Introduction to statistics: Linear mixed models

Shravan Vasishth

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

September 19, 2021

Linear models

Example: SR/OR relative clause data from English (Grodner and Gibson 2005, Expt 1).

```
library(lingpsych)
data("df_gg05e1")
head(df_gg05e1)

##    subject item condition rawRT
## 6        1    1    objgap   320
## 19       1    2   subjgap   424
## 34       1    3    objgap   309
## 49       1    4   subjgap   274
## 68       1    5    objgap   333
## 80       1    6   subjgap   266
```

Linear models

```
dat<- df_gg05e1
dat$logrt<-log(dat$rawRT)
dat$so<-ifelse(dat$condition=="objgap",1,-1)

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

Linear models

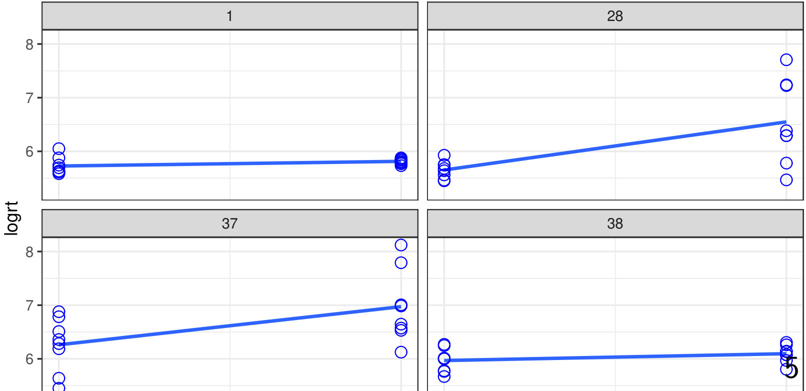
The simple linear model (incorrect for these data):

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

##               Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) 5.88305598 0.01905233 308.784102 0.00000000
## so          0.06201673 0.01905233   3.255074 0.00119067
```

Linear models

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

```
## `geom_smooth()` using formula 'y ~ x'
```



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

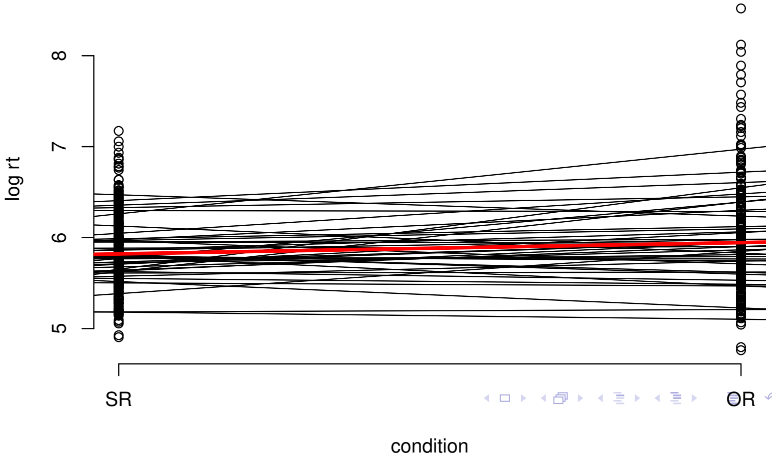
Intercept and slope estimates for three subjects:

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

## (Intercept)          so
##  5.76961670  0.04351522

lmlist.fm1$`28`$coefficients

## (Intercept)          so
##   6.1002087   0.4481361

lmlist.fm1$`37`$coefficients

## (Intercept)          so
##   6.6169864   0.3553716
```

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 = 2.8102, df = 41, p-value = 0.007556
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.01744873 0.10658474
## sample estimates:
##  mean of x
## 0.06201673
```

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
## 2.810207

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

##    Estimate  Std. Error     t value
## -0.12403347  0.04413677 -2.81020732
```

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] -0.103928344  0.077194784 -0.230620884  0.23419780
##  [6] -0.095363336 -0.205571345 -0.155370800  0.07594364
```

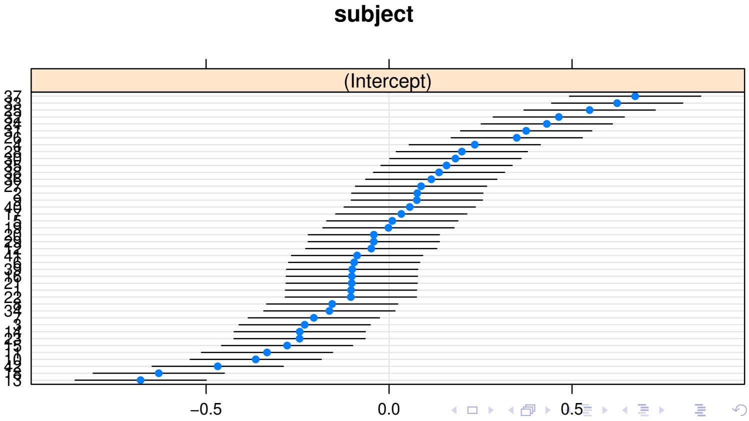
Visualizing random effects

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    logrt so
## 6       1    1    objgap   320 5.768321  1
```

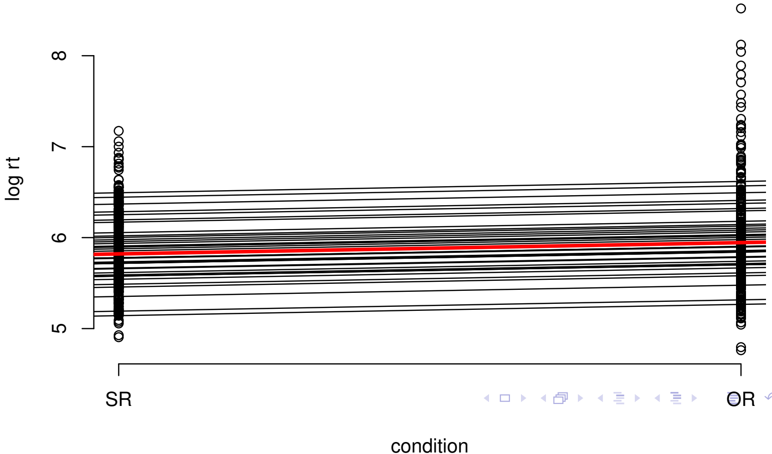
$$y_{ij} = \beta_0 + u_{0i} + \beta_1 \times so_{ij} + \epsilon_{ij} \tag{1}$$

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)
```

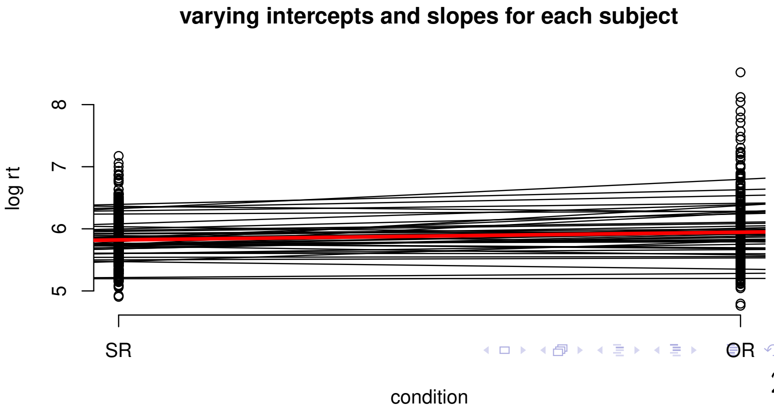
Linear mixed models

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

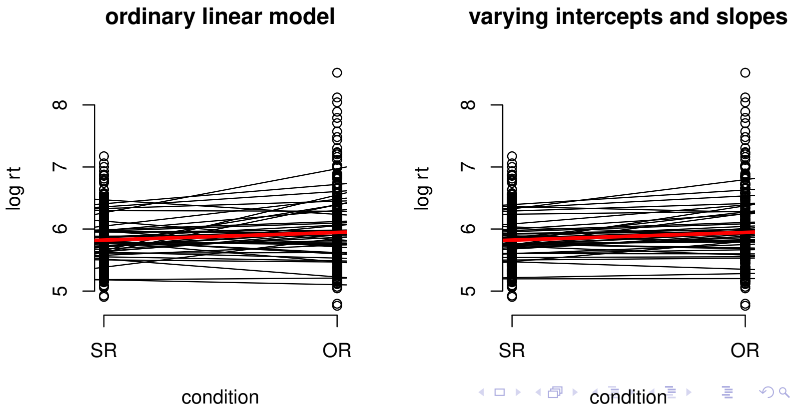
These fits for each subject are visualized below (the red line shows the model with a single intercept and slope, i.e., our old model m0):



Linear mixed models

Comparing lmList model with varying intercepts model

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

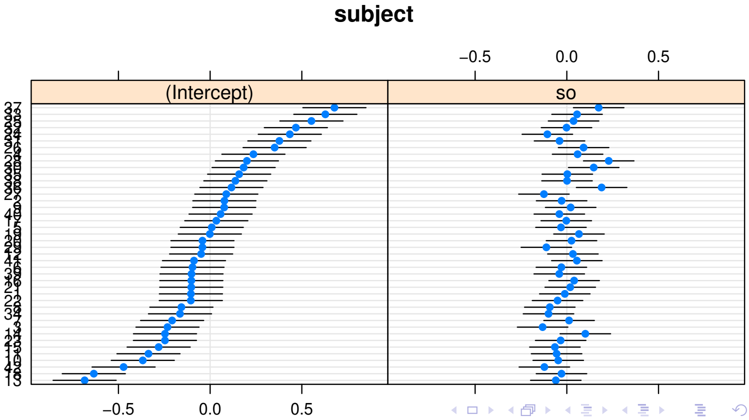


Visualizing random effects

```
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.

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

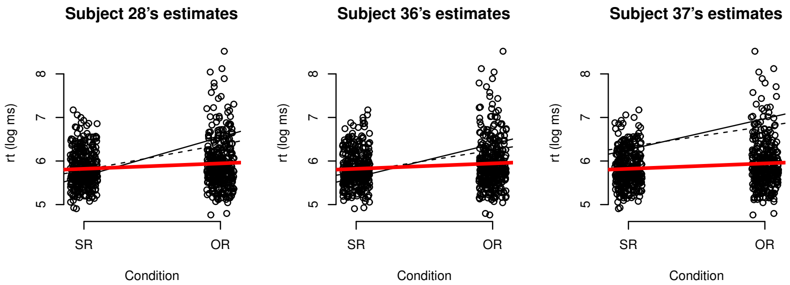
Variance components:

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

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] 6.616986

slopes_old$so[37]

## [1] 0.3553716
```

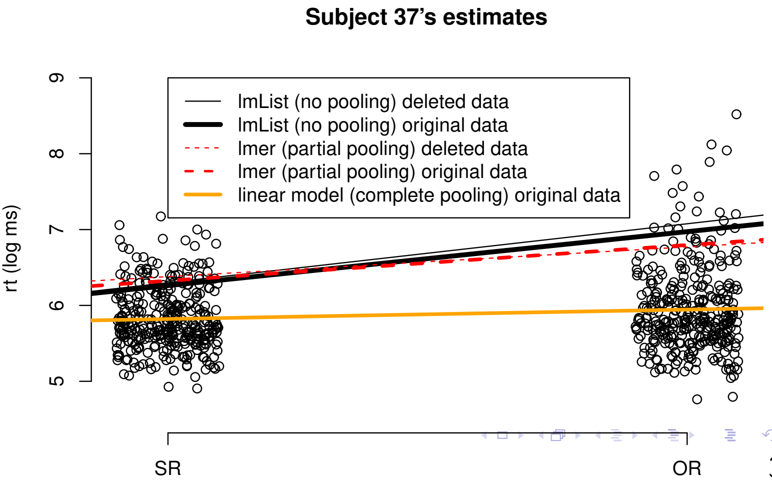
Shrinkage in linear mixed models

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] 6.687857

slopes$so[37]

## [1] 0.3884298
```

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

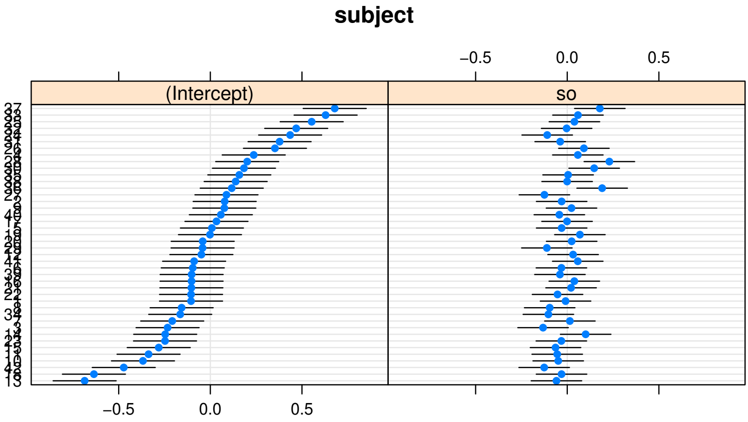
Linear mixed model with crossed subject and items random effects.

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

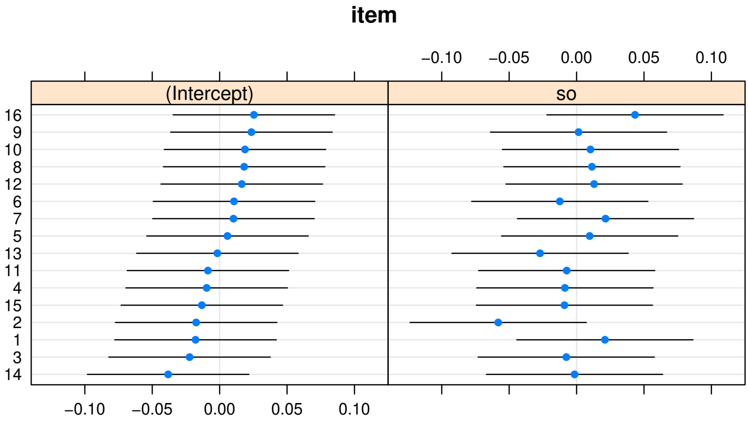
Linear mixed models

```
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.

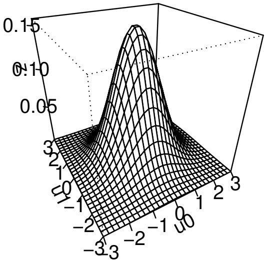
Linear mixed models

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
```

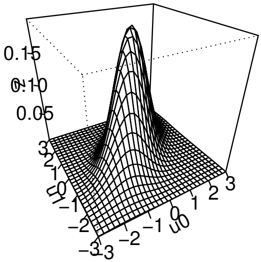
Bivariate distributions

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



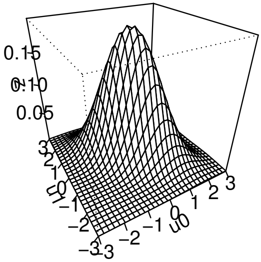
Bivariate distributions

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

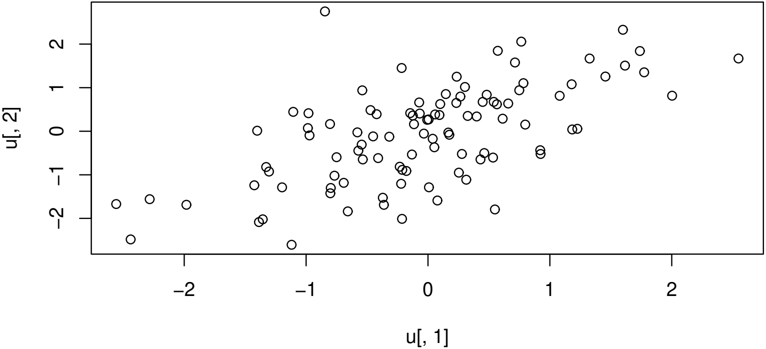
How to generate simulated bivariate correlated data:

```
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,] -0.17762140 -0.9115780
## [2,]  0.27814228 -0.5214098
## [3,] -0.98049868  0.4135393
## [4,] -0.36188591 -1.6893614
## [5,] -0.12482941  0.3601409
## [6,] -0.06761991  0.4062991
```

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.

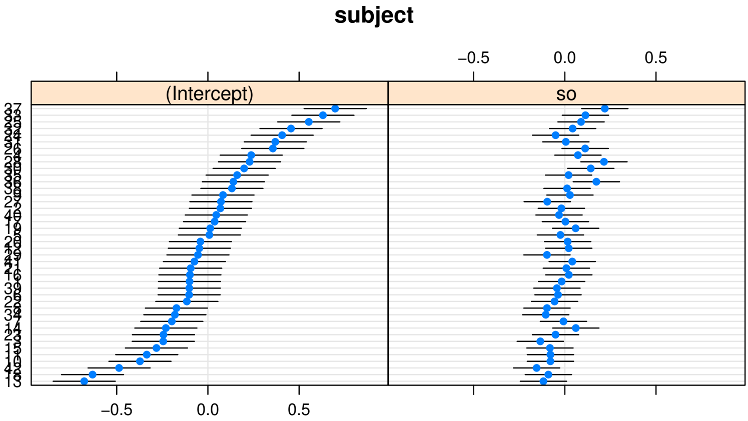
$$y_{ij} = \alpha + u_{0i} + w_{0j} + (\beta + u_{1i} + w_{1j}) * so_{ij} + \varepsilon_{ij} \tag{7}$$

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

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

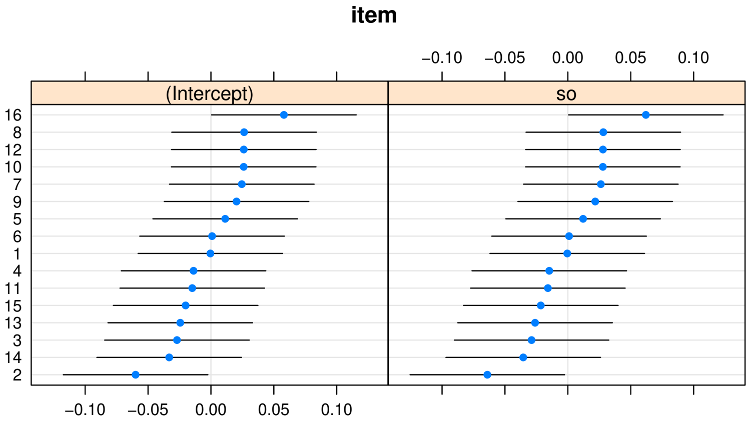
$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_u \right), \quad \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_w \right) \tag{9}$$

Visualizing random effects



Visualizing random effects

These are degenerate estimates



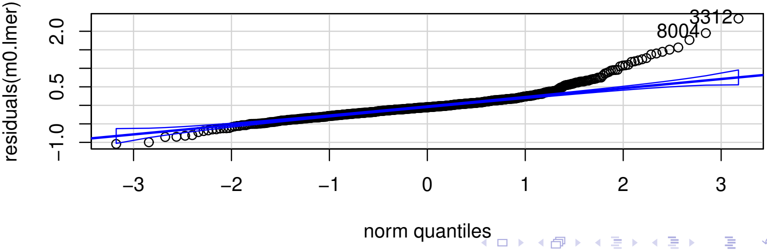
Model assumptions, model selection

Goals:

- learn to check for the normality of residuals
- find the appropriate transform for 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

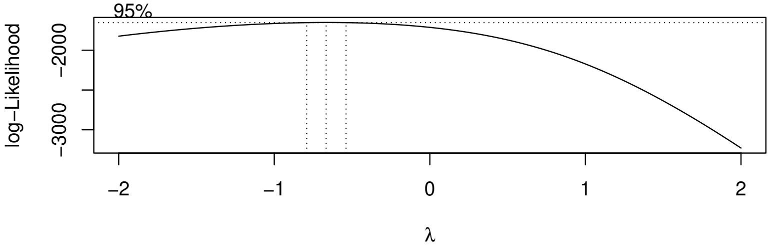
```
car::qqPlot(residuals(m0.lmer))

## 3312 8004
##  216  521
```



Model assumptions, model selection

Box-Cox transform

```
MASS::boxcox(lm(rawRT~so,dat))
```



Model assumptions, model selection

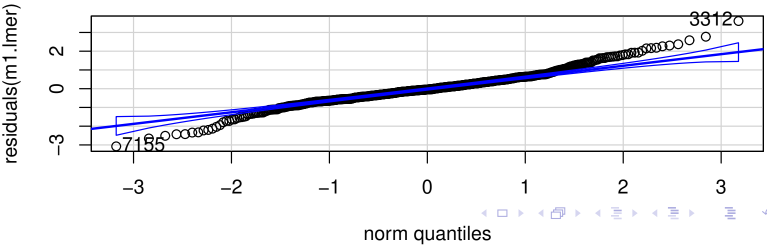
Box-Cox transform

```
m1.lmer<-lmer(I(-1000/rawRT)~so + (1+so|subject)+(1+so||ite
summary(m1.lmer)$coefficients

##               Estimate Std. Error    t value
## (Intercept) -3.0914933  0.1489319 -20.757762
## so           0.1063228  0.0611100   1.739859
```

Model assumptions, model selection

Box-Cox transform

```
car::qqPlot(residuals(m1.lmer))

## 3312 7155
##  216  466
```



Model assumptions, model selection

Use likelihood ratio test to choose the most parsimonious model

Example:

```
mNULL<-lmer(logrt~so + (1+so||subject)+(1+so||item),dat)
m<-lmer(logrt~so + (1+so|subject)+(1+so||item),dat)
anova(mNULL,m)

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

## Data: dat
## Models:
## mNULL: logrt ~ so + ((1 | subject) + (0 + so | subject)
## m: logrt ~ so + (1 + so | subject) + ((1 | item) + (0 +
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(
## mNULL     7 709.65 741.23 -347.83   695.65
## m         8 702.96 739.04 -343.48   686.96 8.6953
```

Model assumptions, model selection

Additionally, simulate data to check parameter recovery

See my textbook draft for code and discussion.

Model assumptions, model selection

NHST for the fixed effect

```
mNULL<-lmer(logrt~1+so + (1+so|subject)+(1+so||item),dat)
m<-lmer(logrt~1 + (1+so|subject)+(1+so||item),dat)
anova(mNULL,m)

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

## Data: dat
## Models:
## m: logrt ~ 1 + (1 + so | subject) + ((1 | item) + (0 + so | i
## mNULL: logrt ~ 1 + so + (1 + so | subject) + ((1 | item) + (0
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq
## m         7 707.10 738.68 -346.55   693.10
## mNULL     8 702.96 739.04 -343.48   686.96 6.1466  1    0.0131
```

Model assumptions, model selection

NHST for the fixed effect

```
mNULL<-lmer(I(-1000/rawRT)~1+so + (1+so|subject)+(1+so||item),da
m<-lmer(I(-1000/rawRT)~1 + (1+so|subject)+(1+so||item),dat)
anova(mNULL,m)

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

## Data: dat
## Models:
## m: I(-1000/rawRT) ~ 1 + (1 + so | subject) + ((1 | item) + (0
## mNULL: I(-1000/rawRT) ~ 1 + so + (1 + so | subject) + ((1 | i
##       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq
## m        7 1854.8 1886.3 -920.38   1840.8
## mNULL    8 1853.8 1889.9 -918.89   1837.8 2.9843  1    0.0840
```

For more details

https://vasishth.github.io/Freq_CogSci/