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?

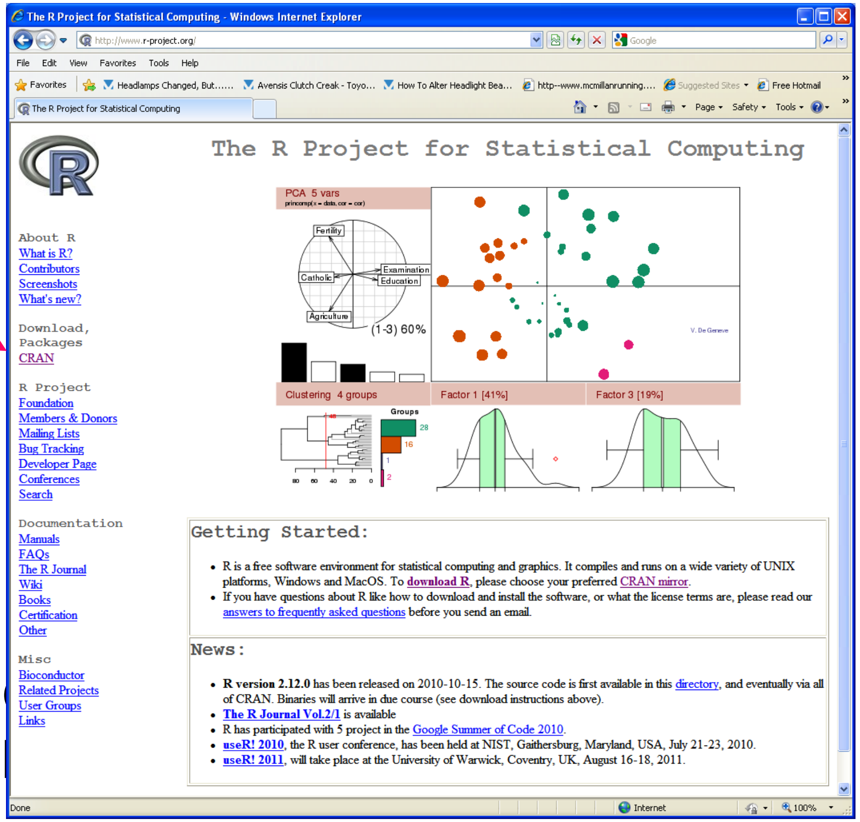
- We need to use R, a statistical programming language: http://www.r-project.org/

- 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

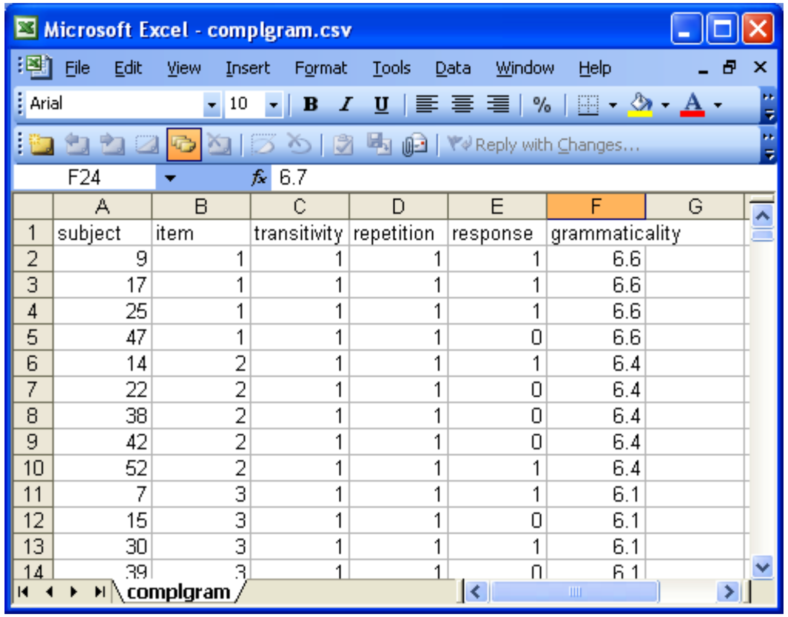
- Go to R website: http://www.r-project.org/



-

Preparing a data file for R

- Each row should have a single observation/data point.
- Variables should be in the columns.



	A	B	C	D	E	F	G
1	subject	item	transitivity	repetition	response	grammaticality	
2	9	1	1	1	1	6.6	
3	17	1	1	1	1	6.6	
4	25	1	1	1	1	6.6	
5	47	1	1	1	0	6.6	
6	14	2	1	1	1	6.4	
7	22	2	1	1	0	6.4	
8	38	2	1	1	0	6.4	
9	42	2	1	1	0	6.4	
10	52	2	1	1	1	6.4	
11	7	3	1	1	1	6.1	
12	15	3	1	1	0	6.1	
13	30	3	1	1	1	6.1	
14	39	3	1	1	0	6.1	

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)

- 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

Reading a data file

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

Set to "FALSE" if you don't have a header

```
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
```

R Console

```
> head(talkdata, n=6) # Shows rows 1-6
  subject item transitivity repetition response grammaticality
1       9    1            0          0        1            6.6
2      17    1            0          0        1            6.6
3      25    1            0          0        1            6.6
4      47    1            0          0        0            6.6
5      14    2            0          0        1            6.4
6      22    2            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      13    5            0          0        0            6.7
22      21    5            0          0        0            6.7
23      37    5            0          0        0            6.7
24      51    5            0          0        0            6.7
25       2    6            0          0        1            5.7
>
> tapply(talkdata$response, list(talkdata$repetition, talkdata$transitivity), mean)
          0         1
0 0.6109325 0.3798701
1 0.6262626 0.5261438
>     # Response mean broken down by repetition and transitivity
>
> tapply(talkdata$response, list(talkdata$transitivity), mean)
        0         1
0.6184211 0.4527687
>     # 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

```
> tapply(talkdata$response, list(talkdata$repetitionc, talkdata$transitivityc), mean)
                  -1.00451083359817 0.994694766820338
-0.986587390734082         0.6109325         0.3798701
1.01276549728756           0.6262626         0.5261438
>     # 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.

Your data set

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

- summary(talkdatalmer1) # Shows output of model

Output

```
> talkdatalmer1 <- lmer(response ~ repetitionc*transitivityc + (1|subject) + (1|item), family = binomial, data = talkdata)
> summary(talkdatalmer1)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repetitionc * 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    (Intercept) 1.2301   1.1091
Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                0.26230    0.34051   0.770 0.441109
repetitionc                0.28162    0.07467   3.772 0.000162 ***
transitivityc             -0.55342    0.07587  -7.294 3.01e-13 ***
repetitionc:transitivityc  0.24455    0.07474   3.272 0.001068 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) rpttnc trnstv
repetitionc -0.005
transitvtyc -0.009 -0.043
rpttnc:trns -0.001  0.003 -0.061
```

Information about goodness of fit of model

z- and p-values for independent variables

Fixed-effect correlations: High correlations are problematic (but won't occur in a balanced design).

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:

```
> talkdatalmerRTsubslope=lmer(response ~repetitionc*transitivityc + (repetitionc*transitivityc + 1|subject) + (1|item), data=talkdata, fam
> summary(talkdatalmerRTsubslope)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repetitionc * transitivityc + (repetitionc * transitivityc +      1 | subject) + (1 | item)
   Data: talkdata
  AIC  BIC logLik deviance
 1269 1345 -619.4     1239
Random effects:
 Groups  Name                       Variance Std.Dev. Corr
 subject (Intercept)                3.159934 1.77762
         repetitionc                0.163875 0.40481  -0.735
         transitivityc              0.092625 0.30434   0.098 -0.323
         repetitionc:transitivityc 0.050403 0.22451   0.700 -0.191 -0.598
 item    (Intercept)                1.265835 1.12509
Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                0.24939    0.35391   0.705 0.481008
repetitionc                0.25801    0.10069   2.562 0.010394 *
transitivityc             -0.57726    0.09240  -6.247 4.17e-10 ***
repetitionc:transitivityc  0.30209    0.08567   3.526 0.000422 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) rpttnc trnstv
repetitionc -0.399
transitvtyc  0.035 -0.113
rpttnc:trns  0.238 -0.042 -0.265
```

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 ~ repetitionc * transitivityc + (1 | subject) + (1 |
talkdatalmer1:     item)
talkdatalmerRTsubslope: response ~ repetitionc * transitivityc + (repetitionc * transitivityc +
talkdatalmerRTsubslope:     1 | subject) + (1 | item)
                       Df    AIC    BIC  logLik  Chisq Chi Df Pr(>Chisq)
talkdatalmer1           6 1263.8 1294.4 -625.89
talkdatalmerRTsubslope 15 1268.8 1345.4 -619.40 12.970      9     0.1640
```

- 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

```
> # Simple effect
> talkdatasimple1 <- lmer(response ~ repetition*transitivity + (1|subject) + (1|item), family = binomial, data = talkdata)
> summary(talkdatasimple1)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repetition * transitivity + (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    (Intercept) 1.2301   1.1091
Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)              0.78278    0.36539   2.142  0.03217 *
repetition               0.07184    0.21140   0.340  0.73400
transitivity            -1.58882    0.21783  -7.294 3.01e-13 ***
repetition:transitivity  0.97757    0.29874   3.272  0.00107 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) repttn trnstv
repetition  -0.291
transitivty -0.298  0.488
rpttn:trnst  0.214 -0.708 -0.719
```

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 loose some power.

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

```
> # 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 -318.8    637.6
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 3.2089   1.7913
 item    (Intercept) 1.6578   1.2876
Number of obs: 608, groups: subject, 40; item, 32

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.86796    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 -0.008
```

Simple effects

trans <- talkdata[(talkdata$transitivity == "1"),] # Select intransitive only

```
> # select only intransitive.
> intrans <- talkdata[(talkdata$transitivity == "1"),]
> # Redo effect coding, because half of trials is excluded
> intrans$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)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repetitionc + (1 | subject) + (1 | item)
   Data: intrans
   AIC   BIC logLik deviance
 689.8 707.5 -340.9    681.8
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 2.79917  1.67307
 item    (Intercept) 0.79128  0.88954
Number of obs: 614, groups: subject, 40; item, 32

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.2502     0.3253  -0.769    0.442
repetitionc   0.5082     0.1039   4.889 1.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
repetitionc -0.022
```

- Remember to check whether random slopes for participants and items improve fit.

Adding a continuous independent variable

- Analysing a continous independent variable is the same as analysing a categorical variable:

continuous variable

```
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

```
> # 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)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ repetitionc * transitivityc * grammaticalityc + (1 |      subject) + (1 | item)
   Data: talkdata
  AIC  BIC logLik deviance
 1266 1317 -622.8     1246
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 2.9329   1.7126
 item    (Intercept) 1.0993   1.0485
Number of obs: 1222, groups: subject, 40; item, 32

Fixed effects:
                                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                                0.212634   0.342055   0.622  0.53418
repetitionc                                0.236963   0.088330   2.683  0.00730 **
transitivityc                             -0.629067   0.095305  -6.601  4.1e-11 ***
grammaticalityc                           -0.160675   0.118005  -1.362  0.17333
repetitionc:transitivityc                  0.243393   0.088645   2.746  0.00604 **
repetitionc:grammaticalityc                0.003588   0.101182   0.035  0.97171
transitivityc:grammaticalityc             -0.078167   0.119393  -0.655  0.51266
repetitionc:transitivityc:grammaticalityc -0.105672   0.101346  -1.043  0.29710
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) rpttnc trnstv grmmtc rpttnc:t rpttnc:g trnst:
repetitionc -0.002
transitvtyc -0.051 -0.061
grammtcltyc -0.083 -0.041  0.600
rpttnc:trns  0.002 -0.106 -0.058 -0.035
rpttnc:grmm  0.003 -0.251 -0.014 -0.024  0.533
trnstvtyc:g  0.163  0.007 -0.328 -0.588  0.022    0.023
rpttnc:trn:  0.000  0.527 -0.012  0.008 -0.252   -0.534   -0.014
```

Few correlations, so collinearity probably not an issue.

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

Specifies columns numbers in data file

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

```
> head(talkdata, n=6) # Shows rows 1-6
  subject item transitivity repetition response grammaticality transitivityc repetitionc grammaticalityc
1       9    1            0          0        1            6.6     -1.004511  -0.9865874       0.6442307
2      17    1            0          0        1            6.6     -1.004511  -0.9865874       0.6442307
3      25    1            0          0        1            6.6     -1.004511  -0.9865874       0.6442307
4      47    1            0          0        0            6.6     -1.004511  -0.9865874       0.6442307
5      14    2            0          0        1            6.4     -1.004511  -0.9865874       0.3761851
6      22    2            0          0        0            6.4     -1.004511  -0.9865874       0.3761851
> cor(talkdata[,c(5,7,8,9)]) # Correlation for relevant columns
                   response transitivityc repetitionc grammaticalityc
response         1.00000000  -0.166062076 0.079685591    -0.009526330
transitivityc   -0.16606208   1.000000000 0.009885222    -0.469733802
repetitionc      0.07968559   0.009885222 1.000000000     0.006483904
grammaticalityc -0.00952633  -0.469733802 0.006483904     1.000000000
```

Checking for collinearity

- Calculate K (Kappa):

Specifies columns numbers in data file

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

```
> 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.

```
library(languageR)

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

# talkdata <-  read.table("H:\\Roger\\talks\\LMEtalk\\mydatafile.txt", header=TRUE)

head(talkdata, n=6) # Shows rows 1-6
talkdata[20:25,] # Shows rows 20-45

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 factor
talkdata$subject=as.factor(talkdata$subject)  # convert subject to factor
talkdata$item=as.factor(talkdata$item)  # convert item to factor

# effect code vars 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

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

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

# 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)

# 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)

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

### Simple effect on 2 separate data sets
# 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)

# select only intransitive.
intrans <- talkdata[(talkdata$transitivity == "1"),]
# Redo effect coding, because half of trials is excluded
intrans$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(5,7,8,9)]) # Correlation for relevant columns

# Checking for collinearity
```

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

```
> levels3lmer1 <- lmer(response ~ condition*verbc + (1|subject) + (1|item), family = binomial, data = data3levels)
> summary(levels3lmer1)
Generalized linear mixed model fit by the Laplace approximation
Formula: response ~ condition * verbc + (1 | subject) + (1 | item)
   Data: data3levels
   AIC BIC logLik deviance
 875.8 915 -429.9    859.8
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 4.7759   2.1854
 item    (Intercept) 2.6145   1.6169
Number of obs: 1002, groups: subject, 36; item, 30

Fixed effects:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.4342     0.4949  -0.877   0.3803
condition2         2.2613     0.2522   8.967  < 2e-16 ***
condition3         1.6067     0.2374   6.767 1.32e-11 ***
verbc             -0.2808     0.3343  -0.840   0.4009
condition2:verbc   0.4535     0.2404   1.887   0.0592 .
condition3:verbc   0.5412     0.2310   2.343   0.0191 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) cndtn2 cndtn3 verbc  cndt2:
condition2  -0.211
condition3  -0.215  0.512
verbc        0.008 -0.018 -0.008
cndtn2:vrbc -0.006  0.089  0.026 -0.315
cndtn3:vrbc -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)
```

Analysing continuous dependent variables

```
> levels3lmer1 <- lmer(acceptability ~ condition*verbc + (1|subject) + (1|item), data = data3levels)
> summary(levels3lmer1)
Linear mixed model fit by REML
Formula: acceptability ~ condition * verbc + (1 | subject) + (1 | item)
   Data: data3levels
  AIC  BIC logLik deviance REMLdev
 1617 1661 -799.5     1573    1599
Random effects:
 Groups   Name        Variance  Std.Dev.
 subject  (Intercept) 0.0024184 0.049177
 item     (Intercept) 0.1571061 0.396366
 Residual             0.2561925 0.506155
Number of obs: 1002, groups: subject, 36; item, 30

Fixed effects:
                  Estimate Std. Error t value
(Intercept)       5.768887   0.077943   74.01
condition2        0.694156   0.039120   17.74
condition3        0.690889   0.039303   17.58
verbc             0.003084   0.077568    0.04
condition2:verbc -0.037302   0.039330   -0.95
condition3:verbc -0.022939   0.039506   -0.58

Correlation of Fixed Effects:
            (Intr) cndtn2 cndtn3 verbc  cndt2:
condition2  -0.252
condition3  -0.251  0.500
verbc        0.010  0.004  0.004
cndtn2:vrbc  0.004 -0.006 -0.008 -0.255
cndtn3:vrbc  0.004 -0.008 -0.001 -0.254  0.501
```

- 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
- Baayen, R.H. (2008). *Analyzing linguistic data: A practical introduction to statistics using R.* Cambridge: Cambridge University Press.
- 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/