Hierarchical Linear Models: Additional notes on shrinkage and partial pooling

Shravan Vasishth

5 Dec 2018

The main point of this note

Recall that the posterior mean is a compromise between the prior mean m and the sample mean $\bar{x}$, weighted by the precision of the prior and the data.

$$m\frac{w_1}{w_1+w_2} + \bar{x}\frac{w_2}{w_1+w_2} \tag{1}$$

Here:

- w_1 is the precision (1/variance) of the prior distribution
- w_2 is the precision of the data

This fact has a practical consequence for us.

The simple varying intercepts model we are considering in the lecture notes (Example 1 in the lecture notes on hierarchical models) is:

$y_{jk} \sim Normal(\beta_0 + \beta_1 \times so_{jk} + u_{0j}, \sigma)$

where j indexes subject id's, k indexes item, and $u_{0j} \sim Normal(0, \sigma_{u0})$. The predictor so is coded as -1/+1, as in the lecture notes.

In the Bayesian framework, each of the u_{0j} is a parameter. When there is too little data to estimate a subject j's parameter, then the posterior is determined by the prior $Normal(0, \sigma_{u0})$ and the parameter u_{0j} shrinks to 0. When there is a lot of data from subject j, then the parameter u_{0j} gets closer and closer to the maximum likelihood estimate for that subject. This phenomenon is called shrinkage or partial pooling. We illustrate this next.

First we load the data from our running example, the Grodner and Gibson experiment 1 data.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
gg05e1 <- read.table("data/GrodnerGibson2005E1.csv",sep=",", header=TRUE)
gge1 <- gg05e1 %>% filter(item != 0)

gge1 <- gge1 %>% mutate(word_positionnew = ifelse(item != 15 & word_position > 10,
                                                  word_position-1, word_position))
#there is a mistake in the coding of word position,
#all items but 15 have regions 10 and higher coded
#as words 11 and higher

## get data from relative clause verb:
gge1crit <- subset(gge1, ( condition == "objgap" & word_position == 6 ) |
            ( condition == "subjgap" & word_position == 4 ))
gge1crit<-gge1crit[,c(1,2,3,6)]
gge1crit$so<-ifelse(gge1crit$condition=="objgap",1,-1)

dat<- gge1crit
```

Visualizing the data by subject

The figure below shows that participants don't all show the same difference between object and subject relatives.

```
library(ggplot2)

xlab <- "Condition (SR=-1 or OR=1)"
ylab <- "Reading time (log ms)"

ggplot(dat) +
  aes(x = so, y = log(rawRT)) +
  stat_smooth(method = "lm", se = FALSE) +
  geom_point(shape=1,position="jitter") +
  facet_wrap("subject") +
  labs(x = xlab, y = ylab) + scale_x_continuous(breaks = c(-1,1))
```

```
ggplot(dat) +
  aes(x = so, y = log(rawRT)) +
  stat_smooth(method = "lm", se = FALSE) +
  geom_point(position="jitter") +
  facet_wrap("item") +
  labs(x = xlab, y = ylab) + scale_x_continuous(breaks = c(-1,1))
```

Complete pooling model

We could fit a single linear model for all subjects (this was a homework assignment):

```
m<-lm(log(rawRT)~so,dat)

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
abline(m,lwd=3,col="red")
```

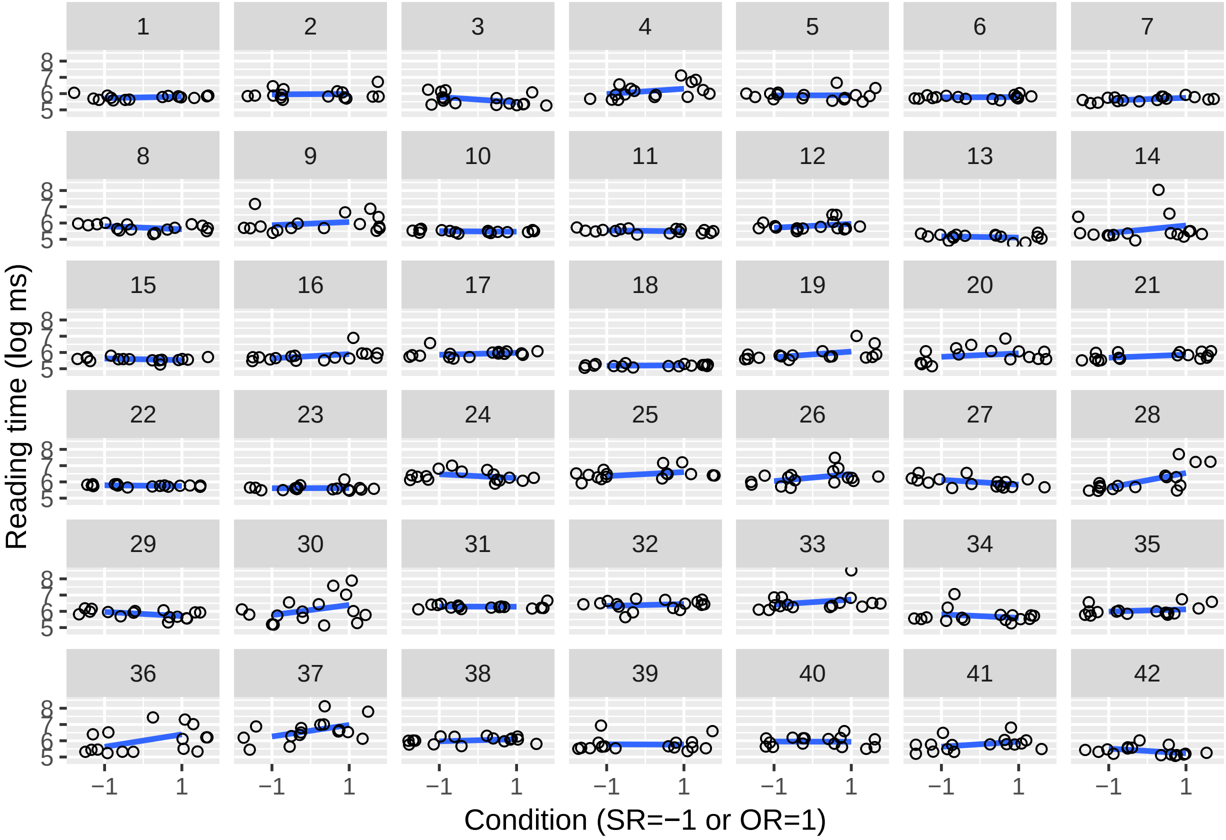


Figure 1: Relative clause effect by subject.

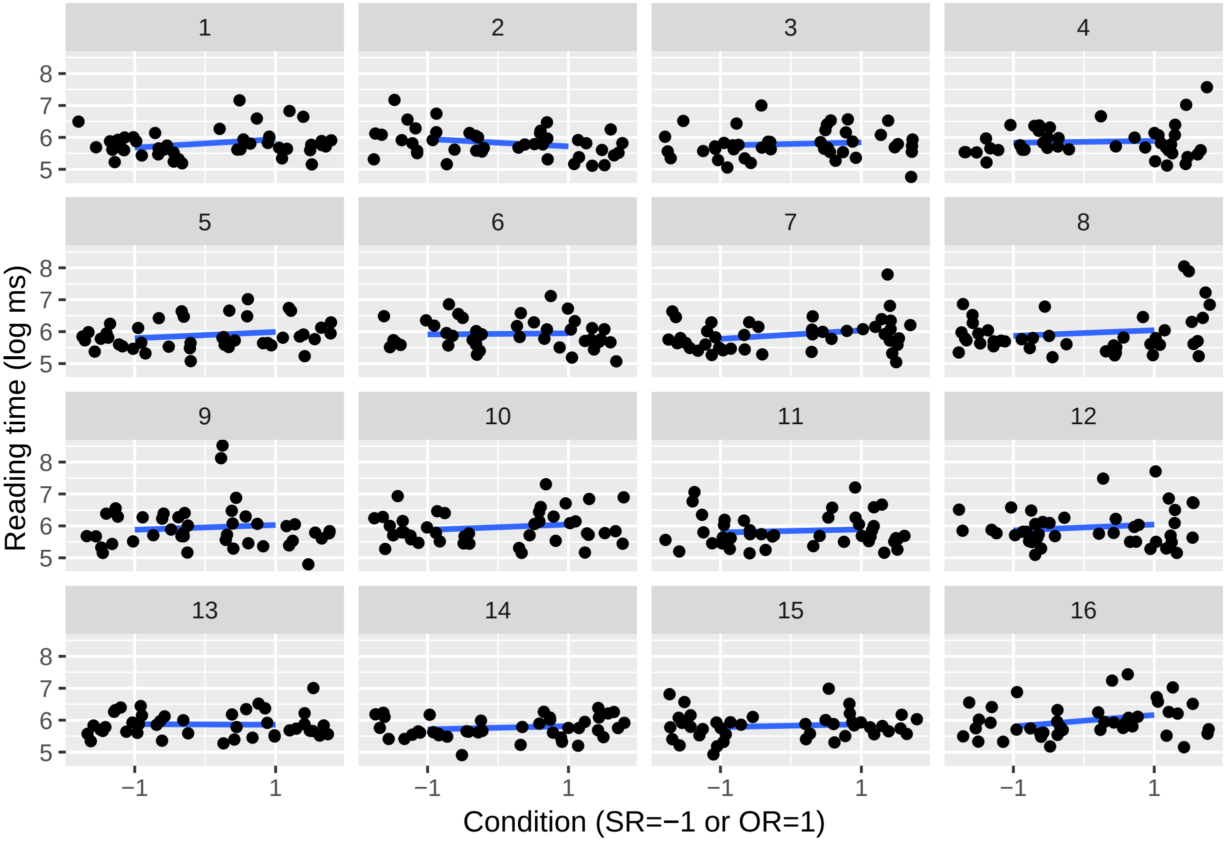


Figure 2: Relative clause effect by item.

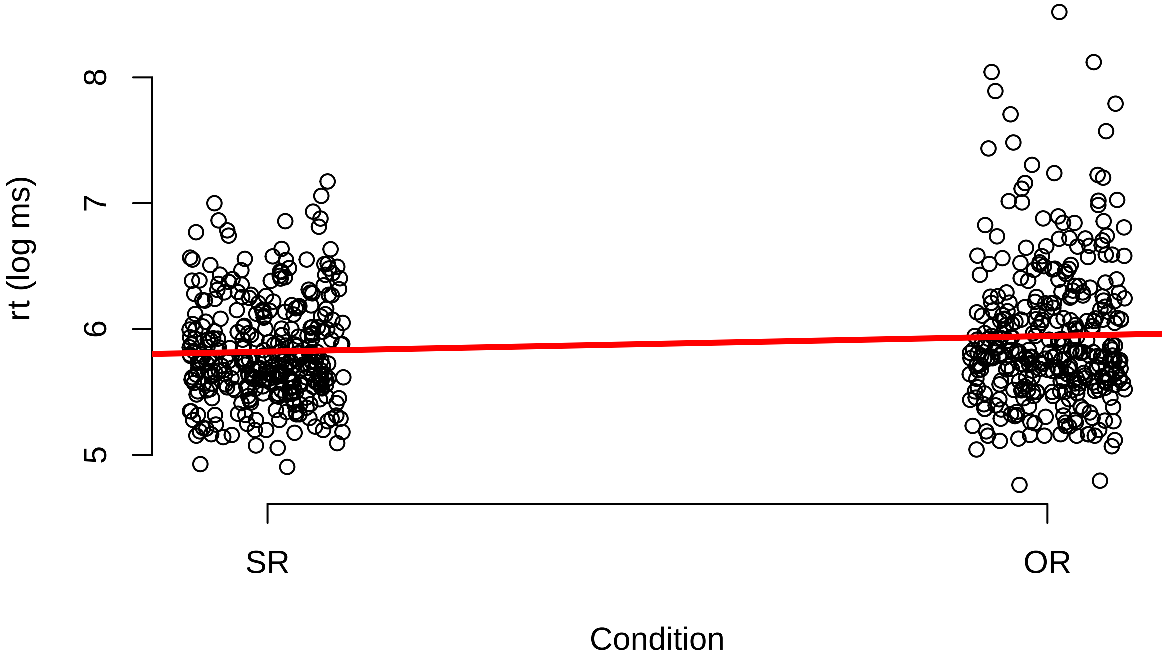


Figure 3: The complete pooling model (the simple linear model).

The no-pooling model

We could fit a **separate** linear model for each subject:

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.5.2
```

```
## Loading required package: Matrix
```

```
(nopoolingLM<-lmList(log(rawRT)~so|subject,dat))
```

```
## Call: lmList(formula = log(rawRT) ~ so | subject, data = dat)
## Coefficients:
##    (Intercept)           so
## 1     5.769617  0.043515218
## 2     5.967315  0.014033234
## 3     5.631330 -0.161005567
## 4     6.138686  0.161366792
## 5     5.892692  0.008069146
## 6     5.778966  0.011968178
## 7     5.658672  0.082841230
## 8     5.713467 -0.094740521
## 9     5.965950  0.097541658
## 10    5.485344 -0.017123723
## 11    5.519085 -0.032127097
## 12    5.829955  0.118084514
## 13    5.142064 -0.039498786
## 14    5.616587  0.232435767
## 15    5.579065 -0.047761178
## 16    5.771999  0.129391167
## 17    5.919563  0.056999301
## 18    5.196127  0.012958781
## 19    5.881306  0.173980181
## 20    5.837775  0.105200477
## 21    5.771624  0.093705895
## 22    5.769037 -0.023352276
## 23    5.616130  0.005248235
## 24    6.353523 -0.117294454
## 25    6.481291  0.124159900
## 26    6.263874  0.216525046
## 27    5.978521 -0.148962243
## 28    6.100209  0.448136057
## 29    5.837519 -0.126574081
## 30    6.081042  0.309882545
## 31    6.291959 -0.005109423
## 32    6.389314  0.059237674
## 33    6.563372  0.156377558
## 34    5.705108 -0.106648519
## 35    6.054440  0.066622990
## 36    6.009012  0.382529713
## 37    6.616986  0.355371621
## 38    6.032134  0.063833420
## 39    5.773078 -0.007693597
## 40    5.944800 -0.006987837
## 41    5.787777  0.154456355
```

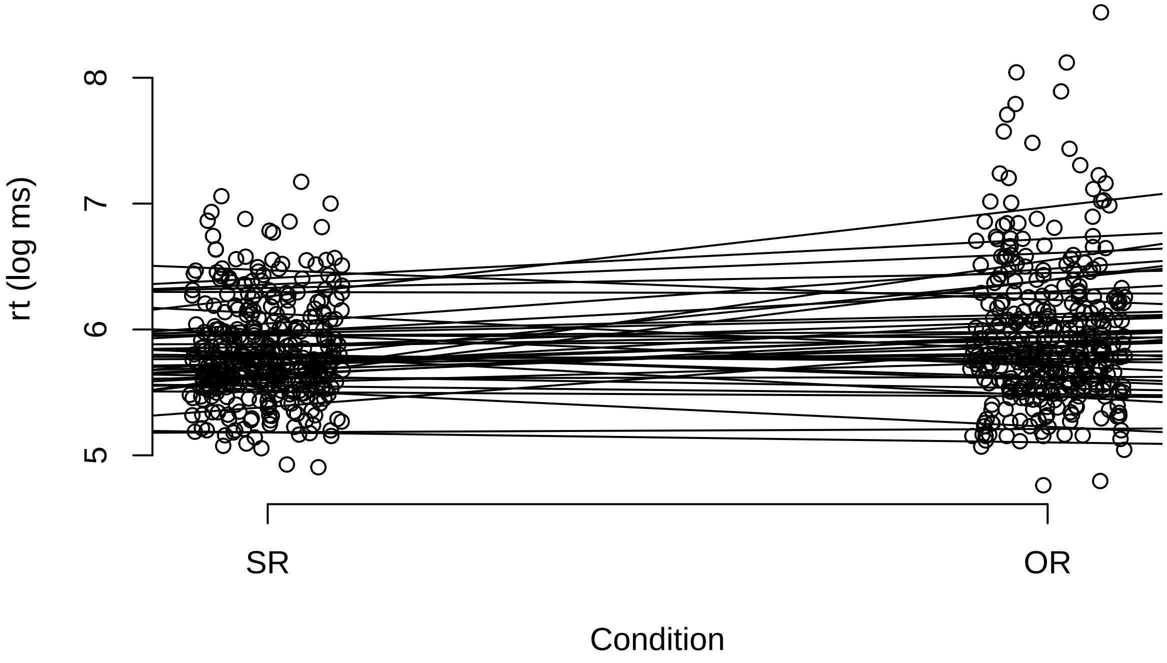


Figure 4: No pooling model's estimates for each subject (also shown is the complete pooling estimate, in red).

```
## 42    5.372041 -0.144890490
##
## Degrees of freedom: 672 total; 588 residual
## Residual standard error: 0.3654506
```

Compare the no-pooling estimates of the intercepts and slopes with the complete pooling estimates:

```
## extract intercepts and slopes by subject:
coefs<-coef(nopoolingLM)
intercepts<-coefs[1]
colnames(intercepts)<-"intercept"
slopes<-coefs[2]

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
subjects<-1:42
for(i in subjects){
   abline(intercepts$intercept[i],slopes$so[i])
   }
abline(lm(rawRT~so,dat),lwd=3,col="red")
```

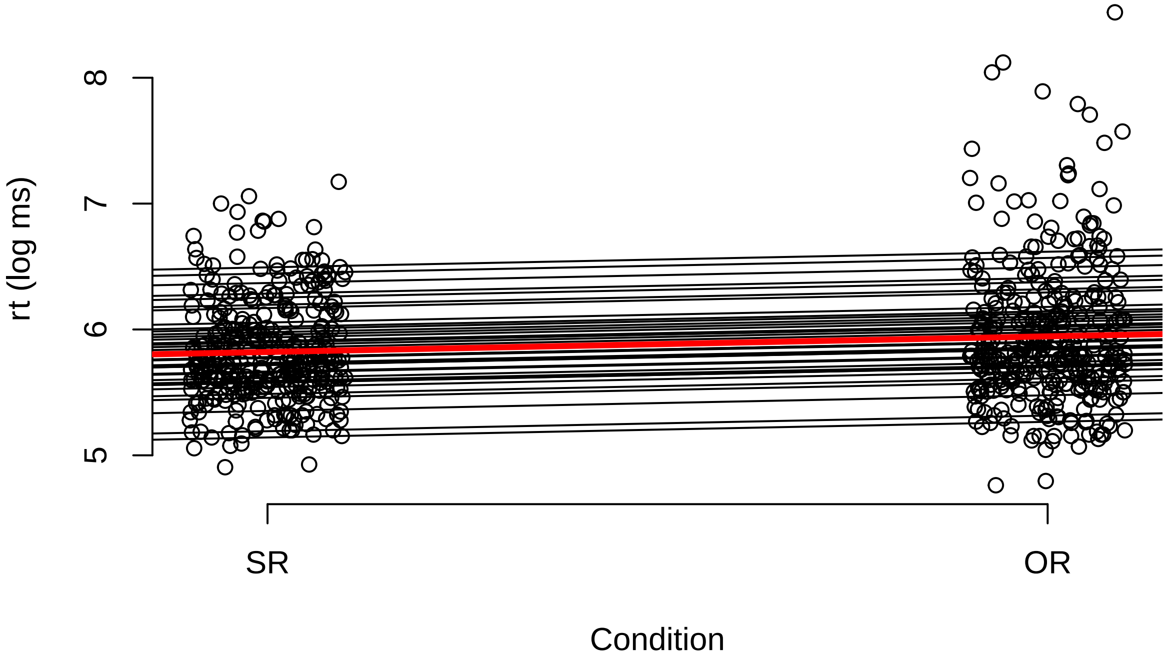


Figure 5: Partial pooling by subject (also shown is the complete pooling estimate, in red).

Hierarchical models: partial pooling

Varying intercepts by subject

First, consider the varying intercepts model. We will now visualize this model and compare it graphically to the no pooling and complete pooling models.

```
m1<-lmer(log(rawRT)~so+(1|subject),dat)
## estimates of intercept and slope
b0<-fixef(m1)[1]
b1<-fixef(m1)[2]

## intercept adjustments by subject
u0<-ranef(m1)
plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
subjects<-1:42
for(i in subjects){
   abline(b0+u0$subject[i,],b1)
   }
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")
```

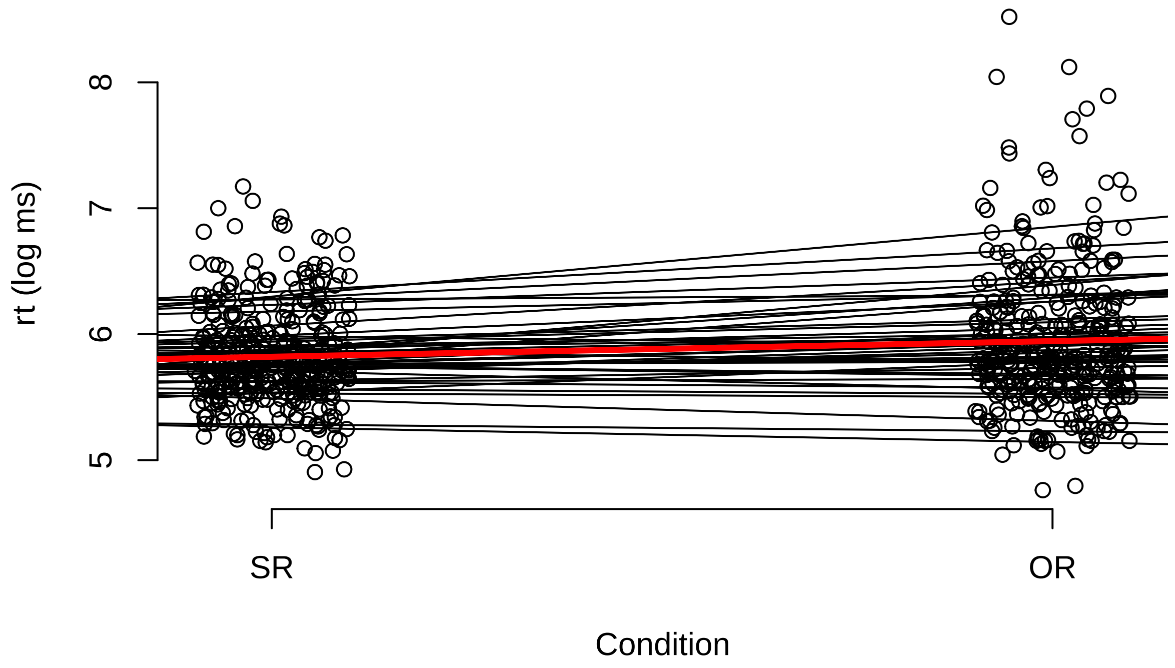


Figure 6: The partial pooling model with adjustments to intercepts and slopes by subject.

Varying intercepts and slopes by subject (the maximal model)

```
m2<-lmer(log(rawRT)~so+(1+so|subject),dat)
## estimates of fixed effects intercept and slope
b0<-fixef(m2)[1]
b1<-fixef(m2)[2]

## intercept adjustments by subject
u0<-ranef(m2)$subject[,1]
## slope adjustments by subject
u1<-ranef(m2)$subject[,2]

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
for(i in subjects){
   abline(b0+u0[i],b1+u1[i])
   }
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")
```

The practical implications of shrinkage or partial pooling

Regularization: Extreme behavior of individual subjects is shrunk towards the grand mean

Compare both the no pooling and partial pooling estimates for subjects 28, 36, 37. These subjects show the most extreme effects of relative clause type:

```
coef(nopoolingLM)[2][28,]
```

```
## [1] 0.4481361
```

```
coef(nopoolingLM)[2][36,]
```

```
## [1] 0.3825297
```

```
coef(nopoolingLM)[2][37,]
```

```
## [1] 0.3553716
```

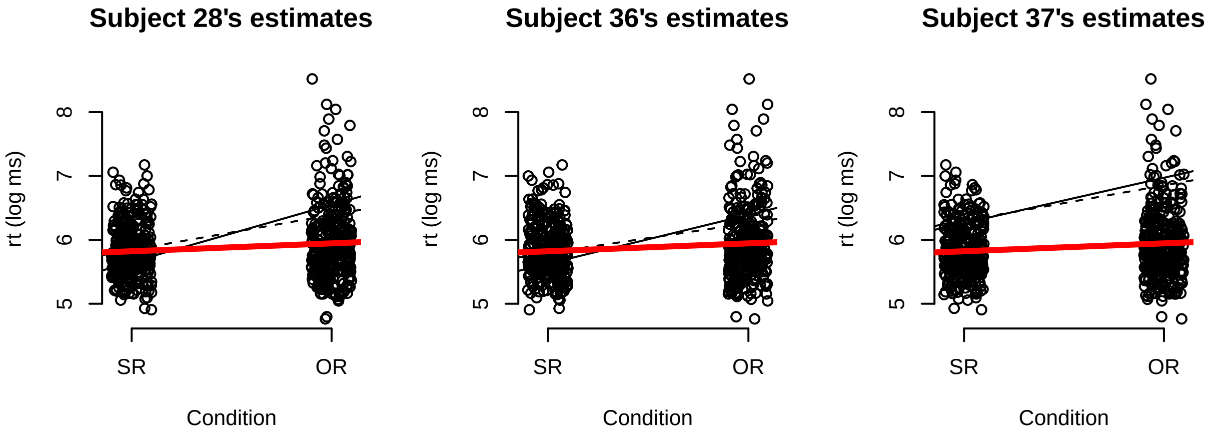
```
op<-par(mfrow=c(1,3),pty="s")
coefs<-coef(nopoolingLM)
intercepts<-coefs[1]
colnames(intercepts)<-"intercept"
slopes<-coefs[2]

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)",
        main="Subject 28's estimates")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
## no pooling estimate:
abline(intercepts$intercept[28],slopes$so[28])
## partial pooling:
abline(b0+u0[28],b1+u1[28],lty=2)
## complete pooling:
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)",
        main="Subject 36's estimates")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
## no pooling estimate:
abline(intercepts$intercept[36],slopes$so[36])
## partial pooling:
abline(b0+u0[36],b1+u1[36],lty=2)
## complete pooling:
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")

plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)",
        main="Subject 37's estimates")
```

```
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
## no pooling estimate:
abline(intercepts$intercept[37],slopes$so[37])
## partial pooling:
abline(b0+u0[37],b1+u1[37],lty=2)
## complete pooling:
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")
```

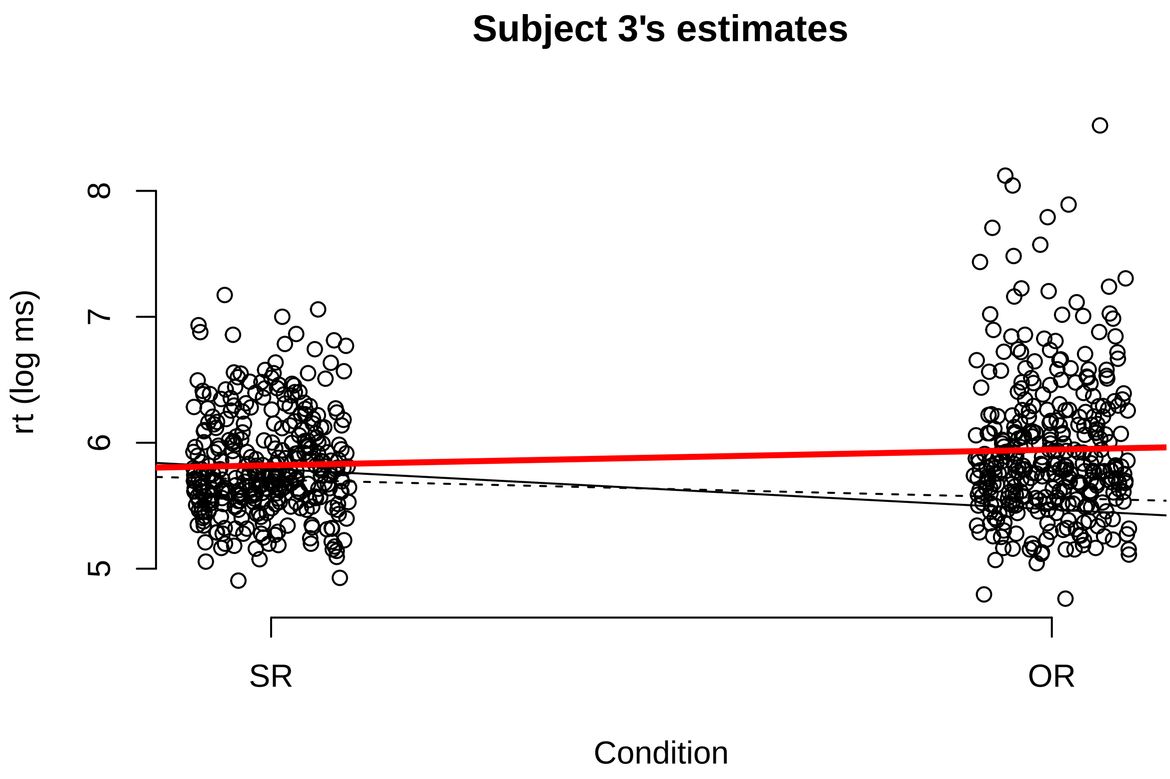

Subject 28's estimates

Subject 36's estimates

Subject 37's estimates

In the no-pooling analysis, subject 3 shows the biggest negative slope, which goes against the hypothesis that object relatives are harder to process than subject relatives. Compare the no-pooling estimate for this subject with the partial pooling or shrunk estimates from the hierarchical model:

```
plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)",
        main="Subject 3's estimates")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
## no pooling estimate:
abline(intercepts$intercept[3],slopes$so[3])
## partial pooling:
abline(b0+u0[3],b1+u1[3],lty=2)
## complete pooling:
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")
```



Regularization 2: When data from a subject are sparse, the estimates for a subject shrinks to the grand mean

First save the original estimates of the intercept and slope for subject 37:

```
intercept37orig<-intercepts$intercept[37]
slope37orig<-slopes$so[37]
```

Then delete about half the data from this subject:

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

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

```
##  [1]  770  536  686  578  457  487 2419  884 3365  233  715  671 1104  281
## [15] 1081  971
```

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

Now fit the hierarchical model and extract the estimates by subject:

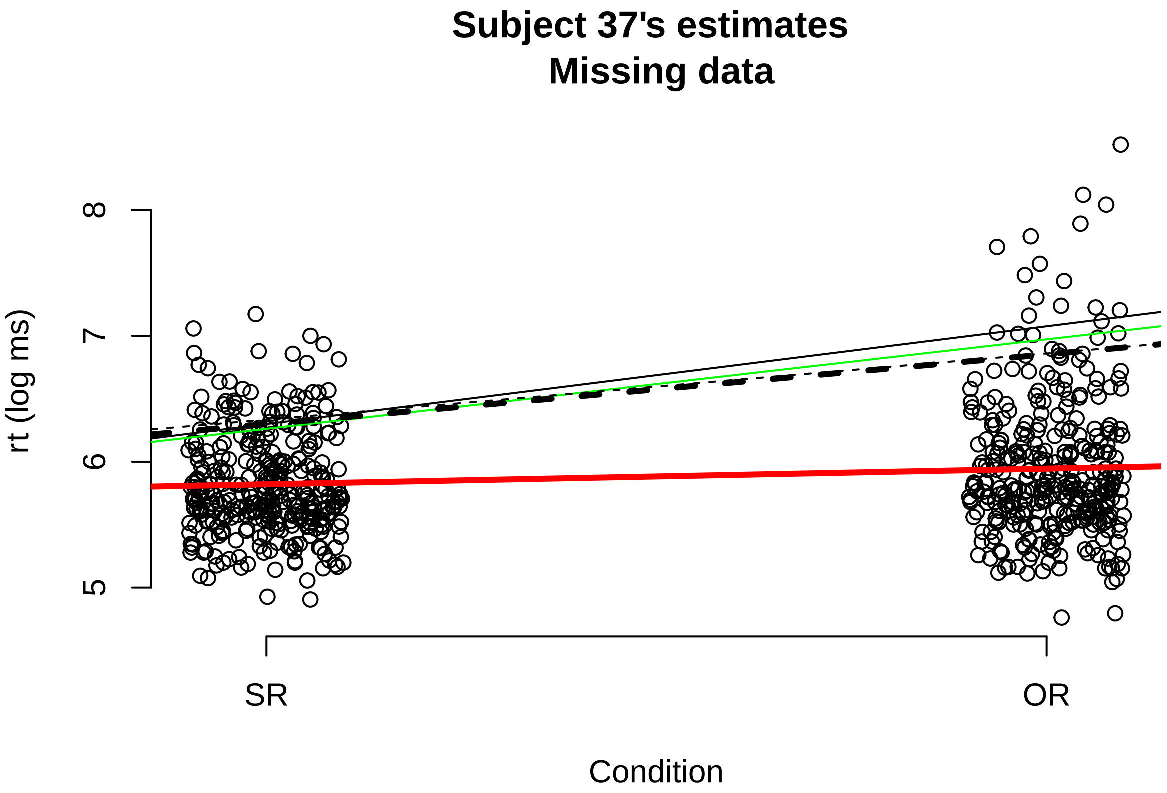
```
b0orig<-fixef(m2)[1]
b1orig<-fixef(m2)[2]
## intercept adjustments by subject
u0orig<-ranef(m2)$subject[,1]
## slope adjustments by subject
u1orig<-ranef(m2)$subject[,2]
```

```
m3<-lmer(log(deletedRT)~so+(1+so|subject),dat)

b0<-fixef(m3)[1]
b1<-fixef(m3)[2]
## intercept adjustments by subject
u0<-ranef(m3)$subject[,1]
## slope adjustments by subject
u1<-ranef(m3)$subject[,2]

## No pooling estimates for deleted data
nopoolingLMdeleted<-lmList(log(deletedRT)~so|subject,dat)
coefs<-coef(nopoolingLMdeleted)
intercepts<-coefs[1]
colnames(intercepts)<-"intercept"
slopes<-coefs[2]


plot(jitter(dat$so,.5),
        log(dat$rawRT),axes=FALSE,
        xlab="Condition",ylab="rt (log ms)",
        main="Subject 37's estimates \n Missing data")
axis(1,at=c(-1,1),labels=c("SR","OR"))
axis(2)
## no pooling estimate from deleted data:
abline(intercepts$intercept[37],slopes$so[37])
abline(intercept37orig,slope37orig,col="green")
## partial pooling deleted data:
abline(b0+u0[37],b1+u1[37],lty=2)
## partial pooling original data:
abline(b0orig+u0orig[37],b1orig+u1orig[37],lty=2,lwd=3)
## complete pooling:
abline(lm(log(rawRT)~so,dat),lwd=3,col="red")
```



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.

Summary

- When an individual subject's data has sparse measurements, the hierarchical model "shrinks" the subject's estimates to the grand mean (complete pooling model) estimates. **This is because the subject's adjustments to intercept and slope shrink towards 0, which is the prior mean.**
- When the subject provides sufficient data, the estimates for the adjustment to the intercept and slope for that subject are closer to the individual subject's means (the sample mean for that subject).