broken down by repetition and transitivity
> tapply(talkdata$response, list(talkdata$transitivity), mean)
> 0 1
> 0 0.6184211 0.4527587
> # Response mean broken down by transitivity
> xtabs(~transitivity + repetition, data=talkdata)
> repetition
> transitivity   0  1
> 0 311 297
> 1 308 306
> # Number of observations broken down by transitivity and repetition
Coding participants and items as factors

- I want to ensure that R treats participants and items as factors rather than continuous variables.

```
talkdata$subject=as.factor(talkdata$subject)  # convert subject to factor
talkdata$item=as.factor(talkdata$item)     # convert item to factor
```
Mixed effect analyses: Effect coding

• If you have more than one independent variable, then it's important that the variables are centred and sum coded.

• Effect coding of 2-level variable: Mean = 0, range = 2.
• In a completely balanced design, transitive = -1, intransitive = 1. However, design isn't completely balanced due to exclusions. And non-experimental data won’t
• As a result, the intercept in the LME model is the (logit) grand mean.

• If you don't do this, then the effects that LME shows are not interpretable like main effects in ANOVAs! (They show simple effects instead.)
Mixed effect analyses: Effect coding

- `talkdata$transitivityc <- scale(as.numeric(talkdata$transitivity))` # effect code transitivity
- `talkdata$repetitionc <- scale(as.numeric(talkdata$repetition))` # effect code repetition

```r
> tapply(talkdata$response, list(talkdata$repetitionc, talkdata$transitivityc), mean)
  -1.004510833359817 0.994694766820338
  -0.086587350734082 0.6109325 0.3798701
  1.01276549729756 0.6262625 0.5261430
>
# Response mean broken down by repetition and transitivity
```

- I now have 2 new variables (transitivityc and repetitionc) which I am going to use in my mixed effect analyses.
Running LME analyses

Model you make  Dependent var.  Fixed variables (centred)
- talkdatalmer1 <- lmer(response ~ repetitionc*transitivityc + (1|subject) + (1|item), family = binomial, data = talkdata)

Random variables

Logistic regression, for binomial data.

Predicted variable is logit ln(p/(p-1))

- summary(talkdatalmer1)  # Shows output of model
> talkdata <- lmer(response ~ repetitions*transitivityc + (1|subject) + (1|item), family = binomial, data = talkdata)
> summary(talkdata)

Generalized linear mixed model fit by the Laplace approximation

Formula: response ~ repetitions * transitivityc + (1 | subject) + (1 | item)

Data: talkdata

AIC  BIC   logLik deviance
1264 1294 -625.9   1252

Random effects:

Groups   Name        Variance  Std.Dev.
subject  (Intercept) 2.8545     1.6895
          item         1.2301     1.1091
Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:

                         Estimate Std. Error z value Pr(>|z|)
(Intercept)               0.28230    0.34051   0.829   0.40830
repetitions               0.07467    0.07587   0.982   0.32556
transitivityc            -0.23941    0.07587  -3.175   0.00151 **
repetitions:transitivityc -0.01001    0.07587  -0.132   0.89412

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

                      (Intr) repetitions transivityc repetitions:transivityc
repetitions          -0.005
transitivityc        -0.009  -0.043
repetitions:transivityc -0.001  0.003  -0.051
Random intercepts and random slopes

talkdatalmer1 <- lmer(response ~ repetitionc*transitivityc + (1|subject) + (1|item),
family = binomial, data = talkdata)

• This model has a random intercept for each subject and each item.

• However, a few subjects or items can result in a significant effect: Not what we want.

• We need to check whether including a random slope for each subject or item results in a better model (usually, it doesn't):

talkdatalmerRTsubslope=lmer(response ~repetitionc*transitivityc +
(repetitionc*transitivityc + 1|subject) + (1|item), data=talkdata, family = "binomial")

Random intercept and slope for each subject in each condition
Random intercepts and random slopes

- New model shows similar results:

```r
> talkdata <- nlsRsibls <- lmer(response ~ repititions + transitivityc + (repititions + transitivityc + 1 | subject) + (1 | item), data = talkdata, fam = summary(talkdata)[“R1subls”])
```

Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repititions + transitivityc + (repititions + transitivityc + 1 | subject) + (1 | item)
Data: talkdata

\[
\text{AIC} = 1345 \quad \text{BIC} = 1345 - 619.4 \quad \text{logLik} = 1239
\]

Random effects:

<table>
<thead>
<tr>
<th>Groups</th>
<th>Name</th>
<th>Variance</th>
<th>Std.Dev.</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>(Intercept)</td>
<td>3.159534</td>
<td>1.77762</td>
<td></td>
</tr>
<tr>
<td></td>
<td>repititions</td>
<td>0.163875</td>
<td>0.40431</td>
<td>-0.735</td>
</tr>
<tr>
<td></td>
<td>transitivityc</td>
<td>0.092625</td>
<td>0.30434</td>
<td>0.096 -0.323</td>
</tr>
<tr>
<td></td>
<td>repititions:transitivityc</td>
<td>0.050401</td>
<td>0.22451</td>
<td>0.700 -0.191 -0.595</td>
</tr>
<tr>
<td>item</td>
<td>(Intercept)</td>
<td>1.265351</td>
<td>1.12509</td>
<td></td>
</tr>
</tbody>
</table>

Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:

|                  | Estimate | Std. Error | z value | Pr(>|z|) |
|------------------|----------|------------|---------|---------|
| (Intercept)      | 3.949391 | 0.38391    | 10.05   | 0.000001*** |
| repititions      | 0.25901  | 0.10069    | 2.56    | 0.00394*   |
| transitivityc    | -0.57726 | 0.09240    | -6.24   | 4.17e-10*** |
| repititions:transitivityc | 0.30209 | 0.08567    | 3.52    | 0.000322*** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

\[
\begin{array}{ccc}
\text{(Inter)} & \text{rpttns} & \text{trns} \\
\hline
\text{rpttns}     & -0.399 & . \\
\text{trns}      & 0.035 & -0.113 \\
\text{rpttns:trns} & 0.238 & -0.092 & -0.265 \\
\end{array}
\]
Model comparison

• Which model should we use?
  - The simplest model (random intercepts only), except if the more complex model has a better fit.

• Comparing the models' fit:
  anova(talkdatalmer1, talkdatalmerRTsubslope)

```
> > anova(talkdatalmer1, talkdatalmerRTsubslope)
Data: talkdata
Models:
talkdatalmer1: response ~ repetitions * transitivityc + (1 | subject) + (1 |
talkdatalmer1: item)
talkdatalmerRTsubslope: response ~ repetitions * transitivityc + (repetitions * transitivityc +
talkdatalmerRTsubslope: 1 | subject) + (1 | item)
                  Df    AIC    BIC  logLik   Chisq Chi Df Pr(>Chisq)
talkdatalmer1     6  1263.3  1294.4 -625.69    12.95  12.95  9e-05

talkdatalmerRTsubslope 15  1268.3  1345.4 -619.40   12.970 12.970  9e-05
```

• Log likelihood ratio test shows that the more complex model doesn't significantly improve the fit ($\chi^2(9) = 12.95, p = .16$).
  This takes into account the model's complexity (df).
More on random slopes

• After determining whether random slopes need to be included for subjects, you need to do the same for items:
  talkdatalmerRTitemslope=lmer(response ~repetitionc*transitivityc + (1|subject) + (repetitionc*transitivityc + 1|item), data=talkdata, family = "binomial")

• In fact, it may be best to include random slopes using forward selection (both for subjects and items):
  - transitivity
  - repetition
  - transitivity x repetition
Add random slopes for a variable or interaction if it improves the fit relative to the simpler model.

• Different ways of adding random slopes. Important to report what you did.
Simple effects

- If you don't effect-code your variables, but treatment-code them as 0 and 1, then you get simple effects.
  0 = repeated, 1 = non-repeated
  0 = transitive, 1 = intransitive
- Repetition = Effect of repetition in transitive conditions
- Transitivity = Effect of transitivity in repeated conditions

Problem: You tend to get correlations between the fixed effects -> Problems with collinearity
Simple effects

• Probably simpler to split the data into two data sets, e.g., transitive-only and intransitive-only, and do two separate analyses. Note that you lose some power.

trans <- talkdata[(talkdata$transitivity == "0"),]  # Select transitive only

```r
> # select only transitive.
> trans <- talkdata[(talkdata$transitivity == "0"),]
> # Redo effect coding, because half of trials is excluded
> trans$repetitionc <- scale(as.numeric(trans$repetition))  # effect code repetition
> # Simple effect with random intercepts
> translmer <- lmer(response ~ repetitionc + (1|subject) + (1|item), family = binomial, data = trans)
> summary(translmer)

Generalized linear mixed model fit by the Laplace approximation  
Formula: response ~ repetitionc + (1 | subject) + (1 | item)  
Data: trans
   AIC  BIC logLik deviance
645.6 663.3 -310.3    627.6
Random effects:
 Groups  Name        Variance Std.Dev.  
subject (Intercept) 3.2089   1.7813
item   (Intercept) 1.6578   1.2878
Number of obs: 605, groups: subject, 40; item, 32

Fixed effects:
 Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.85795   0.38182   2.273   0.023 *
repetitionc  0.05698   0.10732   0.531    0.595
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
 (Intr)
repetitionc -0.005
```
Simple effects

\[
\text{trans} \leftarrow \text{talkdata}[(\text{talkdata}$\text{transitivity} == "1"),] \quad \# \text{Select intransitive only}
\]

> # select only intransitive.
> incrans <- talkdata[(talkdata$transitivity == "1"),]
> # Adds effect coding, because half of trials is excluded
> incrans$repetitions <- scale(as.numeric(intrans$repetition)) \# effect code repetition
> # Simple effect with random intercepts
> intranlmer <- lmer(response ~ repetitions + (1|subject) + (1|item), family = binomial, data = incrans)
> summary(intranlmer)

Generalized linear mixed model fit by the Laplace approximation

Formula: response ~ repetitions + (1 | subject) + (1 | item)

Data: incrans

AIC  BIC logLik deviance
689.8 707.5  -340.9   681.8

Random effects:
  Groups   Name Variance Std.Dev.   
  subject (Intercept) 2.79317   1.67307
  item    (Intercept) 0.79125   0.88954

Number of obs: 614, groups: subject, 40; item, 32

Fixed effects:

  Estimate Std. Error  z value Pr(>|z|)
(Intercept)   -0.2552    0.3253  -0.784    0.432
repetitions    0.5052    0.1039   4.889   1.01e-06 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

  |   |
  | Intr|
repetitions -0.022

• Remember to check whether random slopes for participants and items improve fit.
Adding a continuous independent variable

• Analysing a continuous independent variable is the same as analysing a categorical variable:

```r
talkdatalmercont <- lmer(response ~ repetitionc*transitivityc*grammaticalityc + (1|subject) + (1|item), family = binomial, data = talkdata)
```

• If you include a continuous variable, this doesn't mean that you control for/partial out this variable (unlike in an ANCOVA). To do this, you'd have to residualise responses for grammaticality.
Adding a continuous independent variable

```r
> # Main effects with grammaticality as continuous variable
> talkdata$grammaticalityc <- scale(as.numeric(talkdata$grammaticality)) # effect code grammaticality
> # interaction including random intercepts
> talkdatainteractionc <- lmer(response ~ repetitions*transitivity*grammaticalityc + (1|subject) + (1|item), family = binomial, data = talkdata)
> summary(talkdatainteractionc)
```

Generalized linear mixed model fit by the Laplace approximation

Formula: response ~ repetitions * transitivity * grammaticalityc + (1 | subject) + (1 | item)

Data: talkdata

|        | Estimate | Std. Error | z value | Pr(>|z|) |
|--------|----------|------------|---------|----------|
| (Intercept) | 0.232034 | 0.343955   | 0.662   | 0.5061   |
| repetitions | 0.234963 | 0.068330   | 2.663   | 0.00730  ** |
| transitivityc | -0.825667 | 0.068305   | -5.601  | 2.1e-11  *** |
| grammaticalityc | -0.160675 | 0.118905   | -1.350  | 0.17799  |
| repetitions:transitivityc | 0.243239 | 0.068645   | 2.746   | 0.00609  ** |
| repetitions:grammaticalityc | 0.002888 | 0.101182   | 0.003   | 0.99845  |
| transitivityc:grammaticalityc | -0.070167 | 0.119390   | -0.580  | 0.56166  |
| repetitions:transitivityc:grammaticalityc | -0.108573 | 0.010146   | -1.043  | 0.29710  |

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:

```
   | (Intercept) | repetitions | transivityc | grammaticalityc | repetitions:transivityc | repetitions:grammaticalityc | transivityc:grammaticalityc | repetitions:transivityc:grammaticalityc |
---|-------------|------------|-------------|-----------------|------------------------|----------------------------|-------------------------------|----------------------------------------|
(Intercept) | 1.0000000  | 0.0020290  | -0.0512730   | -0.0352030          | -0.0009000               | -0.0252030                 | -0.0009000                  | -0.0252030       |
repetitions  | 0.0020290  | 1.0000000  | -0.0009000   | -0.0660100          | 0.0009000                | 0.0009000                  | 0.0009000                   | 0.0009000       |
transivityc  | -0.0512730 | -0.0009000 | 1.0000000    | 0.0009000           | 0.0009000                | 0.0009000                  | 0.0009000                   | 0.0009000       |
grammaticalityc | -0.0352030 | -0.0660100 | 0.0009000    | 1.0000000           | 0.0009000                | 0.0009000                  | 0.0009000                   | 0.0009000       |
repetitions:transivityc | -0.0009000 | 0.0009000 | 0.0009000    | 0.0009000           | 1.0000000                | 0.0009000                  | 0.0009000                   | 0.0009000       |
repetitions:grammaticalityc | -0.0252030 | 0.0009000 | 0.0009000    | 0.0009000           | 0.0009000                | 1.0000000                  | 0.0009000                   | 0.0009000       |
transivityc:grammaticalityc | -0.0009000 | 0.0009000 | 0.0009000    | 0.0009000           | 0.0009000                | 0.0009000                  | 1.0000000                   | 0.0009000       |
repetitions:transivityc:grammaticalityc | -0.0252030 | 0.0009000 | 0.0009000    | 0.0009000           | 0.0009000                | 0.0009000                  | 0.0009000                   | 1.0000000       |
```
Collinearity (see Florian Jaeger's slides)

- A predictor is collinear with other predictors in the model if there are high (partial) correlations between them.

- Even if a predictor is not highly correlated with any single other predictor in the model, it can be highly collinear with the combination of predictors -> collinearity will affect the predictor.

- If you have collinearity, then it's difficult to determine cause-effect relations (e.g., are reading times affected by frequency or length?)

- No collinearity in balanced designs -> good reason for doing experiments.
Checking for collinearity

- First check column numbers:
  head(talkdata, n=6) # Shows rows 1-6

  cor(talkdata[,c(5,7,8,9)]) # Correlation for relevant columns

```r
> head(talkdata, n=6) # Shows rows 1-6
table: item transitivity repetition response grammaticality transitivityc repetitionc grammaticalityc
1   9   1      0        0   1    6.6   -1.004511   -0.9855874    0.6442307
2   17  1      0        0   1    6.6   -1.004511   -0.9855874    0.6442307
3   25  1      0        0   1    6.6   -1.004511   -0.9855874    0.6442307
4   47  1      0        0   0    6.6   -1.004511   -0.9855874    0.6442307
5   14  2      0        0   1    6.4   -1.004511   -0.9855874    0.3761851
6   22  2      0        0   0    6.4   -1.004511   -0.9855874    0.3761851
> cor(talkdata[,c(5,7,8,9)]) # Correlation for relevant columns

table: response transitivityc repetitionc grammaticalityc
response     1.00000000 -0.166082076  0.079685591  -0.009526330
transitivityc -0.16608208  1.000000000  0.009885222  -0.469733802
repetitionc   0.079685591  0.009885222  1.000000000   0.006483904
grammaticalityc -0.009526330  -0.469733802  0.006483904  1.000000000
```
Checking for collinearity

- Calculate K (Kappa):

  \[ \text{collin.fnc(talkdata[,c(5,7,8,9)])$cnumber} \]

  \[
  > \text{collin.fnc(talkdata[,c(5,7,8,9)])$cnumber
  [1] 2.638989
  \]

  K > 15: Medium collinearity
  K > 30: Potentially harmful collinearity

- Not sure what to do when you have collinearity: See Florian Jaeger's slides for suggestions.
```r
library(languageR)

talkdata <- read.csv(file="H:\Roger\talks\LMEtalk\mydatafile.csv",header=TRUE,sep="\")

# Responses mean broken down by repetition and transitivity

tapply(talkdata$response, list(talkdata$repetition, talkdata$transitivity), mean)
# Response mean broken down by repetition and transitivity

tapply(talkdata$response, list(talkdata$transitivity), mean)
# Response mean broken down by transitivity

xtabs(~transitivity + repetition, data=talkdata)
# Number of observations broken down by transitivity and repetition

# Convert to factors

talkdata$subject<- as.factor(talkdata$subject)  # convert subject to factor

talkdata$item<- as.factor(talkdata$item)  # convert item to factor

# effect code so that they can be interpreted as main effects in ANOVAs

talkdata$transitivityc <- scale(as.numeric(talkdata$transitivity)) # effect code transitivity
talkdata$repetitionc <- scale(as.numeric(talkdata$repetition)) # effect code repetition

# Response mean broken down by repetition and transitivity

tapply(talkdata$response, list(talkdata$repetitionc, talkdata$transitivityc), mean)
# Response mean broken down by transitivity and repetition

# model that includes random intercepts and slopes for each subject

talkdatalmer1 <- lmer(response ~ repetitionc * transitivityc + (1|subject) + (1|item), family = binomial, data = talkdata)
summary(talkdatalmer1)

# model that includes random intercepts and slopes for each item

talkdatalmerRTitemslope <- lmer(response ~ repetitionc * transitivityc + (1|subject) + (repetitionc * transitivityc + 1|item), data=talkdata, family = "binomial")
summary(talkdatalmerRTitemslope)

# to check whether the models differ (they don't)
# in addition, BIC is lower for simpler model, so this is the better model
anova(talkdatalmer1, talkdatalmerRTitemslope)

# model that includes random intercepts and slopes for each subject

talkdatalmerRTsubslope <- lmer(response ~ repetitionc * transitivityc + (repetitionc * transitivityc + 1|subject) + (1|item), data=talkdata, family = "binomial")
summary(talkdatalmerRTsubslope)

# to check whether the models differ (they don't)
# in addition, BIC is lower for simpler model, so this is the better model
anova(talkdatalmer1, talkdatalmerRTsubslope)

# Simple effect

talkdatasimple1 <- lmer(response ~ repetition * transitivity + (1|subject) + (1|item), family = binomial, data = talkdata)
summary(talkdatasimple1)

# Simple effect on 2 separate data sets

trans <- talkdata[talkdata$transitivity == "0"]

trans$repetitionc <- scale(as.numeric(trans$repetition)) # effect code repetition

# Simple effect with random intercepts

translmer <- lmer(response ~ repetitionc + (1|subject) + (1|item), family = binomial, data = trans)
summary(translmer)

intrans <- talkdata[talkdata$transitivity == "1"]

intrans$repetitionc <- scale(as.numeric(intrans$repetition)) # effect code repetition

# Simple effect with random intercepts

intranslmer <- lmer(response ~ repetitionc + (1|subject) + (1|item), family = binomial, data = intrans)
summary(intranslmer)

# Main effects with grammaticality as continuous variable

talkdata$grammaticalityc <- scale(as.numeric(talkdata$grammaticality)) # effect code grammaticality

# interaction including random intercepts

talkdatalmercont <- lmer(response ~ repetitionc * transitivityc * grammaticalityc + (1|subject) + (1|item), family = binomial, data = talkdata)
summary(talkdatalmercont)

head(talkdata, n=6) # Shows rows 1-6
cor(talkdata[,c(0,7,8,9)]) # Correlation for relevant columns
```

Analysis of variables with more than 3 levels

- Let's say I have the following variables:
  - Condition: 3 levels: baseline, transitive, intransitive
  - Verb: 2 levels

- Centre verb variable; it doesn't matter whether you centre condition.

- Code condition as a factor (otherwise R considers it a continuous variable):
  data3levels$condition=as.factor(data3levels$condition)

- LME syntax is the same as with 2-level variables:
  levels3lmer1 <- lmer(response ~ condition*verbc + (1|subject) + (1|item), family = binomial, data = data3levels)
Analysis of variables with more than 3 levels

```r
> levels3lmer1 <- lmer(response ~ condition*verb + (1|subject) + (1|item), family = binomial, data = data3levels)
> summary(levels3lmer1)

Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ condition * verb + (1 | subject) + (1 | item)
    Data: data3levels

        AIC      BIC   logLik deviance
875.8 915.9 -429.9   859.8

Random effects:
 Groups   Name     Variance Std.Dev. 
subject  (Intercept) 4.7759   2.1854
        verb              2.1067   1.4523
        condition2        0.9535   0.9768
        condition3        1.5847   1.2604
        condition2:verb   0.2490   0.4989
        condition3:verb   0.3037   0.5509
        condition2:condition3 -0.0361 0.1901
        condition3:condition3 -0.0150 0.1234

Number of obs: 1002, groups: subject 36, item 30

Fixed effects:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.4342    0.4949   -0.88 0.3803
condition2        2.2613    0.2522    8.97 < 2e-16 ***
condition3        1.6067    0.2374    6.76 1.32e-11 ***
verb              -0.2808    0.3343   -0.84 0.4009
condition2:verb   0.4535    0.2404    1.88 0.0692 .
condition3:verb   0.5412    0.2310    2.34 0.0191 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
   (Intr) condtn2 condtn3 verb cndt2:
condition2  -0.211
condition3  -0.215  0.512
verb        0.008 -0.018 -0.008
condtn2:verb -0.006  0.089  0.026 -0.315
condtn3:verb -0.007  0.061  0.052 -0.321  0.469
```

Contrast conditions 1 (base) and 2 (trans)
Contrast conditions 1 (base) and 3 (intrans)
Main effect of verb
Interaction between verb and cond 1 vs. 2
Interaction between verb and cond 1 vs. 3

You don't get a main effect of condition!
Analysis of variables with more than 3 levels

• If you'd like to run different contrasts, then recode the conditions:
  
  # recode conditions so that trans = 1, baseline = 2, intrans = 3
  data3levels$newcond = 3
  data3levels$newcond = ifelse(data3levels$condition == 1, "2", data3levels$newcond)
  data3levels$newcond = ifelse(data3levels$condition == 2, "1", data3levels$newcond)

• As always, whether random slopes improve the model.
Analysing continuous dependent variables

- Syntax of lmer command the same as before, but we now use a linear rather than logit model.
  - We remove family = binomial from command

- Let's assume that acceptability is a continuous, normally distributed variable.
  levels3lmer1 <- lmer(acceptability ~ condition*verbc + (1|subject) + (1|item), data = data3levels)
Unfortunately, you don't get p-values.

We could assume that p < .05 if t > 2.
More about mixed effect models and R

- Florian Jaeger's website and slides: http://www.bcs.rochester.edu/people/fjaeger/
- Mark Gardener's website: http://www.gardenersown.co.uk/Education/Lectures/R/index.htm#introduction
- Jaeger, F. (2008): Categorical data analysis: Away from transformations (transformations or not) and towards logit mixed models *Journal of Memory and Language*, 59, 434-446.
- R-lang mailing list archives: http://pidgin.ucsd.edu/pipermail/r-lang/