Introduction to statistics: Linear mixed models

Shravan Vasishth

Universität Potsdam
vasishth@uni-potsdam.de
http://www.ling.uni-potsdam.de/∼vasishth

April 12, 2020

Summary

1. We know how to do simple t-tests.
2. We know how to fit simple linear models.
3. We saw that the paired t-test is identical to the varying intercepts linear mixed model.

Now we are ready to look at linear mixed models in detail.

Linear models

Returning to our SR/OR relative clause data from English (Grodner and Gibson, Expt 1). First we load the data as usual (not shown).

```
gge1crit<-read.table("data/grodnergibson05data.txt",
                     header=TRUE)

gge1crit$so<-ifelse(gge1crit$condition=="objgap",1,-1)

dat<- gge1crit
dat$logrt<-log(dat$rawRT)

bysubj<-aggregate(logrt˜subject+condition,
                  mean,data=dat)
```

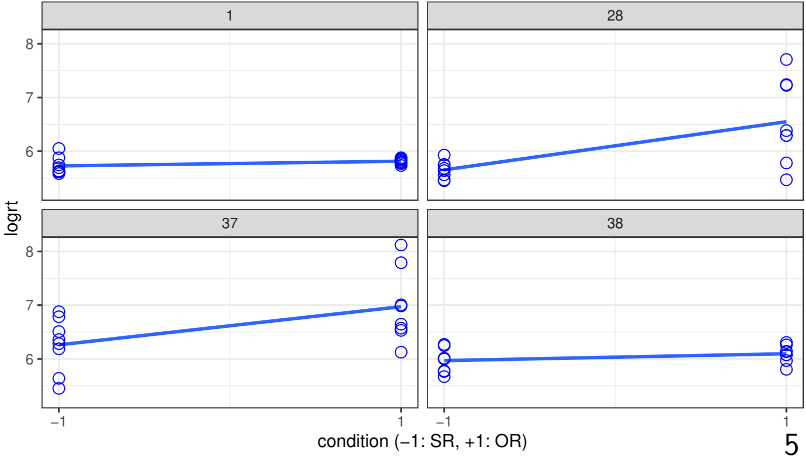
Linear models

The simple linear model (incorrect for these data):

```
summary(m0<-lm(logrt~so,dat))$coefficients

##             Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 5.883056   0.019052 308.7841 0.0000000
## so          0.062017   0.019052   3.2551 0.0011907
```

Linear models

We can visualize the different responses of subjects (four subjects shown):



Linear models

Given these differences between subjects, you could fit a separate linear model for each subject, collect together the intercepts and slopes for each subject, and then check if the intercepts and slopes are significantly different from zero.
We will fit the model using log reading times because we want to make sure we satisfy model assumptions (e.g., normality of residuals).

Linear models

There is a function in the package `lme4` that computes separate linear models for each subject: `lmList`.

```
library(lme4)

## Loading required package:  Matrix

lmlist.fm1<-lmList(logrt~so|subject,dat)
```

Linear models

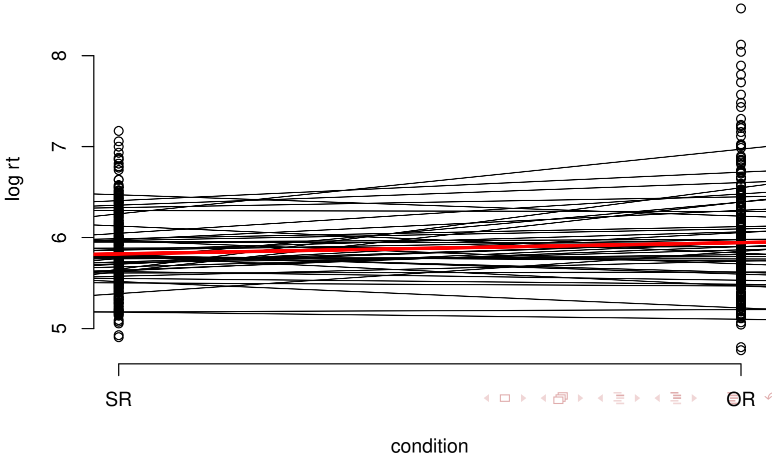
Intercept and slope estimates for three subjects:

```
lmlist.fm1$`1`$coefficients

## (Intercept)          so
##    5.769617    0.043515

lmlist.fm1$`28`$coefficients

## (Intercept)          so
##     6.10021     0.44814

lmlist.fm1$`37`$coefficients

## (Intercept)          so
##     6.61699     0.35537
```

Linear models

One can plot the individual lines for each subject, as well as the linear model m0's line (this shows how each subject deviates in intercept and slope from the model m0's intercept and slopes).

Linear models



Linear models

To find out if there is an effect of RC type, you can simply check whether the slopes of the individual subjects' fitted lines taken together are significantly different from zero.

Linear models

```
t.test(coef(lmlist.fm1)[2])

##
##  One Sample t-test
##
## data:  coef(lmlist.fm1)[2]
## t = 2.81, df = 41, p-value = 0.0076
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.017449 0.106585
## sample estimates:
## mean of x
##  0.062017
```

Linear models

The above test is exactly the same as the paired t-test and the varying intercepts linear mixed model **on aggregated data**:

```
t.test(logrt~condition,bysubj,paired=TRUE)$statistic

##      t
## 2.8102

## also compare with linear mixed model:
summary(lmer(logrt~condition+(1|subject),
             bysubj))$coefficients[2,]

##   Estimate Std. Error    t value
##  -0.124033   0.044137  -2.810207
```

Linear models

- The above lmList model we fit is called **repeated measures regression**. We now look at how to model unaggregated data using the linear mixed model.
- This model is now only of historical interest, and useful only for understanding the linear mixed model, which is the modern standard approach.

Linear mixed models

- The **linear mixed model** does something related to the above by-subject fits, but with some crucial twists, as we see below.
- In the model shown in the next slide, the statement
  (1|subject)
  adjusts the grand mean estimates of the intercept by a term (a number) for each subject.

Linear mixed models

Notice that we did not aggregate the data here.

```
m0.lmer<-lmer(logrt~so+(1|subject),dat)
```

Abbreviated output:

```
Random effects:
 Groups   Name        Variance Std.Dev.
 subject  (Intercept) 0.09983  0.3160
 Residual             0.14618  0.3823
Number of obs: 672, groups:  subject, 42

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.88306    0.05094 115.497
so           0.06202    0.01475   4.205
```

Linear mixed models

One thing to notice is that the coefficients (intercept and slope) of the fixed effects of the above model are identical to those in the linear model m0 above.

The varying intercepts for each subject can be viewed by typing:

```
ranef(m0.lmer)$subject[,1][1:10]

##  [1] -0.1039283  0.0771948 -0.2306209  0.2341978  0.0088
##  [7] -0.2055713 -0.1553708  0.0759436 -0.3643671
```

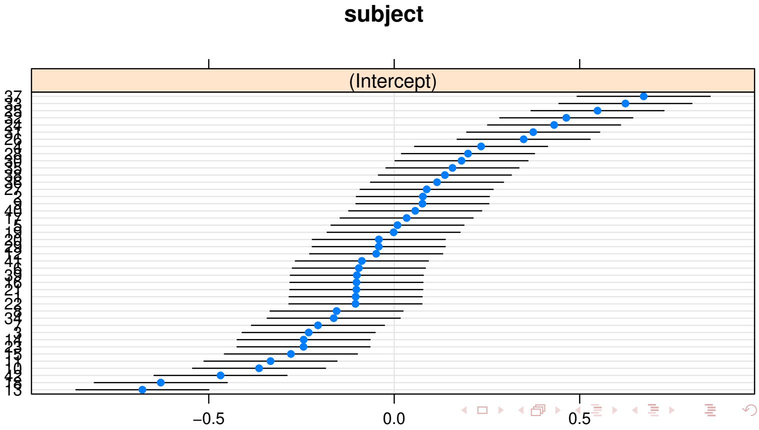
Visualizing random effects

Here is another way to summarize the adjustments to the grand mean intercept by subject. The error bars represent 95% confidence intervals.

```
library(lattice)
print(dotplot(ranef(m0.lmer,condVar=TRUE)))
```

Visualizing random effects

```
## $subject
```



Linear mixed models

The model m0.lmer above prints out the following type of linear model. i indexes subject, and j indexes items.
Once we know the subject id and the item id, we know which subject saw which condition:

```
subset(dat,subject==1 & item == 1)

##   subject item condition rawRT so  logrt
## 6       1    1    objgap   320  1 5.7683
```

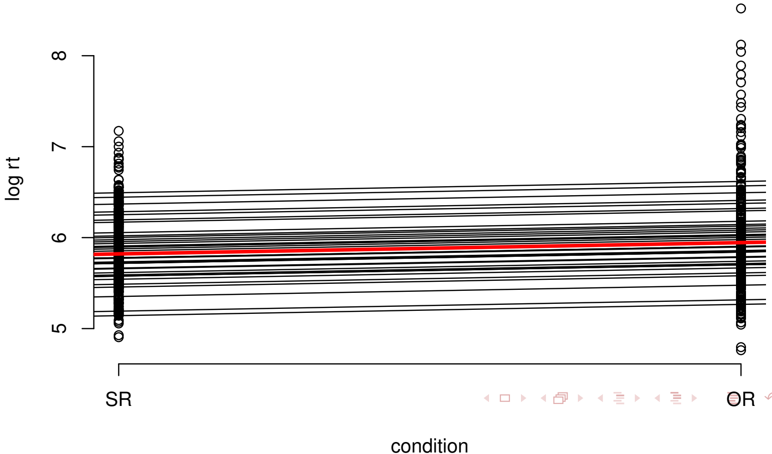
$$y_{ij} = \beta_0 + u_{0i} + \beta_1 \times so_{ij} + \epsilon_{ij} \tag{1}$$

The **only** new thing here is the by-subject adjustment to the intercept.

Linear mixed models

- Note that these by-subject adjustments to the intercept u_{0i} are assumed by lmer to come from a normal distribution centered around 0:
  $u_{0i} \sim Normal(0, \sigma_{u0})$
- The ordinary linear model m0 has one intercept β_0 for all subjects, whereas the linear mixed model with varying intercepts m0.lmer has a different intercept $(\beta_0 + u_{0i})$ for each subject i.
- We can visualize the adjustments for each subject to the intercepts as shown below.

Linear mixed models



Formal statement of varying intercepts linear mixed model

i indexes subjects, j items.

$$y_{ij} = \beta_0 + u_{0i} + (\beta_1) \times so_{ij} + \epsilon_{ij} \tag{2}$$

Variance components:

- $u_0 \sim Normal(0, \sigma_{u0})$
- $\epsilon \sim Normal(0, \sigma)$

Linear mixed models

Note that, unlike the figure associated with the lmlist.fm1 model above, which also involves fitting separate models for each subject, the model m0.lmer assumes **different intercepts** for each subject **but the same slope**.
We can have lmer fit different intercepts AND slopes for each subject.

Linear mixed models

Varying intercepts and slopes by subject

We assume now that each subject's slope is also adjusted:

$$y_{ij} = \beta_0 + u_{0i} + (\beta_1 + u_{1i}) \times so_{ij} + \epsilon_{ij} \tag{3}$$

That is, we additionally assume that $u_{1i} \sim Normal(0, \sigma_{u1})$.

```
m1.lmer<-lmer(logrt~so+(1+so||subject),dat)
```

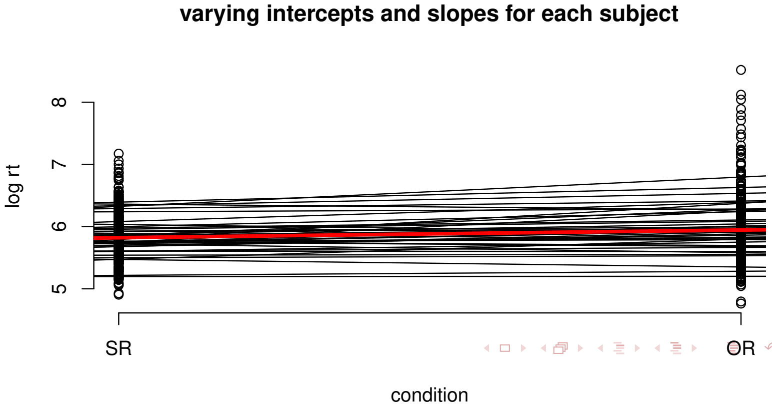
Linear mixed models

Varying intercepts and slopes by subject

```
Random effects:
 Groups    Name        Variance Std.Dev.
 subject   (Intercept) 0.1006   0.317
 subject.1 so          0.0121   0.110
 Residual              0.1336   0.365
Number of obs: 672, groups:  subject, 42

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.8831     0.0509  115.50
so            0.0620     0.0221    2.81
```

Linear mixed models

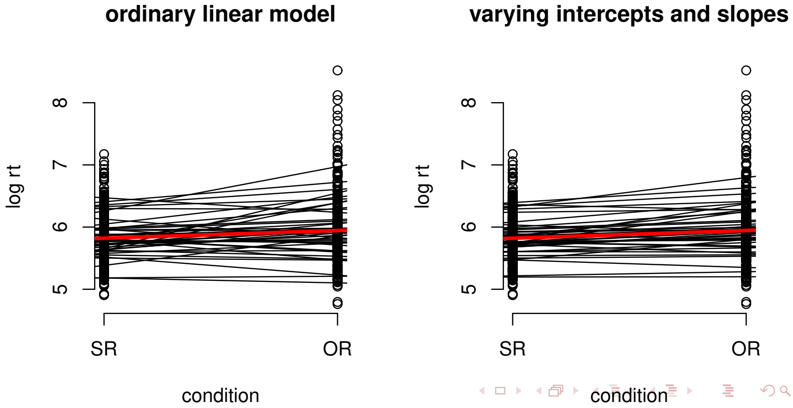
These fits for each subject are visualized below (the red line shows the model with a single intercept and slope, i.e., our old model m0):



Linear mixed models

Comparing lmList model with varying intercepts model

Compare this model with the lmlist.fm1 model we fitted earlier:

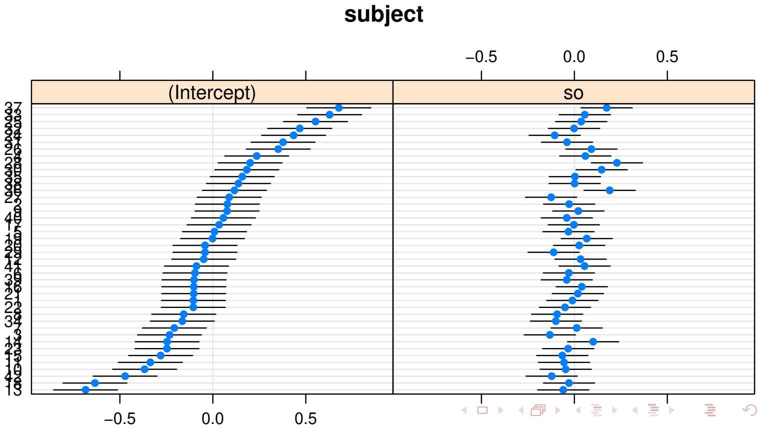


Visualizing random effects

```
print(dotplot(ranef(m1.lmer,condVar=TRUE)))
```

Visualizing random effects

```
## $subject
```



Formal statement of varying intercepts and varying slopes linear mixed model

i indexes subjects, j items.

$$y_{ij} = \beta_0 + u_{0i} + (\beta_1 + u_{1i}) \times so_{ij} + \epsilon_{ij} \tag{4}$$

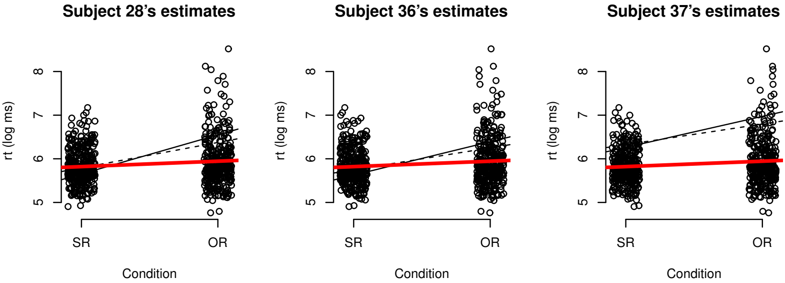
Variance components:

- $u_0 \sim Normal(0, \sigma_{u0})$
- $u_1 \sim Normal(0, \sigma_{u1})$
- $\epsilon \sim Normal(0, \sigma)$

Shrinkage in linear mixed models

- The estimate of the effect by participant is smaller than when we fit a separate linear model to the subject's data.
- This is called shrinkage in linear mixed models: the individual level estimates are shunk towards the mean slope.
- The less data we have from a given subject, the more the shrinkage.

Shrinkage in linear mixed models



Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

Let's randomly delete some data from one subject:

```
set.seed(4321)
## choose some data randomly to remove:
rand<-rbinom(1,n=16,prob=0.5)
```

Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

```
dat[which(dat$subject==37),]$rawRT

##  [1]  770  536  686  578  457  487 2419  884 3365  233
## [15] 1081  971

dat$deletedRT<-dat$rawRT
dat[which(dat$subject==37),]$deletedRT<-
  ifelse(rand,NA,
         dat[which(dat$subject==37),]$rawRT)
```

Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

Now subject 37's estimates are going to be pretty wild:

```
subset(dat,subject==37)$deletedRT

##  [1]  770   NA  686  578   NA   NA   NA   NA 3365  233
## [15]   NA  971
```

Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

```
## original no pooling estimate:
lmList.fm1_old<-lmList(log(rawRT)~so|subject,dat)
coefs_old<-coef(lmList.fm1_old)
intercepts_old<-coefs_old[1]
colnames(intercepts_old)<-"intercept"
slopes_old<-coefs_old[2]
## subject 37's original estimates:
intercepts_old$intercept[37]

## [1] 6.617

slopes_old$so[37]

## [1] 0.35537
```

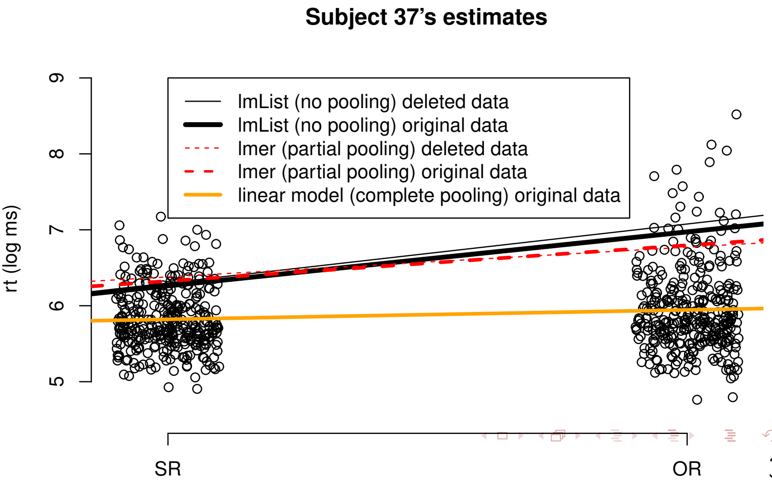
Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

```
## on deleted data:
lmList.fm1_deleted<-lmList(log(deletedRT)~so|subject,dat)
coefs<-coef(lmList.fm1_deleted)
intercepts<-coefs[1]
colnames(intercepts)<-"intercept"
slopes<-coefs[2]
## subject 37's new estimates on deleted data:
intercepts$intercept[37]

## [1] 6.6879

slopes$so[37]

## [1] 0.38843
```

Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs



Shrinkage in linear mixed models

The effect of missing data on estimation in LMMs

- What we see here is that the estimates from the hierarchical model are barely affected by the missingness, but the estimates from the no-pooling model are heavily affected.
- This means that linear mixed models will give you more robust estimates (think Type M error!) compared to no pooling models.
- This is one reason why linear mixed models are such a big deal.

Crossed subjects and items in LMMs

Subjects and items are fully crossed:

```
head(xtabs(~subject+item,dat))

##         item
## subject 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
##       1 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
##       2 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
##       3 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
##       4 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
##       5 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
##       6 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1
```

Linear mixed models

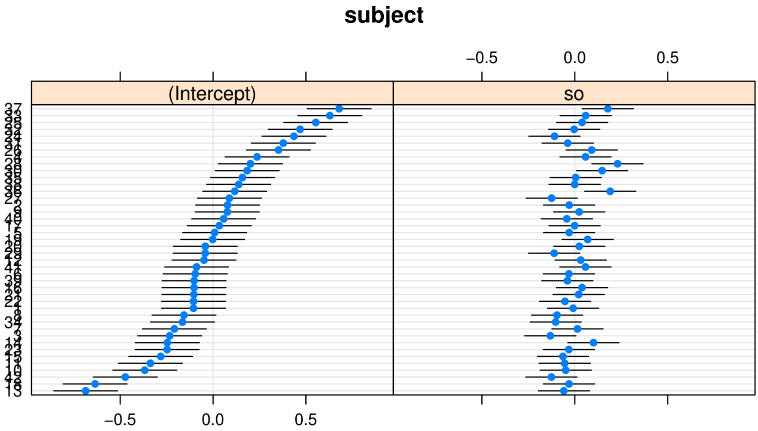
Linear mixed model with crossed subject and items random effects.

```
m2.lmer<-lmer(logrt~so+(1+so||subject)+(1+so||item),dat)
```

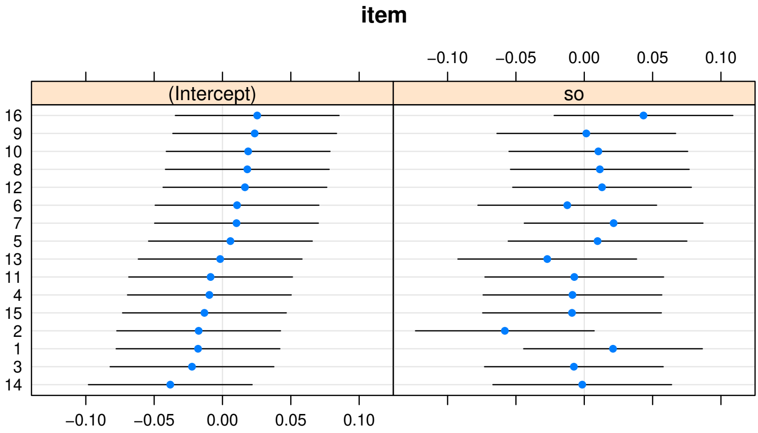
Linear mixed models

```
Random effects:
 Groups    Name        Variance Std.Dev.
 subject   (Intercept) 0.10090  0.3177
 subject.1 so          0.01224  0.1106
 item      (Intercept) 0.00127  0.0356
 item.1    so          0.00162  0.0402
 Residual              0.13063  0.3614
Number of obs: 672, groups:  subject, 42; item, 16

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.8831     0.0517  113.72
so            0.0620     0.0242    2.56
```

Visualizing random effects



Visualizing random effects



Linear mixed models

Linear mixed model with crossed subject and items random effects, with a correlation between varying intercepts and slopes.

```
m3.lmer<-lmer(logrt~so+(1+so|subject)+(1+so|item),
              dat)

## boundary (singular) fit: see ?isSingular
```

To understand what this model is doing, we have to understand what a bivariate/multivariate distribution is.

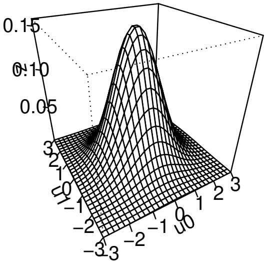
Linear mixed models

Linear mixed model with crossed subject and items random effects.

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subject  (Intercept) 0.10103  0.3178
          so          0.01228  0.1108   0.58
 item     (Intercept) 0.00172  0.0415
          so          0.00196  0.0443   1.00 <= degenerate
 Residual             0.12984  0.3603
Number of obs: 672, groups:  subject, 42; item, 16

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.8831     0.0520  113.09
so            0.0620     0.0247    2.51
```

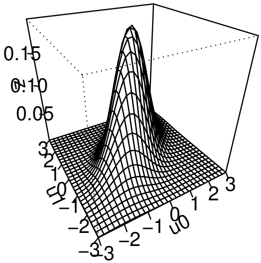
Bivariate distributions

Here are two uncorrelated normal random variables u_0 and u_1, both come from a Normal(0,1) distribution:



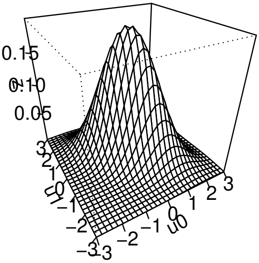
Bivariate distributions

Here is an example of positively correlated bivariate random variables:



Bivariate distributions

And here is an example with a negative correlation:



Bivariate distributions

A bivariate distribution for two random variables u_0 and u_1, each of which comes from a normal distribution, is written as follows:

$$\Sigma = \begin{pmatrix} \sigma_{u0}^2 & \rho_u \sigma_{u0} \sigma_{u1} \\ \rho_u \sigma_{u0} \sigma_{u1} & \sigma_{u1}^2 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \sim \mathcal{N}_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma \right) \tag{6}$$

Bivariate distributions

- Σ is called a variance-covariance matrix. It contains the standard deviations and correlation between the two random variables.
- In a multivariate distribution with, say, three random variables, we would have three standard deviations and two correlations, so the variance covariance matrix would be 3×3.
- Question: if we have eight correlated random variables, what are the dimensions of the variance-covariance (vcov) matrix? And how many correlation parameters will we have in this vcov matrix?

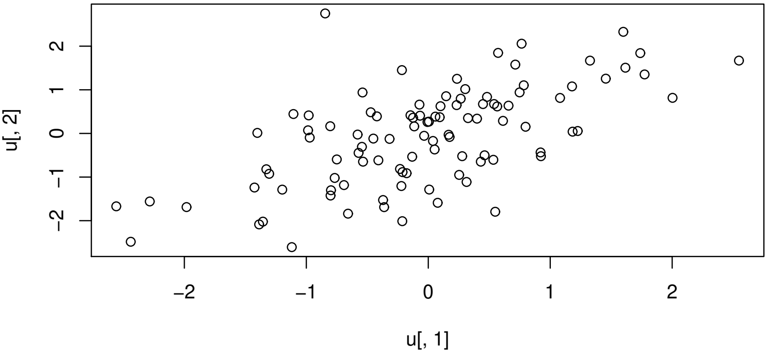
Bivariate distributions

How to generate simulated bivariate correlated data:

```
library(MASS)
Sigma<-matrix(c(1,.6,.6,1),byrow=FALSE,ncol=2)
u<-mvrnorm(100,mu=c(0,0),Sigma=Sigma)
head(u)

##          [,1]     [,2]
## [1,] -0.17762 -0.91158
## [2,]  0.27814 -0.52141
## [3,] -0.98050  0.41354
## [4,] -0.36189 -1.68936
## [5,] -0.12483  0.36014
## [6,] -0.06762  0.40630
```

Bivariate distributions

Visualizing bivariate correlated data:



Linear mixed models

The correlations (0.58 and 1.00) you see in the model output below are the correlations between the varying intercepts and slopes for subjects and for items.

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subject  (Intercept) 0.10103  0.3178
          so          0.01228  0.1108   0.58
 item     (Intercept) 0.00172  0.0415
          so          0.00196  0.0443   1.00 <= degenerate
 Residual             0.12984  0.3603
Number of obs: 672, groups:  subject, 42; item, 16

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.8831     0.0520  113.09
so            0.0620     0.0247    2.51
```

Formal statement of varying intercepts and varying slopes linear mixed model with correlation

i indexes subjects, j items.

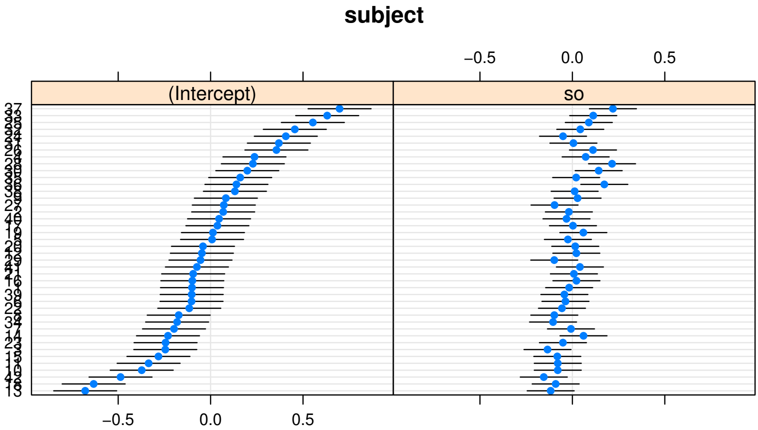
$$y_{ij} = \alpha + u_{0i} + w_{0j} + (\beta + u_{1i} + w_{1j}) * so_{ij} + \varepsilon_{ij} \tag{7}$$

where $\varepsilon_{ij} \sim Normal(0, \sigma)$ and

$$\Sigma_u = \begin{pmatrix} \sigma_{u0}^2 & \rho_u \sigma_{u0} \sigma_{u1} \\ \rho_u \sigma_{u0} \sigma_{u1} & \sigma_{u1}^2 \end{pmatrix} \quad \Sigma_w = \begin{pmatrix} \sigma_{w0}^2 & \rho_w \sigma_{w0} \sigma_{w1} \\ \rho_w \sigma_{w0} \sigma_{w1} & \sigma_{w1}^2 \end{pmatrix} \tag{8}$$

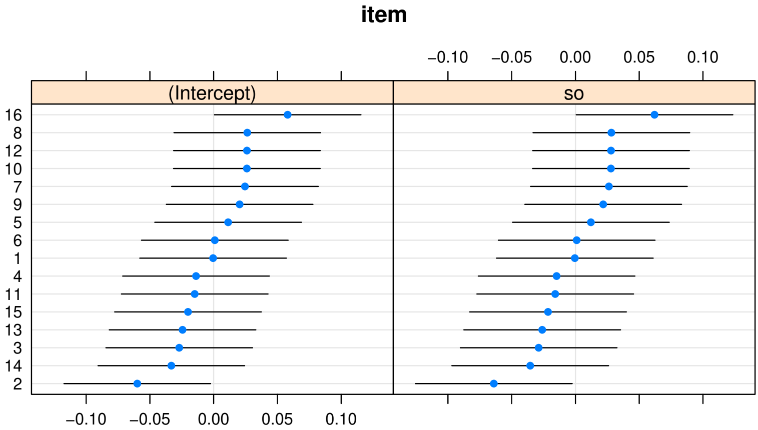
$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_u \right), \quad \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_w \right) \tag{9}$$

Visualizing random effects



Visualizing random effects

These are degenerate estimates



Model assumptions, model selection

Goals:

- learn to check for the normality of residuals
- learn to log-transform the data
- learn to compare models to decide which one to use
- learn carry out your hypothesis test using the likelihood ratio test

Model assumptions, model selection

Example: Chinese relative clause data

```
dat<-read.table("data/gibsonwu2012datarepeat.txt",
                header=TRUE)
head(dat)

##    subj item condition pos   rt    region
## 9   1m1   15   obj-ext   8  832 head noun
## 20  1m1    8  subj-ext   8 2131 head noun
## 33  1m1   11   obj-ext   8  553 head noun
## 46  1m1   10  subj-ext   8 1091 head noun
## 62  1m1   16  subj-ext   8  598 head noun
## 75  1m1   14  subj-ext   8  645 head noun

dat$cond<-ifelse(dat$condition=="subj-ext",-1,1)
```

Model assumptions, model selection

Example: Chinese relative clause data

```r
library(lme4)
m0<-lmer(rt~cond + (1|subj),dat)
```

Model assumptions, model selection

Example: Chinese relative clause data

```
summary(m0)

## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ cond + (1 | subj)
##    Data: dat
##
## REML criterion at convergence: 9423
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.765 -0.289 -0.170 -0.004 14.782
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept)  13491   116
##  Residual             445327   667
## Number of obs: 595, groups:  subj, 40
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    500.7       33.0    15.2
## cond           -54.7       27.4    -2.0
##
## Correlation of Fixed Effects:
##      (Intr)
## cond 0.004
```

Model assumptions, model selection

Example: Chinese relative clause data

```
m1<-lmer(rt~cond + (1+cond||subj),dat)

## boundary (singular) fit:  see ?isSingular
```

Model assumptions, model selection

Example: Chinese relative clause data

```
summary(m1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ cond + ((1 | subj) + (0 + cond | subj))
##    Data: dat
##
## REML criterion at convergence: 9423
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.765 -0.289 -0.170 -0.004 14.782
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept)  13490   116
##  subj.1   cond             0     0
##  Residual             445328   667
## Number of obs: 595, groups:  subj, 40
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    500.7       33.0    15.2
## cond           -54.7       27.4    -2.0
##
## Correlation of Fixed Effects:
##      (Intr)
## cond 0.004
```

Model assumptions, model selection

Example: Chinese relative clause data

```
m2<-lmer(rt~cond + (1+cond|subj),dat)

## boundary (singular) fit:  see ?isSingular
```

Model assumptions, model selection

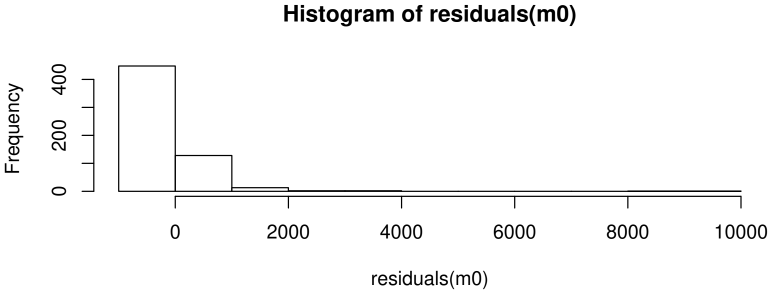
Example: Chinese relative clause data

```
summary(m2)

## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ cond + (1 + cond | subj)
##    Data: dat
##
## REML criterion at convergence: 9415.4
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.436 -0.259 -0.173  0.002 14.376
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subj     (Intercept)  19439   139.4
##           cond          9134    95.6    -1.00
##  Residual             430534   656.2
## Number of obs: 595, groups:  subj, 40
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    501.5       34.8   14.42
## cond           -54.8       30.9   -1.77
##
## Correlation of Fixed Effects:
##      (Intr)
## cond -0.307
```

Model assumptions, model selection

Example: Chinese relative clause data

Model assumption: residuals are normal



Model assumptions, model selection

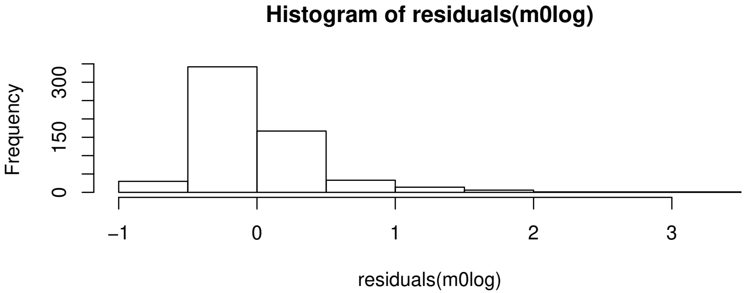
Example: Chinese relative clause data

This assumption is clearly violated. A log-transform on the reading times will reduce the skew:

```
m0log<-lmer(log(rt)~cond + (1|subj),dat)
m1log<-lmer(log(rt)~cond + (1+cond||subj),dat)
m2log<-lmer(log(rt)~cond + (1+cond|subj),dat)
```

Model assumptions, model selection

Example: Chinese relative clause data



Model assumptions, model selection

Example: Chinese relative clause data

This is good enough for now.

Model selection

Example: Chinese relative clause data

Ignoring model assumptions for a second and analyzing raw rt's, we have three models, m0, m1, m2. Which model is best? There are two schools of thought.

Model assumptions, model selection

Example: Chinese relative clause data

Barr et al 2013: Always fit the maximal model

Under this view, m2 is always the best model (as long as it converges). If it doesn't converge, then back to the most complex model (m1 or m0) that converges.

See: http://idiom.ucsd.edu/ rlevy/papers/barr-etal-2013-jml.pdf

Model selection

Example: Chinese relative clause data

The likelihood ratio test (aka analysis of variance or anova)
We compare the ratios of the likelihoods of the two models of interest (= difference in log likelihoods).
For technical reasons, we have to set a value REML to FALSE in the lmer function when doing model comparison. If you forget to do it, lmer will automatically do it for you.

Model selection

Example: Chinese relative clause data

```
m0<-lmer(rt~cond + (1|subj),dat,REML=FALSE)
logLik(m0)

## 'log Lik.' -4720.2 (df=4)

m1<-lmer(rt~cond + (1+cond||subj),dat,REML=FALSE)

## boundary (singular) fit:  see ?isSingular

logLik(m1)

## 'log Lik.' -4720.2 (df=5)
```

Model selection

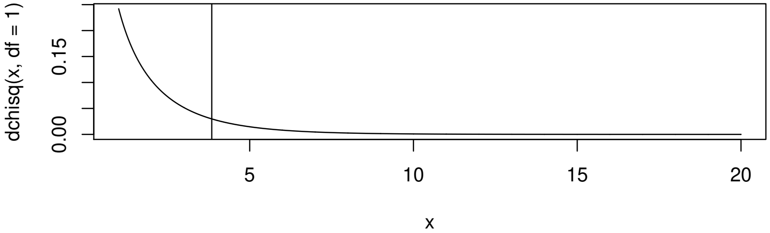
Example: Chinese relative clause data

- You can see that the ratio is 1.
- The difference of the log likelihoods follows a chi-square distribution with the parameter degrees of freedom (df) being the difference in the number of parameters in the two models being compared.
- In model m0 there are 4 parameters, and in m1 there are 5, so the difference is df=1.
- So the relevant Chi-sq distribution is chisq(df=1).

Model selection

Example: Chinese relative clause data

Let's visualize this and draw the critical chi-sq value (just like the critical t-value in the t-distribution):



Model selection

Example: Chinese relative clause data

If the difference in log likelihoods is bigger than the critical chi-squared value (for a given degree of freedom), then we reject the null hypothesis that the two models have the same log likelihoods.

If there is no evidence for one model being better, then we choose the simpler model, on grounds of parsimony (Occam's razor).

See Bates, Kliegl, Vasishth, Baayen, Parsimonious Mixed Models: https://arxiv.org/abs/1506.04967.

Model selection

Example: Chinese relative clause data

In practice, we can use the anova function (this is literally the likelihood ratio test I showed above) for model comparison:

```
anova(m0,m1)

## Data: dat
## Models:
## m0: rt ~ cond + (1 | subj)
## m1: rt ~ cond + ((1 | subj) + (0 + cond | subj))
##    Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0  4 9448 9466  -4720     9440
## m1  5 9450 9472  -4720     9440     0      1          1
```

Model selection

Example: Chinese relative clause data

```
anova(m1,m2)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m1: rt ~ cond + ((1 | subj) + (0 + cond | subj))
## m2: rt ~ cond + (1 + cond | subj)
##    Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m1  5 9450 9472  -4720     9440
## m2  6 9445 9471  -4716     9433  7.43      1     0.0064
```

Model selection

Example: Chinese relative clause data

```
anova(m0,m2)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m0: rt ~ cond + (1 | subj)
## m2: rt ~ cond + (1 + cond | subj)
##    Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0  4 9448 9466  -4720     9440
## m2  6 9445 9471  -4716     9433  7.43      2      0.024
```

Here, m2 is the best model under the likelihood ratio test.

Model selection

Example: Chinese relative clause data

On the log scale, the conclusion is very different!

```
anova(m0log,m1log)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m0log: log(rt) ~ cond + (1 | subj)
## m1log: log(rt) ~ cond + ((1 | subj) + (0 + cond | subj))
##       Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0log  4 840 857   -416      832
## m1log  5 842 863   -416      832  0.24      1       0.62
```

Model selection

Example: Chinese relative clause data

```
anova(m1log,m2log)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m1log: log(rt) ~ cond + ((1 | subj) + (0 + cond | subj))
## m2log: log(rt) ~ cond + (1 + cond | subj)
##       Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m1log  5 842 863   -416      832
## m2log  6 843 870   -416      831  0.02      1      0.88
```

Model selection

Example: Chinese relative clause data

```
anova(m0log,m2log)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m0log: log(rt) ~ cond + (1 | subj)
## m2log: log(rt) ~ cond + (1 + cond | subj)
##       Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0log  4 840 857   -416      832
## m2log  6 843 870   -416      831  0.26      2       0.88
```

m0 is good enough. Since the model assumptions are severely violated in the raw reading time analyses, I would only trust the log rt analyses (well, we didn't learn much from the expt. since we failed to reject the null).

Model selection

Example: Chinese relative clause data

Checking if a predictor is significant:
After you have decided on which model you want to choose as the final one, you can now do a significance test to test whether a predictor is statistical significant, using the likelihood ratio test.

Model selection

Example: Chinese relative clause data

Suppose we decide on the m0log model. Then, we can check if relative clauses have an effect as follows. The null hypothesis is that $\beta_1 = 0$.

```
## Null model:
m0logNULL<-lmer(log(rt)~ 1 + (1|subj),dat)
## Alternative model:
m0log<-lmer(log(rt)~1 + cond + (1|subj),dat)
```

Model selection

Example: Chinese relative clause data

```
anova(m0logNULL,m0log)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m0logNULL: log(rt) ~ 1 + (1 | subj)
## m0log: log(rt) ~ 1 + cond + (1 | subj)
##           Df AIC BIC logLik deviance Chisq Chi Df Pr(>Ch
## m0logNULL  3 843 856   -419      837
## m0log      4 840 857   -416      832   5.3      1      (
```

Model selection

Example: Chinese relative clause data

If you had gone the Barr et al route, then you would do:

```
m2logNULL<-lmer(log(rt)~1 + (1+cond|subj),dat)
m2log<-lmer(log(rt)~1+cond + (1+cond|subj),dat)
```

Model selection

Example: Chinese relative clause data

```
anova(m2logNULL,m2log)

## refitting model(s) with ML (instead of REML)

## Data: dat
## Models:
## m2logNULL: log(rt) ~ 1 + (1 + cond | subj)
## m2log: log(rt) ~ 1 + cond + (1 + cond | subj)
##           Df AIC BIC logLik deviance Chisq Chi Df Pr(>Ch
## m2logNULL  5 846 868   -418      836
## m2log      6 843 870   -416      831  4.54      1      (
```

Model selection

Example: Chinese relative clause data

In this example, the conclusion is the same. But we will see later that the conclusion can change depending on whether you fit a maximal model or not.

Model selection

Example: Chinese relative clause data

How to report your results in a paper:
"A linear mixed model was fit with {varying intercepts/varying intercepts and slopes with no correlation/varying intercepts and slopes, the maximal model}, with SR coded as -1 and OR as +1. The dependent variable (reading time in milliseconds) was log-transformed. The results show that the object relative clause was read faster than the subject relative clause $\chi^2_1 = 4.54, p = 0.033$."