JAGS implementation of our Unstructured Stroop Model

Jeff Rouder

August 27th, 2016

Let Y_{ijk} be the RT for the ith participant ($i=1,\dots,$I) in the jth condition ($j = 1, 2$ for congruent and incogrnuent, respectively) for the kth replicate, $k = 1, \dots, K_{ij}$.

The base model is

$$Y_{ijk} = \mathrm{N}(\alpha_i + x_j\theta_i, \sigma^2)$$

where $x_j = j - 1$ codes zero for congruent and 1 for incongruent. Hence, θ_i is the ith participants Stroop effect and is the target of inquiry. The model on σ^2, measured in seconds is informative but broad ($\sigma^2 \sim IG(2, .1)$). It has a mean and is tuned for standard deviations on the 100 ms scale or so, and is realistic for variabilities of repeated RTs in this task. Moreover, these settings are not too critical. The following code shows the pdf of this model on σ (not σ^2). You need the MCMCpack package

```
library('MCMCpack')
dsinvgamma=function(s,shape,scale) 2*s*dinvgamma(s*s,shape,scale)
s=seq(0,1,.001)
plot(s,dsinvgamma(s,2,.1),typ='l',ylab="Density",xlab="sigma (seconds)")
```
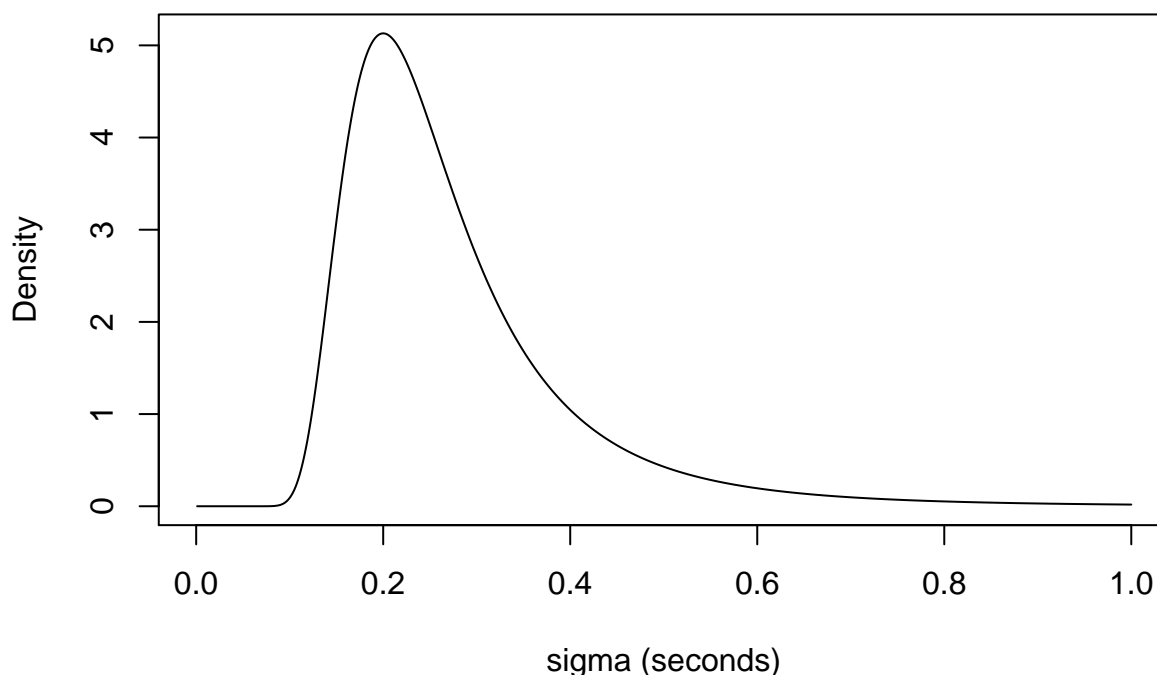

We place hierarchies on α_i and θ_i as follows:

$$\alpha_i \sim \mathrm{N}(\mu_alpha, \sigma_\alpha^2)$$

$$\theta_i \sim \mathrm{N}(\mu_\theta, \sigma_\theta^2)$$

Models on μ_alpha and σ_a^2lpha are not too critical, and we chose

$$\mu_alpha \sim \mathrm{N}(.8, .3^2), \quad \sigma_\alpha^2 \sim IG(2, .1)$$

Likewise, the model on μ_θ is not too critical, and we chose

$$\mu_t heta \sim \mathrm{M}(0, .15^2).$$

The critical prior is the one on σ_θ^2, and we consider

$$\sigma_\theta^2 \sim \mathrm{IG}(a, b)$$

where a, the shape, and b the scale are set. For effects that are 50 ms in size, give or take, we would expect the standard deviation to be on the same order. Here is a graph for two choices $(a = 1, b = .0006)$ and $(a = 2, b = .002)$ that focus mass on the 10 millisecond scale.

```
s=seq(0,.15,.001)
plot(s,dsinvgamma(s,2,.002),typ='l',ylab="Density",xlab="sigma_theta (seconds)")
lines(col='darkred',s,dsinvgamma(s,1,.0006))
```
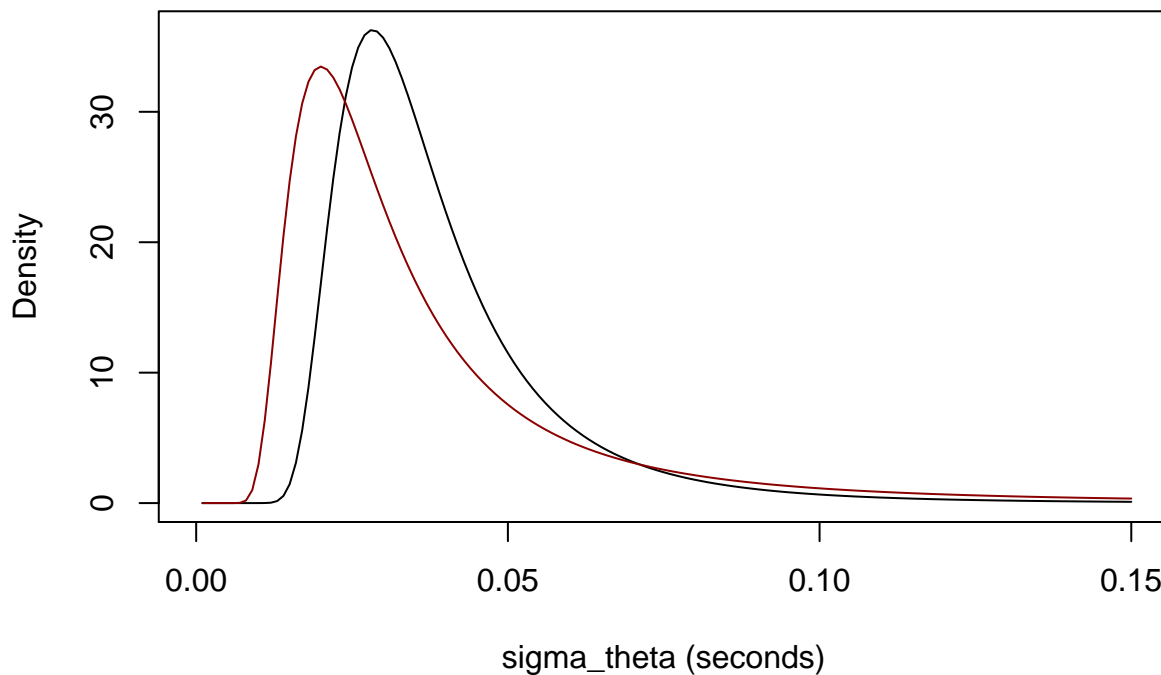

Now, let's load the data and identify naive sample Stroop effects:

```
library('rjags')
```

```
## Warning: package 'rjags' was built under R version 3.2.5
```

```
## Linked to JAGS 4.2.0
```

```
## Loaded modules: basemod,bugs
```

```
library('devtools')
```

```
## Warning: package 'devtools' was built under R version 3.2.5
```

```r
SourceURL <- "https://raw.githubusercontent.com/PerceptionCognitionLab/data0/master/contexteffects/Flan
source_url(SourceURL)
```

```
## SHA-1 hash of file is d0b0268d59853f2b5b98d9f3371e5f0f1fc71b35
```

```r
stroop$sub <- stroop$ID

y=stroop$rt
sub=stroop$sub
cond=stroop$cond
I=max(sub)
R=length(sub)
means=tapply(y,list(sub,cond),mean)
sample.effect=(means[,2]-means[,1])
```

And now the JAGS model specification + analysis:

```r
model2.string <-"
model{
    for(i in 1:R){
            y[i] ~ dnorm(alpha[sub[i]]+(cond[i]-1)*theta[sub[i]], tau)
        }
     for (i in 1:I){
        alpha[i] ~ dnorm(m.alpha, tau.alpha)
        theta[i] ~ dnorm(m.theta, tau.theta)
    }
  tau ~ dgamma(2,.1)
  m.alpha ~ dnorm(.8, 1/.3^2)
  m.theta ~ dnorm(0, 1/.15^2)
  tau.alpha ~ dgamma(2,.1)
  tau.theta ~ dgamma(1,.0006)
}
"
model2.spec<-textConnection(model2.string)

jags <- jags.model(model2.spec,
                   data = list('y' = y,
                               'R' = R,
                               'I'=I,
                               'sub'=sub,
                               'cond'=cond),
                   n.chains=1,
                   n.adapt=100)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 11245
##    Unobserved stochastic nodes: 247
##    Total graph size: 45733
##
## Initializing model
```

```
update(jags, 1000)
out=coda.samples(jags,
                 c('theta'),
                 10000)

mypar=out[[1]]
pm.theta=apply(mypar,2,mean)
```

So, how much shrinkage did we get?

```
plot(sample.effect,pm.theta,xlim=c(-.125,.2),ylim=c(-.125,.2),xlab="Sample Effects",ylab="Model-Shrunk
abline(0,1)
```
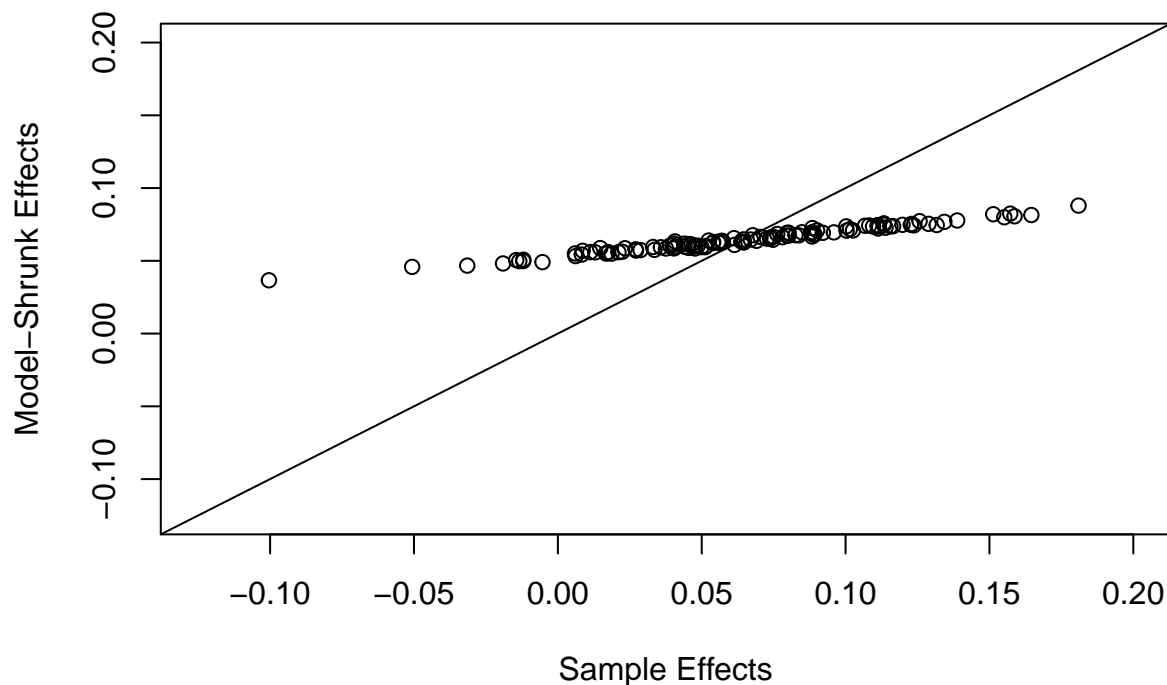

A lot! One measure is the standard deviation of the posterior means of θ.
It is

```
print(sd(pm.theta))
```

```
## [1] 0.008769965
```

This value is about half that estimated for the same model (sans informative prior) by other methods. Where did the half come from. Well, I placed a prior on the 10s of millisecond scale for $\sqrt{\sigma_\theta^2}$ because I know that is the order for potential variation of Stroop effects. Other more diffuse, generic priors will lead to less shrinkage. I view gaining this shrinkage with these specifications as adding the necessary value for judicious analysis. Others may differ. If you want to explore the effects of the prior on σ_θ^2, feel free to adapt the code! A lot of folks use uniforms with big ranges.

'tau.theta <- pow(sigma.theta, -2)   sigma.theta ~ dunif(0,1)

Here, you get smaller shrinkage, but the prior is unreasonalby broad. Also, priors like IG(.5,.5), an objective default, don't give you the same shrinkage because they are not tuned for the same order.