Bayes 2 lecture (EMLAR 2022)

Shravan Vasishth

4/21/2022

Review of Bayes 1 lecture

The main points that I discussed were as follows.

Bayes' rule allows us to compute/derive the posterior distribution of a

parameter or parameters of interest:

$$f(\mu|y) \propto f(y|\mu) \times f(\mu)$$

Example: Complications in an operation

I had discussed the example informally/graphically, but here is a more formal presentation:

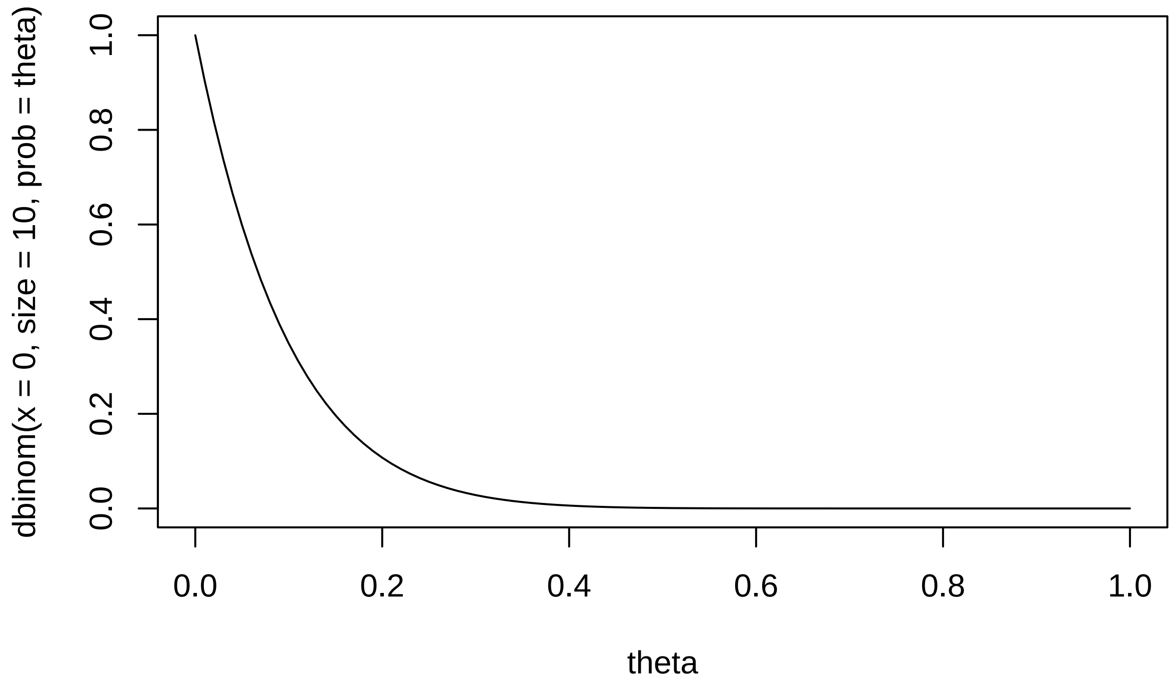
- We are modeling the number of complications that occur in operations.
- Operationalize the occurrence of a complication with 1, no complication 0.
- This suggests a binomial likelihood;
  - k is the number of successes
  - n is the total number of trials (roughly: independent data points)
  - *theta* is the probability parameter (the probability of complications)

$$Binomial(k|n,\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k} \tag{1}$$

In R, this function is the `dbinom` function; in this function, k is called x, n is called size, and prob refers to theta (because otherwise life would be too easy). Once the data are collected, the data are fixed values (constants). One can then see `dbinom` as a function of θ. That is called the likelihood function.

As an example, if we had k=0 complications (successes here), and n=10, then the likelihood function $f(n,k|\theta)$ is:

```
theta<-seq(0,1,by=0.01)
plot(theta,dbinom(x=0,size=10,prob=theta),type="l")
```



Notice that the maximum likelihood estimate is k/n, here 0/10=0. This is the estimate of θ that yields the maximum point in the likelihood function above.

Exercise: Verify graphically that if k=5, n=10, then the maximum point is at $\theta = 0.5$.

Applying Bayes' rule in our simple example

Likelihood: $\binom{n}{k}\theta^k(1-\theta)^{n-k}$

Prior on theta is Beta(3,27).

$$f(\theta|a,b) = \frac{1}{B(a=3,b=27)}\theta^{a-1}(1-\theta)^{b-1}(\#eq:beta) \tag{2}$$

The posterior (up to proportionality):

$$p(\theta|n,k) \propto \theta^k(1-\theta)^{n-k}\theta^{a-1}(1-\theta)^{b-1}(\#eq:beta) \tag{3}$$

This gives us the posterior of $\theta \sim Beta(n+a, n+b-k)$.

Insight: The posterior distribution is a compromise between the prior and the likelihood

[This is an excerpt from our book.]

Let the data be k=80, n=100. This could be a question-response accuracy for example.

Just for the sake of illustration, let's take four different beta priors, each reflecting increasing certainty.

- $Beta(a=2, b=2)$
- $Beta(a=3, b=3)$
- $Beta(a=6, b=6)$
- $Beta(a=21, b=21)$

Each prior reflects a belief that $\theta = 0.5$, with varying degrees of (un)certainty. Given the general formula we developed above for the beta-binomial case, we just need to plug in the likelihood and the prior to get the posterior:

$$p(\theta|n,k) \propto p(k|n,\theta)p(\theta) \tag{4}$$

The four corresponding posterior distributions would be:

$$p(\theta \mid k,n) \propto [\theta^{80}(1-\theta)^{20}][\theta^{2-1}(1-\theta)^{2-1}] = \theta^{82-1}(1-\theta)^{22-1} \tag{5}$$

$$p(\theta \mid k,n) \propto [\theta^{80}(1-\theta)^{20}][\theta^{3-1}(1-\theta)^{3-1}] = \theta^{83-1}(1-\theta)^{23-1} \tag{6}$$

$$p(\theta \mid k,n) \propto [\theta^{80}(1-\theta)^{20}][\theta^{6-1}(1-\theta)^{6-1}] = \theta^{86-1}(1-\theta)^{26-1} \tag{7}$$

$$p(\theta \mid k,n) \propto [\theta^{80}(1-\theta)^{20}][\theta^{21-1}(1-\theta)^{21-1}] = \theta^{101-1}(1-\theta)^{41-1} \tag{8}$$

We can visualize each of these triplets of priors, likelihoods and posteriors; see Figure @ref(fig:postbetavizvar).

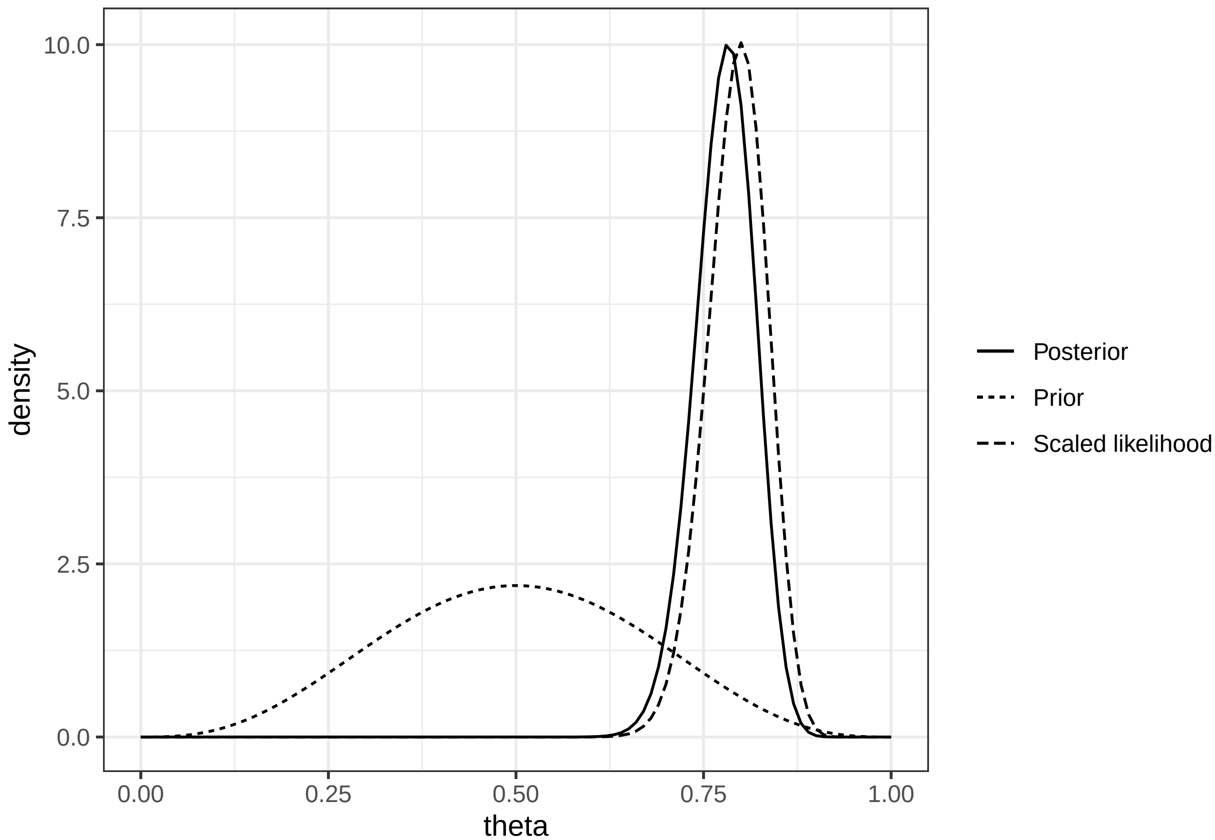


Figure 1: The (scaled) likelihood, prior, and posterior in the beta-binomial conjugate example, for different uncertainties in the prior. The likelihood is scaled to integrate to 1 to make its comparison easier.

If you hold the likelihood function constant (the data are constant at $n = 100, k = 80$ in the above example), the tighter the prior, the greater the extent to which the posterior orients itself towards the prior. In general, we can say the following about the likelihood-prior-posterior relationship:

- The posterior distribution is a compromise between the prior and the likelihood.

- For a given set of data, the greater the certainty in the prior, the more heavily the posterior will be influenced by the prior mean.
- Conversely, for a given set of data, the greater the *un*certainty in the prior, the more heavily the posterior will be influenced by the likelihood.

$ Example: Fitting a linear mixed model for a planned experiment

Read in and prepare the two data sets:

```
## load example data-set:
gw<-read.table("data/gibsonwu2012data.txt",
               header=TRUE)
## sum-contrast coding of predictor:
gw$so <- ifelse(
  gw$type%in%c("subj-ext"),-1,1)
## subset critical region
gw1<-subset(gw,region=="headnoun")

## load second data-set:
gw2<-read.table("data/gibsonwu2012datarepeat.txt",
                header=TRUE)
gw2$so <- ifelse(
  gw2$condition%in%c("subj-ext"),-1,1)
```

Frequentist analysis:

```
## frequentist analysis:
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Attaching package: 'lme4'

## The following object is masked from 'package:brms':
##
##     ngrps
```

```
m_lmer<-lmer(rt~so + (1+so|subj)+(1+so|item),gw1)
```

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

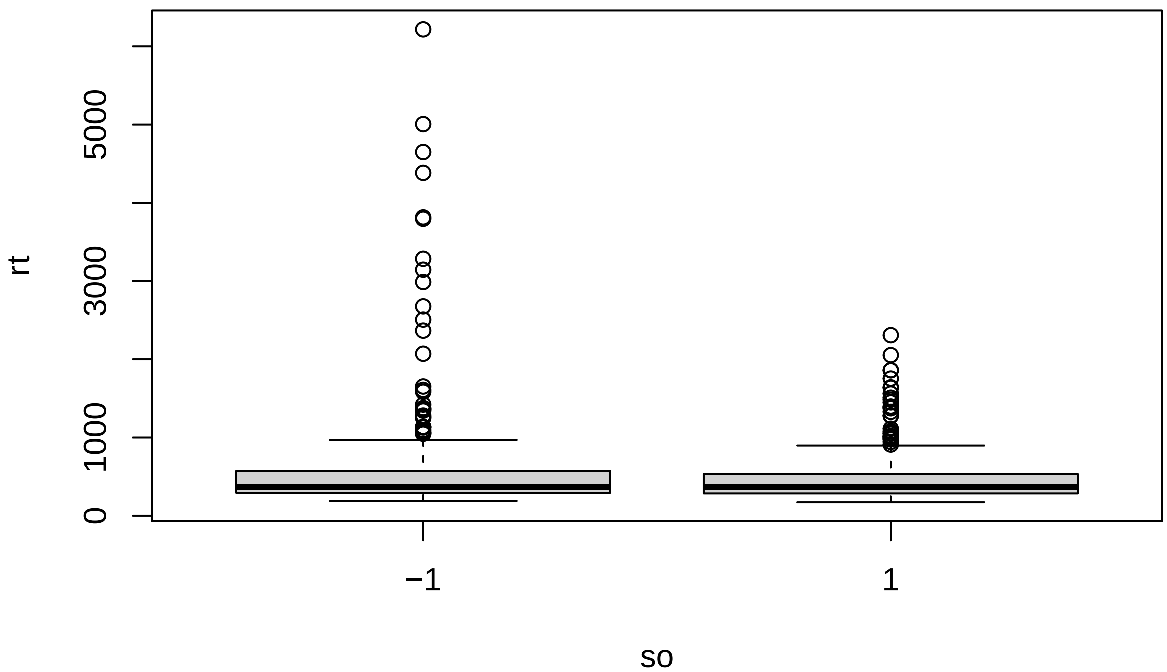
```
summary(m_lmer)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ so + (1 + so | subj) + (1 + so | item)
##    Data: gw1
##
## REML criterion at convergence: 8481.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8275 -0.4036 -0.1886  0.0575  8.4268
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subj     (Intercept)  25727   160.40
##           so            9492    97.43   -1.00
##  item     (Intercept)  23837   154.39
##           so            5036    70.96   -1.00
##  Residual             295555   543.65
## Number of obs: 547, groups:  subj, 37; item, 15
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   547.33      53.21  10.287
## so            -59.85      33.74  -1.774
##
## Correlation of Fixed Effects:
##    (Intr)
## so -0.647
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

Always visualize the data first:

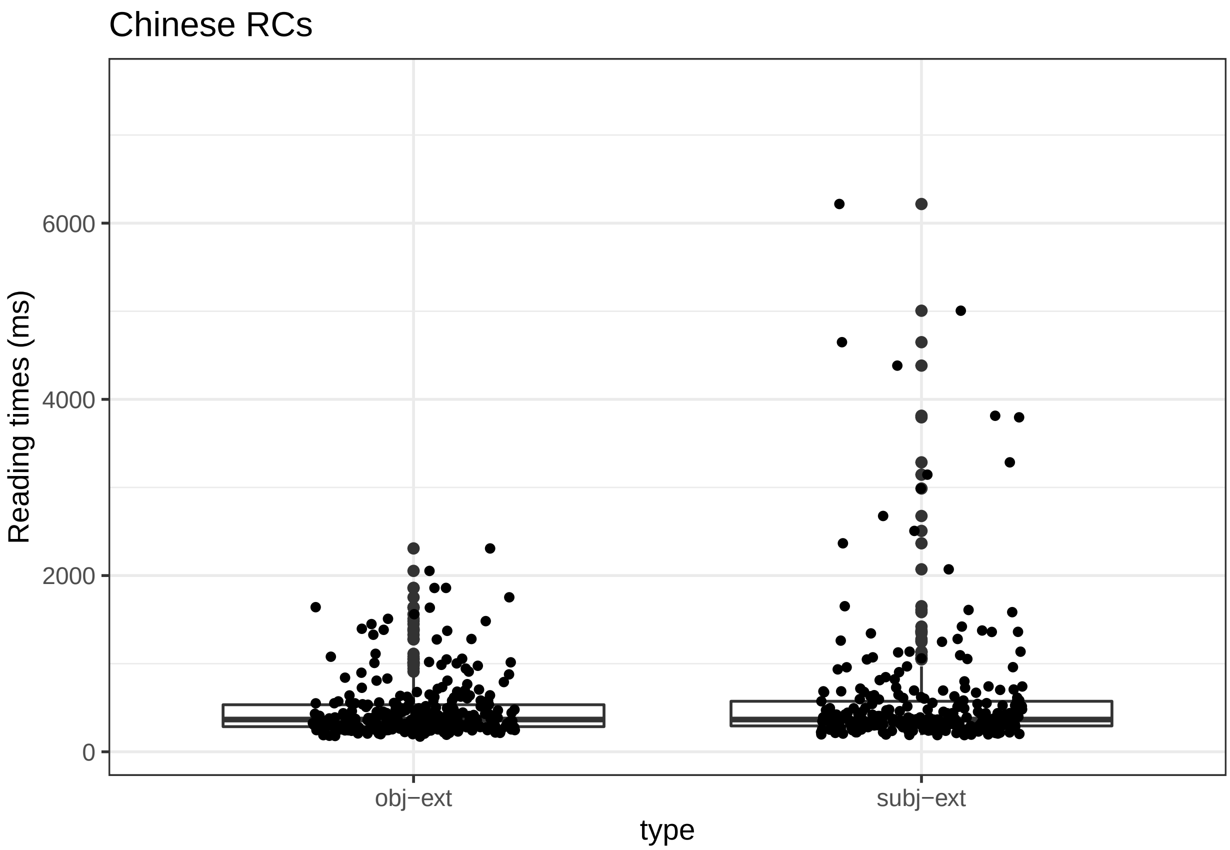
```
boxplot(rt~so,gw1)
```



Better:

```
p1CN <- ggplot(gw1, aes(x=type, y=rt)) +
  geom_boxplot() + ggtitle("Chinese RCs")
p1CN<-p1CN+geom_jitter(shape=16,
                       position=position_jitter(0.2))+
  theme_bw()+#scale_y_continuous(trans='log2')+
  ylab("Reading times (ms)")+
  coord_cartesian(ylim = c(100,7500))
```

```
p1CN
```



Log-transformed reading times:

```
m_lmerlog<-lmer(log(rt)~so + (1+so|subj)+(1+so|item),gw1)
```

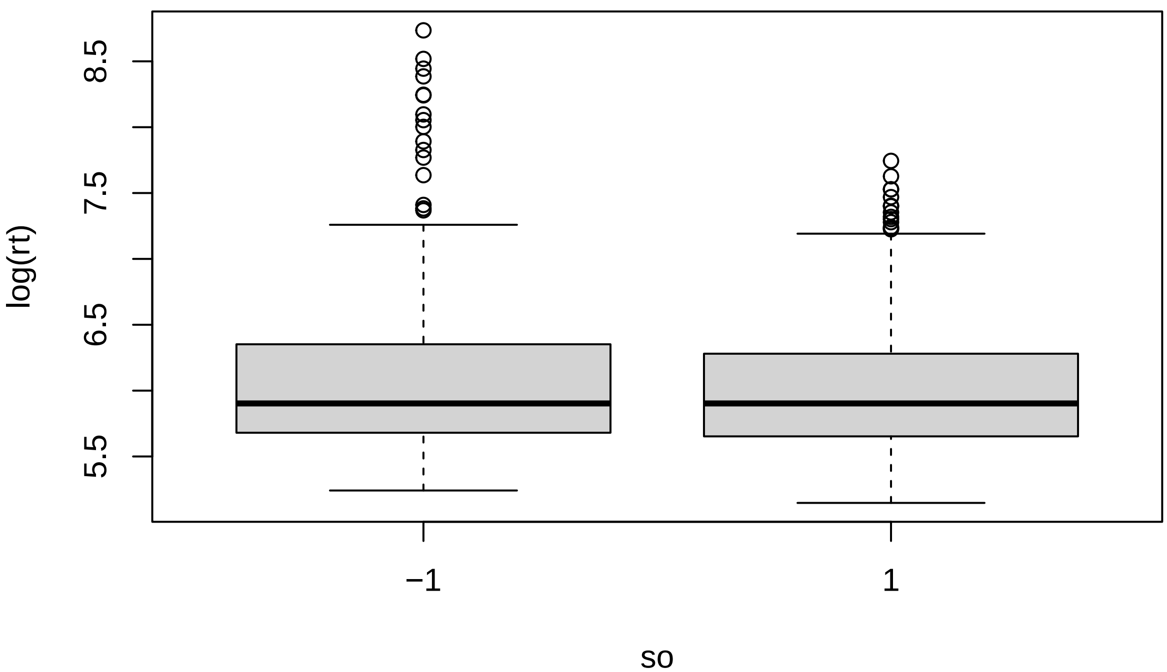
```
## boundary (singular) fit: see ?isSingular
```

```
summary(m_lmerlog)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(rt) ~ so + (1 + so | subj) + (1 + so | item)
##    Data: gw1
##
## REML criterion at convergence: 912.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7867 -0.5977 -0.2524  0.2944  4.4215
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.  Corr
##  subj     (Intercept) 5.993e-02 0.2448124
##           so          3.544e-03 0.0595305 -1.00
##  item     (Intercept) 3.314e-02 0.1820376
##           so          4.690e-08 0.0002166 1.00
##  Residual             2.645e-01 0.5143263
## Number of obs: 547, groups:  subj, 37; item, 15
```

```
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  6.06180    0.06572  92.235
## so          -0.03625    0.02415  -1.501
##
## Correlation of Fixed Effects:
##    (Intr)
## so -0.251
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
boxplot(log(rt)~so,gw1)
```



Bayesian analysis:

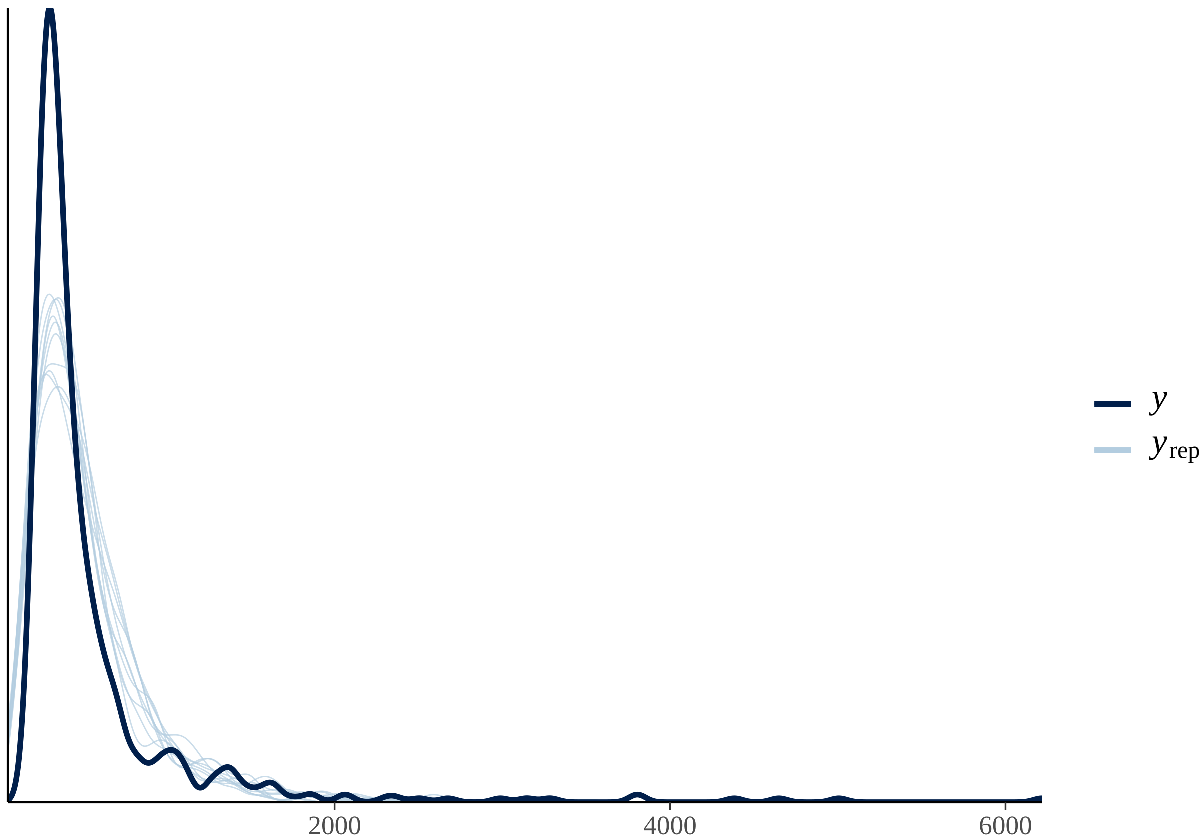
```
priors <- c(set_prior("normal(6, 1.5)", class = "Intercept"),
            set_prior("normal(0, .05)", class = "b",
                      coef = "so"),
            set_prior("normal(0, 1)", class = "sd"),
            set_prior("normal(0, 1)", class = "sigma"),
            set_prior("lkj(2)", class = "cor"))

m_gw<-brm(rt~so + (1+so|subj) + (1+so|item),gw1,family=lognormal(),
          prior=priors)
summary(m_gw)
```

Posterior predictive check:

```
pp_check(m_gw)
```

```
## Using 10 posterior draws for ppc type 'dens_overlay' by default.
```



Summarize the effect of interest (in ms), and summarize individual-level variation:

```
## graphical visualization:
postgw<-posterior_samples(m_gw)
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

```
## extract posteriors:
alpha<-postgw$b_Intercept
beta<-postgw$b_so
cor<-posterior_samples(m_gw,"^cor")
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

```
sd<-posterior_samples(m_gw,"^sd")
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

```
sigma<-posterior_samples(m_gw,"sigma")
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

```
## subject level effects:
subj_re<-posterior_samples(m_gw,"^r_subj")
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

```
item_re<-posterior_samples(m_gw,"^r_item")
```

```
## Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
## recommended alternatives.
```

Mean effect in ms:

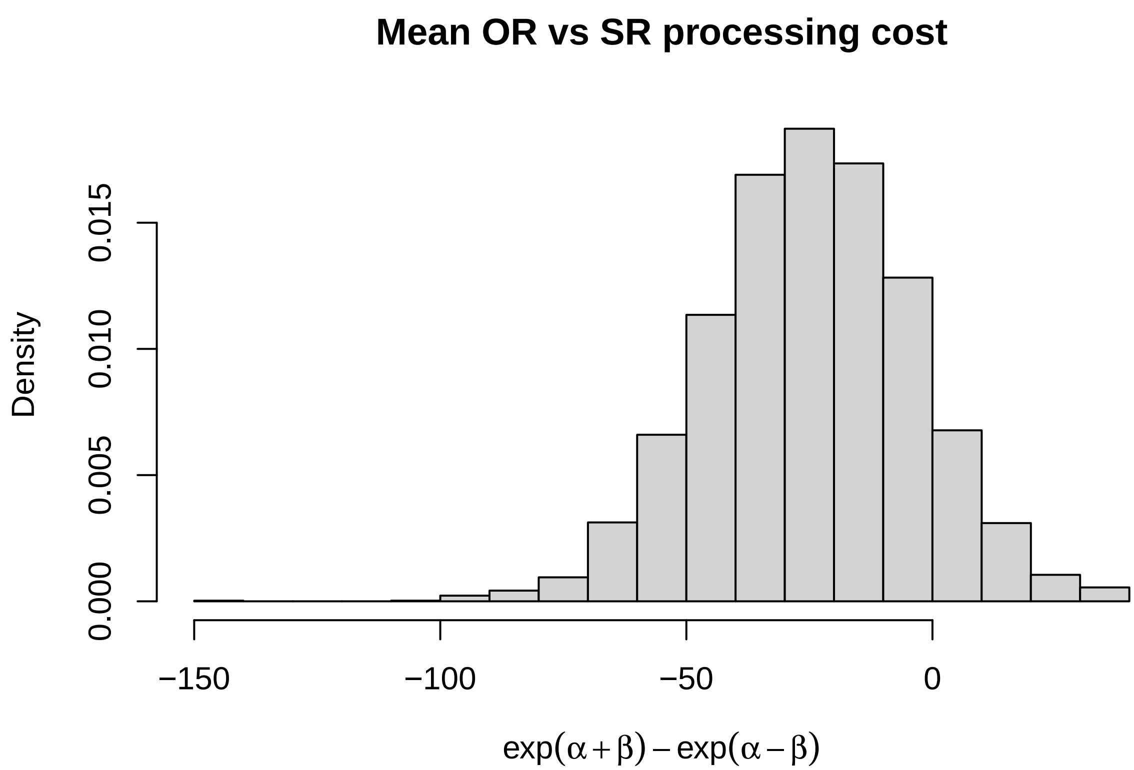
```
## mean effect on ms scale:
meandiff<- exp(alpha + beta) - exp(alpha - beta)
mean(meandiff)
```

```
## [1] -24.76652
```

```
round(quantile(meandiff,prob=c(0.025,0.975)),0)
```

```
##  2.5% 97.5%
##   -66    16
```

```
## mean effect:
hist(meandiff,freq=FALSE,
     main="Mean OR vs SR processing cost",
     xlab=expression(exp(alpha + beta)- exp(alpha - beta)))
```

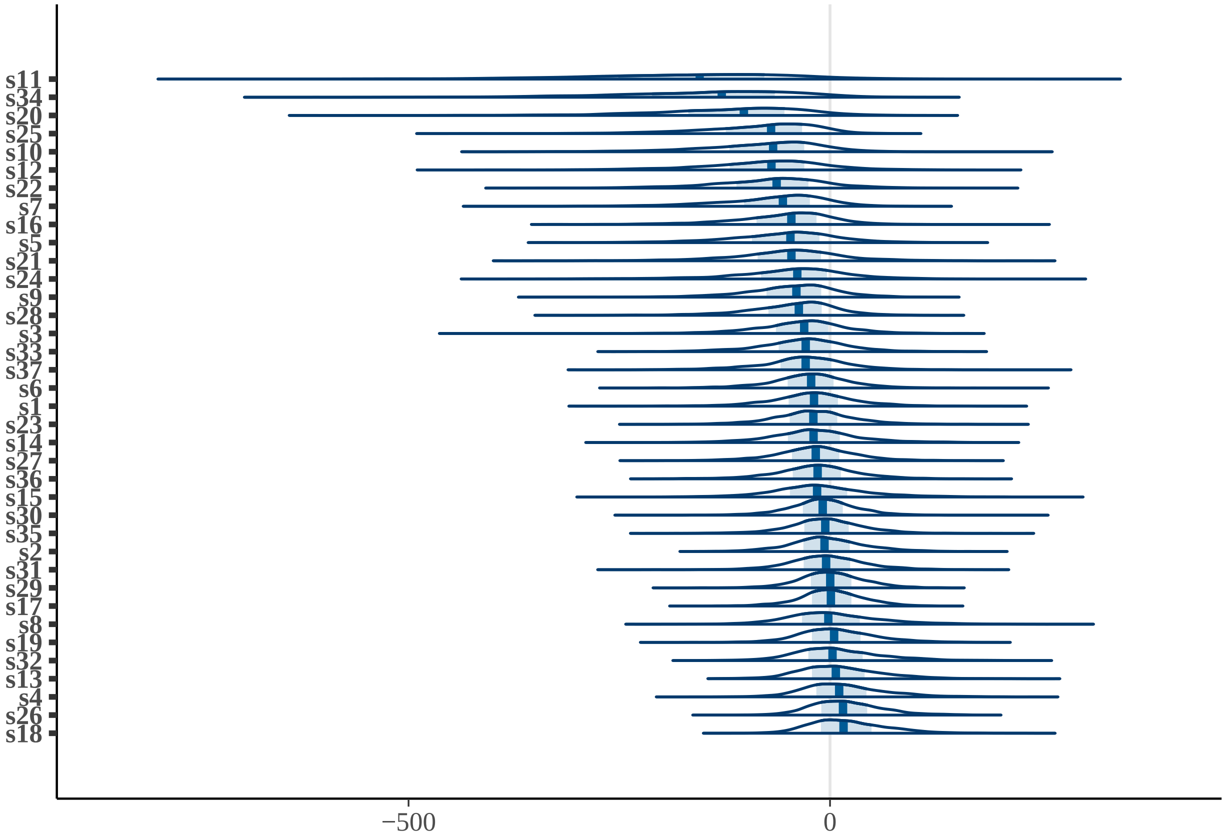


Individual differences:

```
## individual level estimates:
nsubj<-37
subjdiff<-matrix(rep(NA,nsubj*4000),nrow=nsubj)
for(i in 1:nsubj){
  subjdiff[i,]<-exp(alpha + subj_re[,i]  + (beta+subj_re[,i+nsubj])) -
    exp(alpha + subj_re[,i] -
          (beta+subj_re[,i+nsubj]))
}
```

```
subjdiff<-t(subjdiff)

subjdiff<-as.data.frame(subjdiff)
colnames(subjdiff)<-paste("s",c(1:nsubj),sep="")
mns <- colMeans(subjdiff)
subjdiff<-subjdiff[,order(mns)]
mcmc_areas(subjdiff)
```



Shrinkage in action:

```
## lmer
m_lmer<-lmer(log(rt)~so + (1+so|subj),gw1)
```

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

```
summary(m_lmer)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(rt) ~ so + (1 + so | subj)
##    Data: gw1
##
## REML criterion at convergence: 949.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7673 -0.6143 -0.2896  0.2869  4.1670
##
## Random effects:
```

```
##  Groups   Name        Variance Std.Dev. Corr
##  subj     (Intercept) 0.058306 0.24147
##           so          0.003115 0.05581  -1.00
##  Residual             0.297789 0.54570
## Number of obs: 547, groups:  subj, 37
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  6.06326    0.04612 131.470
## so          -0.03825    0.02515  -1.521
##
## Correlation of Fixed Effects:
##    (Intr)
## so -0.320
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
alphalmer<-coef(m_lmer)$subj[,1]
betalmer<-coef(m_lmer)$subj[,2]
lmerest<-exp(alphalmer+betalmer)-exp(alphalmer-betalmer)


## lmList:
m_lmlist<-lmList(log(rt)~so|subj,gw1)
summary(m_lmlist)
```

```
## Call:
##   Model: log(rt) ~ so | NULL
##    Data: gw1
##
## Coefficients:
##    (Intercept)
##    Estimate Std. Error  t value      Pr(>|t|)
## 1  6.061340  0.1417256 42.76814 1.190267e-164
## 2  5.961110  0.1417256 42.06093 6.155046e-162
## 3  6.080284  0.1417256 42.90181 3.678931e-165
## 4  5.877389  0.1417256 41.47021 1.191799e-159
## 5  6.296298  0.1417256 44.42599 6.566986e-171
## 6  5.989485  0.1417256 42.26115 1.043101e-162
## 7  6.090260  0.1417256 42.97220 1.984240e-165
## 8  6.070480  0.1417256 42.83264 6.753110e-165
## 9  6.119232  0.1417256 43.17663 3.313675e-166
## 11 6.260708  0.1417256 44.17486 5.703649e-170
## 12 6.874514  0.1417256 48.50581 1.030612e-185
## 14 6.442723  0.1417256 45.45914 9.754064e-175
## 15 5.909964  0.1417256 41.70005 1.528124e-160
## 16 6.145836  0.1417256 43.36434 6.434453e-167
## 17 6.211829  0.1417256 43.82998 1.124050e-168
## 18 6.106329  0.1417256 43.08558 7.348587e-166
## 19 5.489396  0.1417256 38.73257 8.309366e-149
## 20 5.781275  0.1417256 40.79204 5.305115e-157
## 21 5.835589  0.1417256 41.17527 1.678853e-158
## 22 6.451102  0.1417256 45.51826 5.912397e-175
## 23 6.315054  0.1417256 44.55832 2.108395e-171
## 24 6.375148  0.1417256 44.98234 5.611574e-173
```

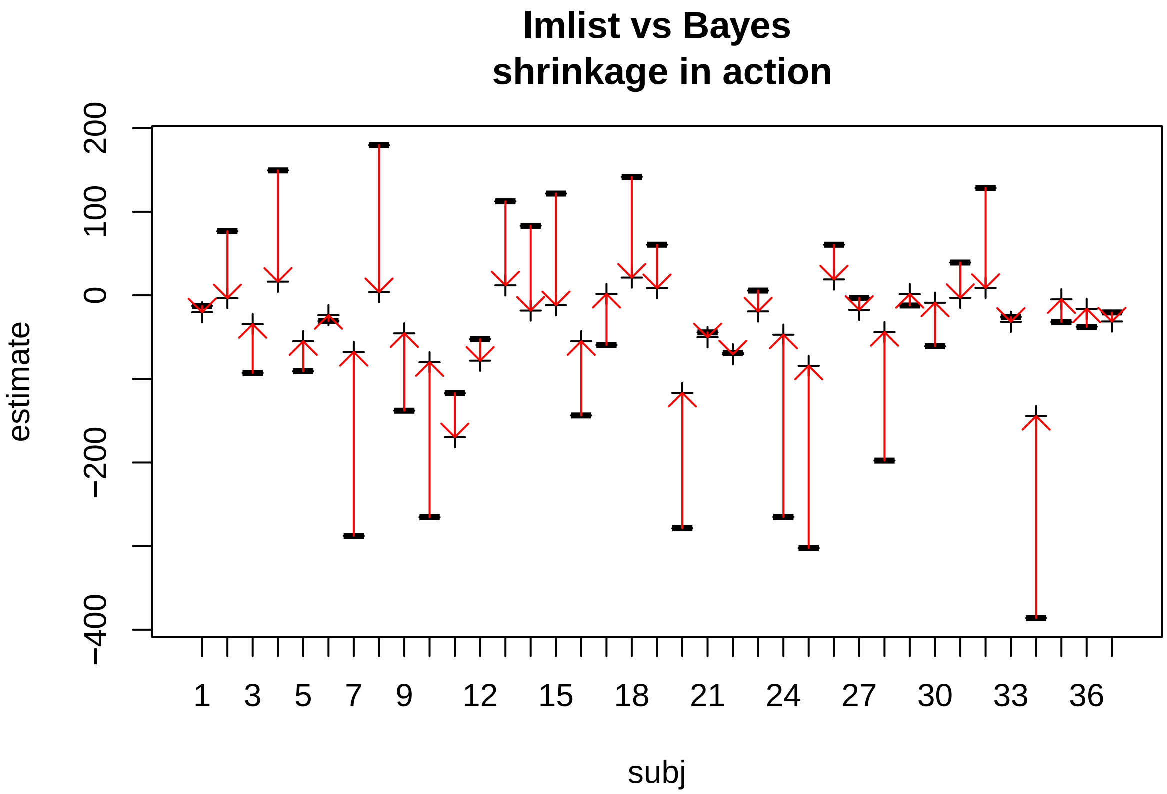
```
## 26 6.007549  0.1417256 42.38860 3.378055e-163
## 27 6.382835  0.2291108 27.85916 8.390473e-102
## 28 6.112141  0.1417256 43.12659 5.132916e-166
## 29 5.625091  0.1417256 39.69002 1.205626e-152
## 30 5.963775  0.1417256 42.07973 5.208933e-162
## 31 5.970888  0.1417256 42.12992 3.336909e-162
## 32 5.617204  0.1417256 39.63437 2.008827e-152
## 33 5.749526  0.1417256 40.56802 4.027046e-156
## 34 5.875937  0.1417256 41.45996 1.306305e-159
## 35 5.990470  0.1417256 42.26809 9.808911e-163
## 36 6.074569  0.1417256 42.86149 5.241552e-165
## 37 6.620538  0.1417256 46.71379 2.589424e-179
## 38 5.767435  0.1417256 40.69438 1.282652e-156
## 39 5.943168  0.1417256 41.93434 1.895725e-161
## 40 6.093228  0.1417256 42.99314 1.651418e-165
##    so
##        Estimate Std. Error     t value   Pr(>|t|)
## 1  -0.014869162  0.1417256 -0.10491516 0.91648760
## 2   0.098648233  0.1417256  0.69605103 0.48673874
## 3  -0.106024406  0.1417256 -0.74809649 0.45477375
## 4   0.207841895  0.1417256  1.46650943 0.14317406
## 5  -0.083519588  0.1417256 -0.58930498 0.55593806
## 6  -0.038598965  0.1417256 -0.27235003 0.78547174
## 7  -0.320374267  0.1417256 -2.26052541 0.02424242
## 8   0.205982257  0.1417256  1.45338803 0.14677917
## 9  -0.151213615  0.1417256 -1.06694655 0.28654024
## 11 -0.250906117  0.1417256 -1.77036582 0.07731029
## 12 -0.060451847  0.1417256 -0.42654155 0.66990734
## 14 -0.041684282  0.1417256 -0.29411969 0.76879547
## 15  0.151928041  0.1417256  1.07198746 0.28427230
## 16  0.089024281  0.1417256  0.62814549 0.53021180
## 17  0.121766731  0.1417256  0.85917259 0.39068051
## 18 -0.159465621  0.1417256 -1.12517179 0.26108686
## 19 -0.122526292  0.1417256 -0.86453197 0.38773394
## 20  0.216742106  0.1417256  1.52930833 0.12685648
## 21  0.088397887  0.1417256  0.62372572 0.53310840
## 22 -0.218256357  0.1417256 -1.53999273 0.12423082
## 23 -0.040092268  0.1417256 -0.28288661 0.77738751
## 24 -0.058696711  0.1417256 -0.41415750 0.67894650
## 26  0.007005734  0.1417256  0.04943168 0.96059615
## 27 -0.222260121  0.2291108 -0.97009867 0.33249310
## 28 -0.329042255  0.1417256 -2.32168577 0.02067377
## 29  0.108974506  0.1417256  0.76891206 0.44232927
## 30 -0.004068741  0.1417256 -0.02870859 0.97710911
## 31 -0.249677089  0.1417256 -1.76169393 0.07876700
## 32 -0.022129152  0.1417256 -0.15614085 0.87598862
## 33 -0.096873073  0.1417256 -0.68352569 0.49460941
## 34  0.054987701  0.1417256  0.38798714 0.69820010
## 35  0.159957194  0.1417256  1.12864027 0.25962172
## 36 -0.029777874  0.1417256 -0.21010939 0.83367277
## 37 -0.254521228  0.1417256 -1.79587364 0.07315270
## 38 -0.049935445  0.1417256 -0.35233898 0.72474099
## 39 -0.049391188  0.1417256 -0.34849876 0.72762070
## 40 -0.023117694  0.1417256 -0.16311589 0.87049685
```

```
##
## Residual standard error: 0.5476797 on 473 degrees of freedom
```

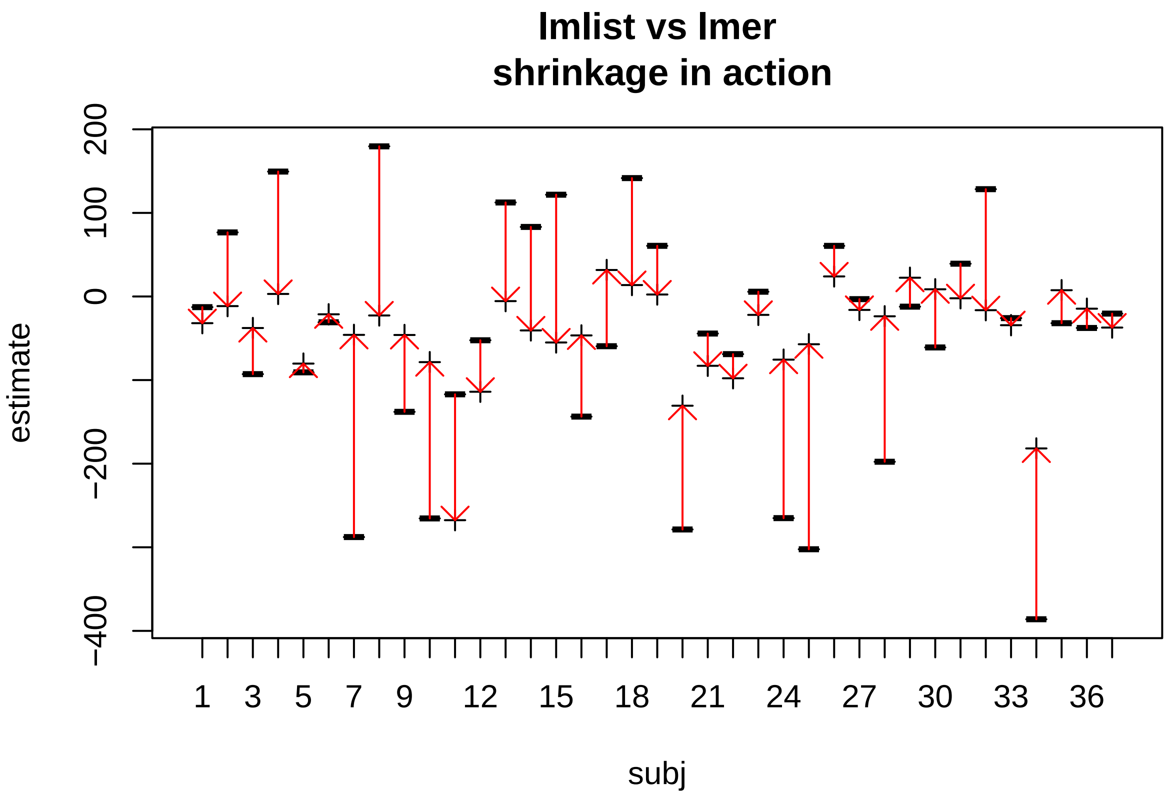
```
alphalmlist<-coef(m_lmlist)[1]
betalmlist<-coef(m_lmlist)[2]
lmlistest<-exp(alphalmlist+betalmlist)-exp(alphalmlist-betalmlist)

comparison<-data.frame(subj=factor(1:37),brm=mns,lmlist=lmlistest,lmer=lmerest)
colnames(comparison)[c(3,4)]<-c("lmlist","lmer")
```

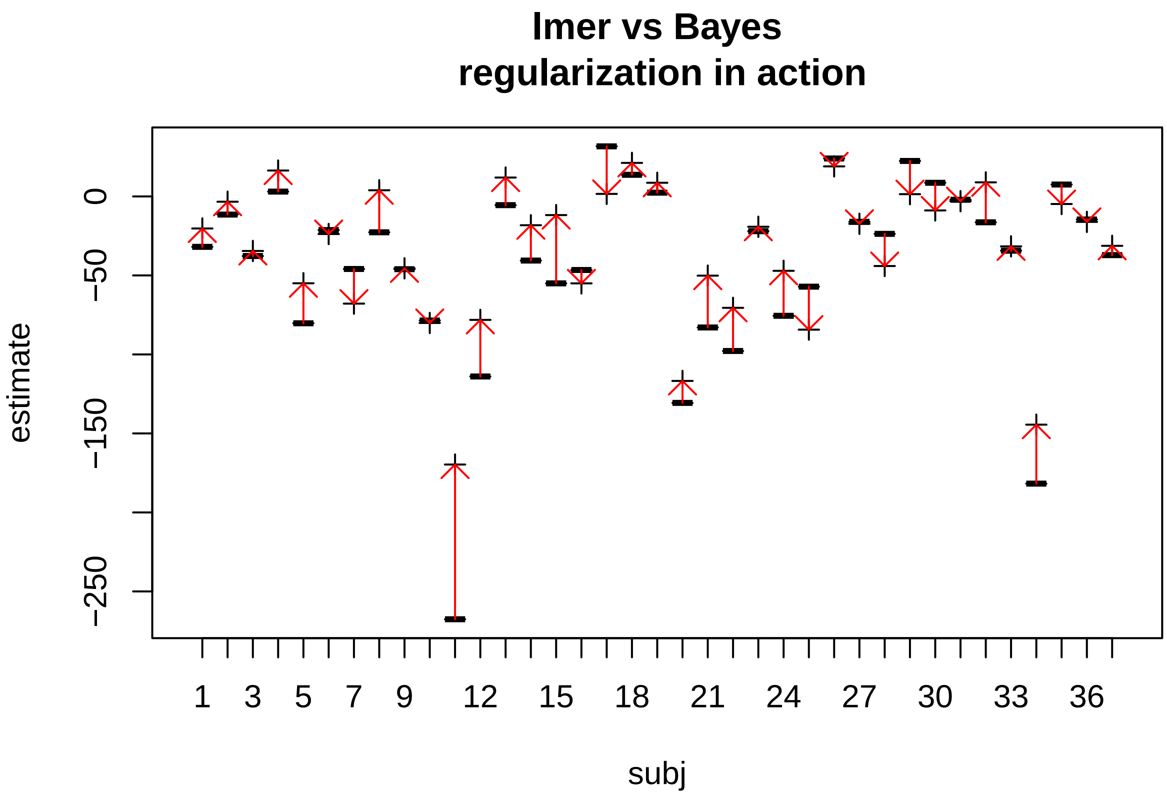
```
plot(lmlist~subj,comparison,ylab="estimate",main="lmlist vs Bayes\n shrinkage in action")
points(brm~subj,pch=3,comparison)
arrows(x0=1:37,y0=comparison$lmlist,x1=1:37,y1=comparison$brm,
       angle=45,code=2,length=0.1,col="red")
```




```
plot(lmlist~subj,comparison,ylab="estimate",main="lmlist vs lmer\n shrinkage in action")
points(lmer~subj,pch=3,comparison)
arrows(x0=1:37,y0=comparison$lmlist,x1=1:37,y1=comparison$lmer,
       angle=45,code=2,length=0.1,col="red")
```




```
plot(lmer~subj,comparison,ylab="estimate",main="lmer vs Bayes\n regularization in action")
points(brm~subj,pch=3,comparison)
arrows(x0=1:37,y0=comparison$lmer,x1=1:37,y1=comparison$brm,
       angle=45,code=2,length=0.1,col="red")
```



Bayes factors analysis

```
## Bayes factor analysis:
m_gw<-brm(rt~so + (1+so|subj) + (1+so|item),gw1,family=lognormal(),
          prior=priors,warmup=5000,iter=20000,
          save_pars=save_pars(all=TRUE))
summary(m_gw)

priors0 <- c(set_prior("normal(6, 1.5)", class = "Intercept"),
             set_prior("normal(0, 1)", class = "sd"),
             set_prior("normal(0, 1)", class = "sigma"),
             set_prior("lkj(2)", class = "cor"))

m_gw0<-brm(rt~1 + (1+so|subj),gw1,family=lognormal(),
           prior=priors0,,warmup=5000,iter=20000,
           save_pars=save_pars(all=TRUE))
```

Compute the BF several times to check that it is stable:

```
bayes_factor(m_gw,m_gw0)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 1
## Iteration: 2
## Iteration: 3
```

```
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of m_gw over m_gw0: 286984.32557
```

```
bayes_factor(m_gw,m_gw0)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of m_gw over m_gw0: 282470.41454
```

The Bayes factor is sensitive to the prior, so a sensitivity analysis is a must. Neer report just one Bayes factor with a (vague) prior on the target parameter. See:

Daniel J. Schad, Bruno Nicenboim, Paul-Christian Bürkner, Michael Betancourt, and Shravan Vasishth. Workflow Techniques for the Robust Use of Bayes Factors. Psychological Methods, 2022. https://osf.io/y354c/