Reanalysis of original data

Kevin Potter

May 9, 2016

Here I reanalysis the final recognition test from the behavioral data of Wimber et al. (2015) using a generalized linear model, the goal being to use the posteriors subsequently as priors for the replication. Furthermore, Nature Neuroscience defines successful replications as ones who fall within the 95% confidence intervals of the original data. We will try to do the same using the credible intervals instead.

```
# We'll use Stan to fit a generalized linear model
library( rstanarm )
```

```
## Loading required package: Rcpp
```

```
## rstanarm (Version 2.9.0-3, packaged: 2016-02-13 16:23:33 UTC)
```

```
## - Do not expect the default priors to remain the same in future rstanarm versions.
```

```
## Thus, R scripts should specify priors explicitly, even if they are just the default
s.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
## options(mc.cores = parallel::detectCores())
```

```
options (mc.cores=parallel::detectCores ()) # Run on multiple cores

# Define a basic logistic (or sigmoidal) function
logistic = function(x) 1/(1+exp(-x))
# Define a basic logit function
logit = function(p) log( p/(1-p) )
```

```
# Load in data
load( 'Original_Rec_Mem_data.RData' )

# Determine the sample size
N = length( unique( OriginalRecMem$Subject ) )
```

The dependent variable of interest is accuracy (coded as 0 for wrong and 1 for correct), which we assume follows a Bernoulli distribution. The group-level factors we'll examine across the models are:

1. Image type (whether an image was a target vs. competitor).
2. Selective retrieval (whether an image was in the selective retrieval condition or not).

Additionally, we'll include specific subject-level and image-level effects.

```
d = OriginalRecMem # Create a data frame for data to be fitted
d$Y = d$Accuracy # Dependent variable
d$IT = as.factor( d$ImageType ) # Image type (1 = target, 2 = competitor)
d$SR = as.factor( 1 - d$Baseline ) # Selective retrieval ( 1 = yes, 0 = no )
d$S = as.factor( d$Subject )
d$I = as.factor( d$ImageNum )
d$Trial = rep(1,nrow(d)) # Nuisance parameter for bernoulli distribution
```

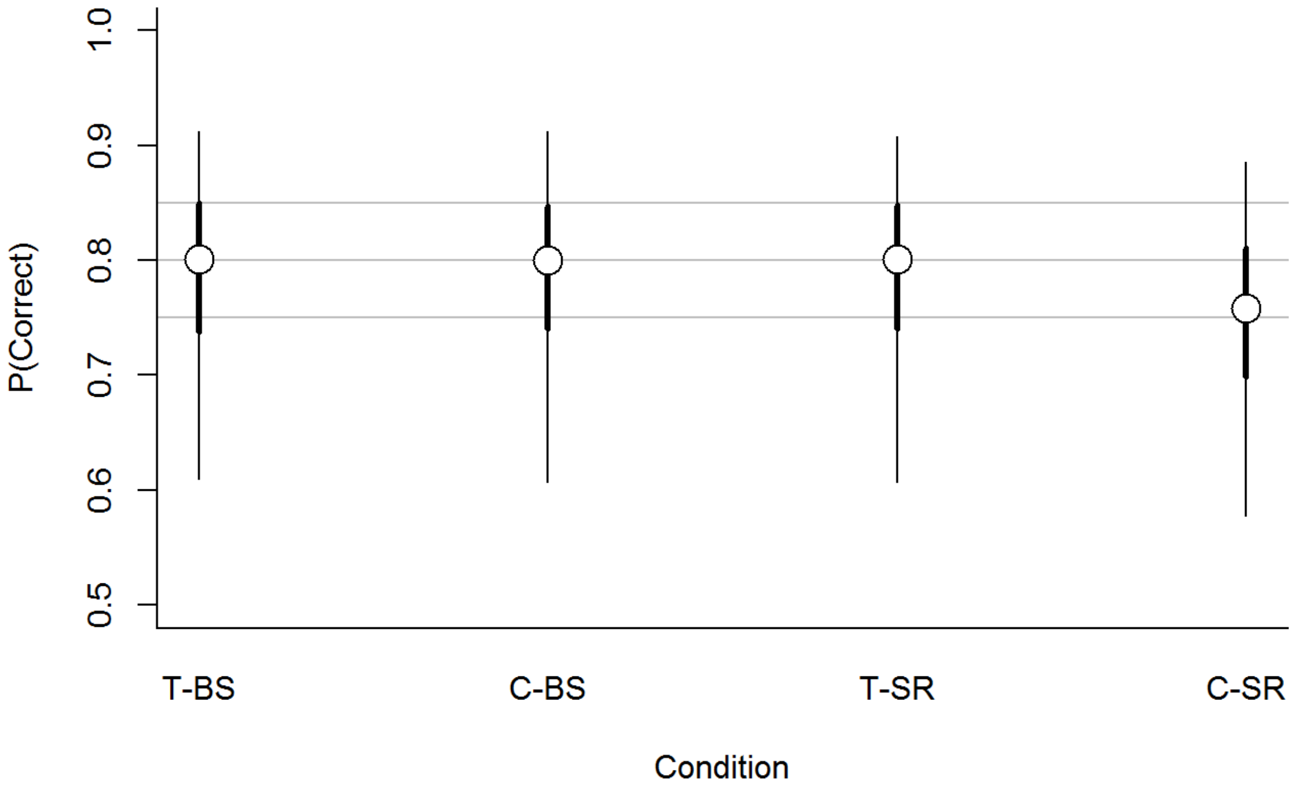
However, the predicted effect is that there should be no difference between baseline conditions and the selective retrieval condition with targets, but there should be a drop in performance for the selective retrieval condition with competitors. We'll create a dummy-coded variable representing that.

```
d$RIF = 0; d$RIF[ d$IT == 2 & d$SR == 1 ] = 1
```

We'll need to specify priors for the model. Specifically, we'll specify the intercept as a Normal(1.775, .3). As for the coefficient representing the RIF effect, we'll specify a Normal(-.3, .3) prior. Given that a large number of studies report a RIF effect, this prior should reflect the average researcher's belief that the effect is present. A prior predictive check can tell us the range of responses the model predicts using these priors.

```
# Fit a model conditioned on the priors instead of the outcomes
fit = stan_glmer(cbind(Y, Trial) ~ RIF + (1|S) + (1|I), data = d, family = binomial("lo
git"), prior_intercept = normal(1.775,.3), prior = normal( -.3, .3 ), prior_PD = TRUE,c
hains = 4, cores = 4, seed = 7453)

# Simulate data from the priors
nd = d; nd$Y = 0;
Sim = posterior_predict( fit, newdata = nd )
```



With these priors, the model predicts an average accuracy of 80% across subjects, and a RIF effect of about 4%. Accuracy ranges roughly between 60 to 90% though. We'll now fit the model to the actual data and update our priors.

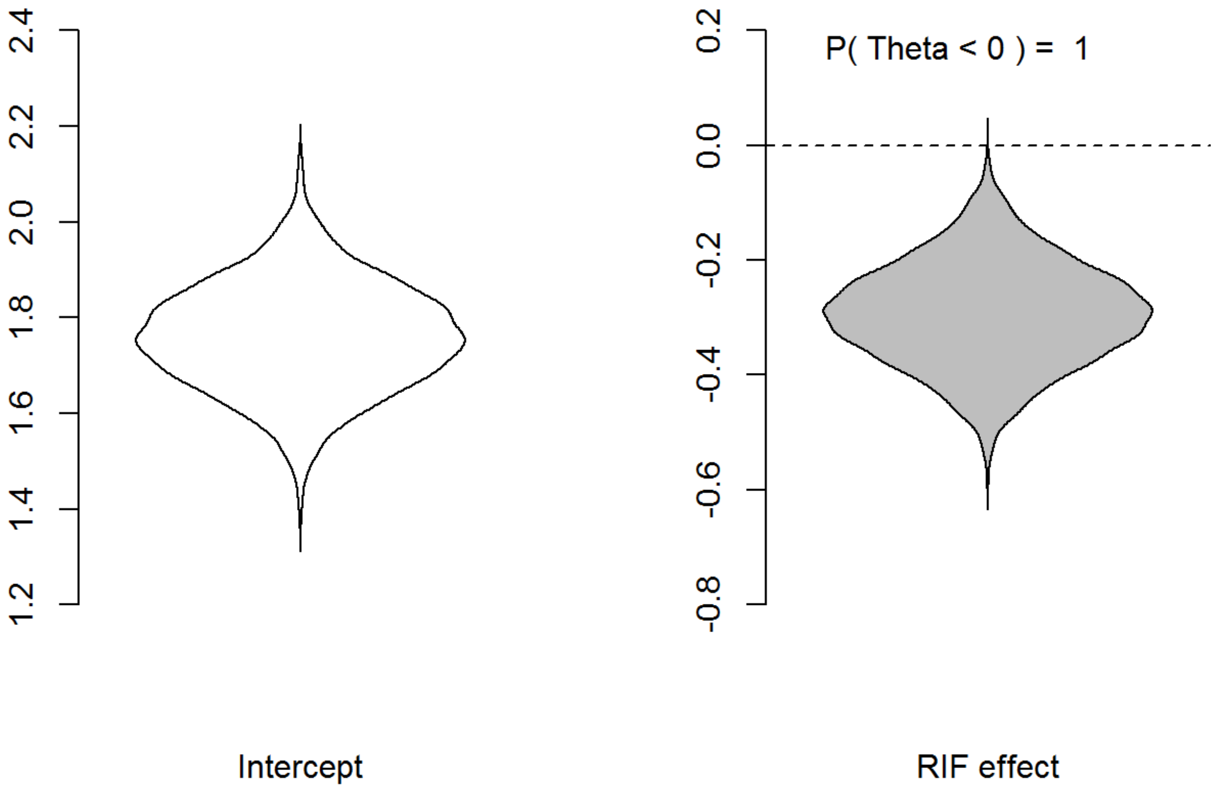
```
# Fit the model to the original data
fit = stan_glmer(cbind(Y, Trial) ~ RIF + (1|S) + (1|I), data = d, family = binomial("lo
git"), prior_intercept = normal(1.775,.3), prior = normal( -.3, .3 ), chains = 8, cores
= 8, seed = 2834,iter=1250,warmup=500)

# Extract the posterior estimates
post = as.matrix( fit )
```

The resulting posteriors suggest that the posterior probability for the RIF effect is essentially guaranteed to be negative (i.e. 100% of credible mass is less than 0).

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

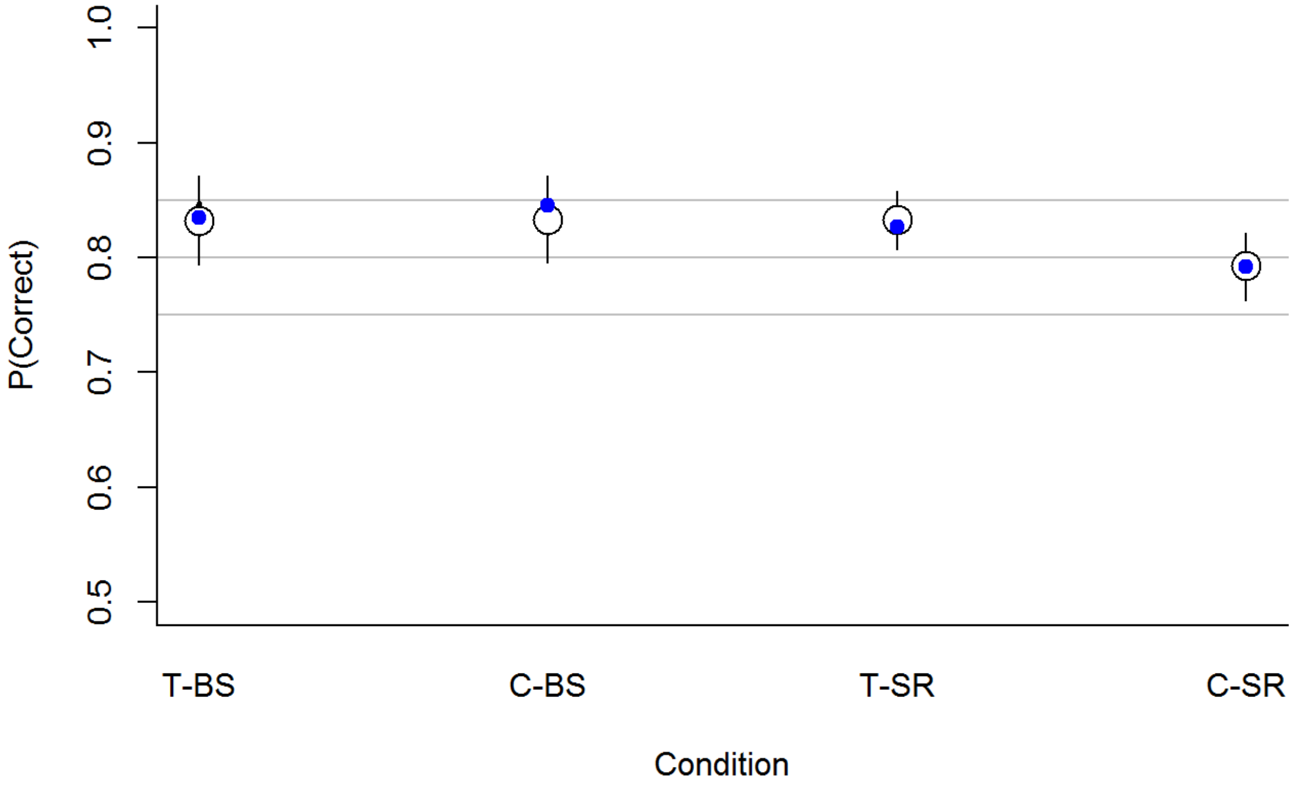
```
## [1] 1
```



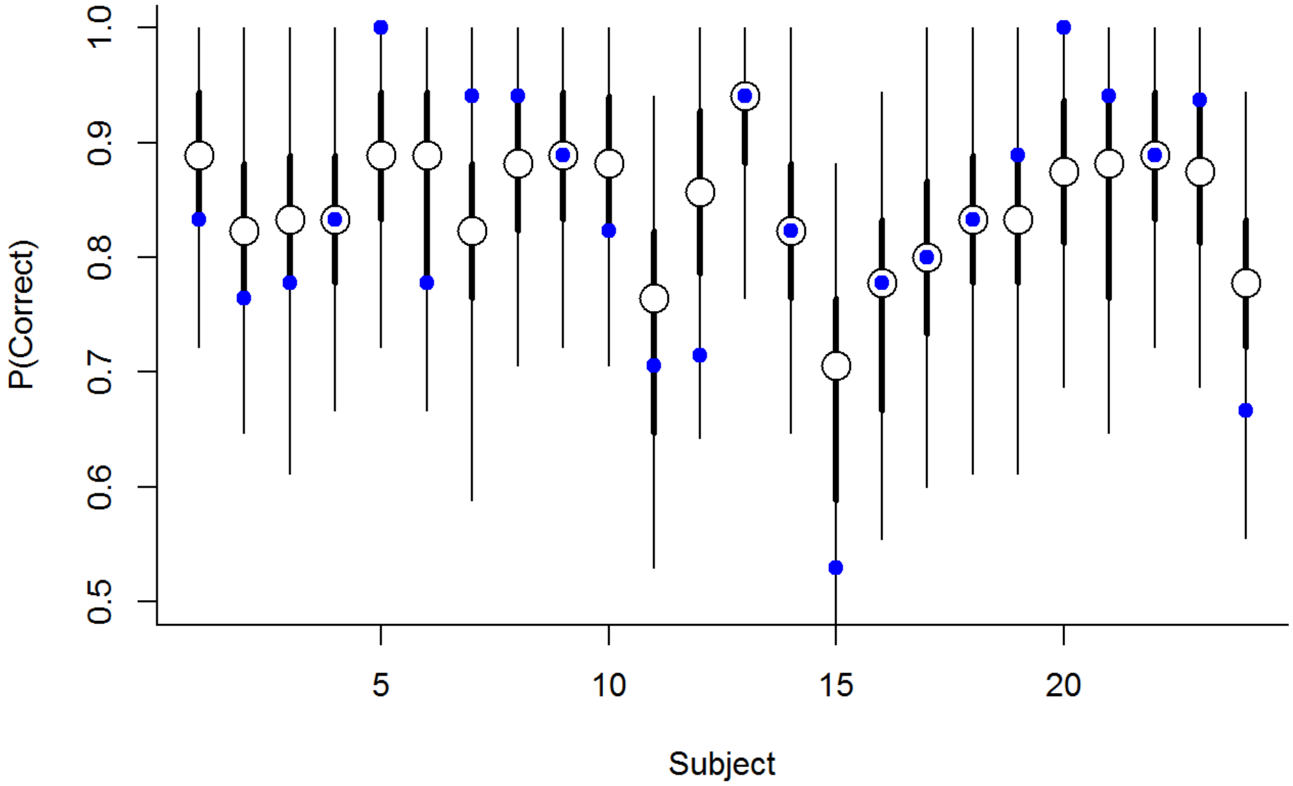
```
# Credible intervals
print( round( quantile( post[,2], c( .025, .975 ) ), 2 ) )
```

```
##  2.5% 97.5%
## -0.47 -0.11
```

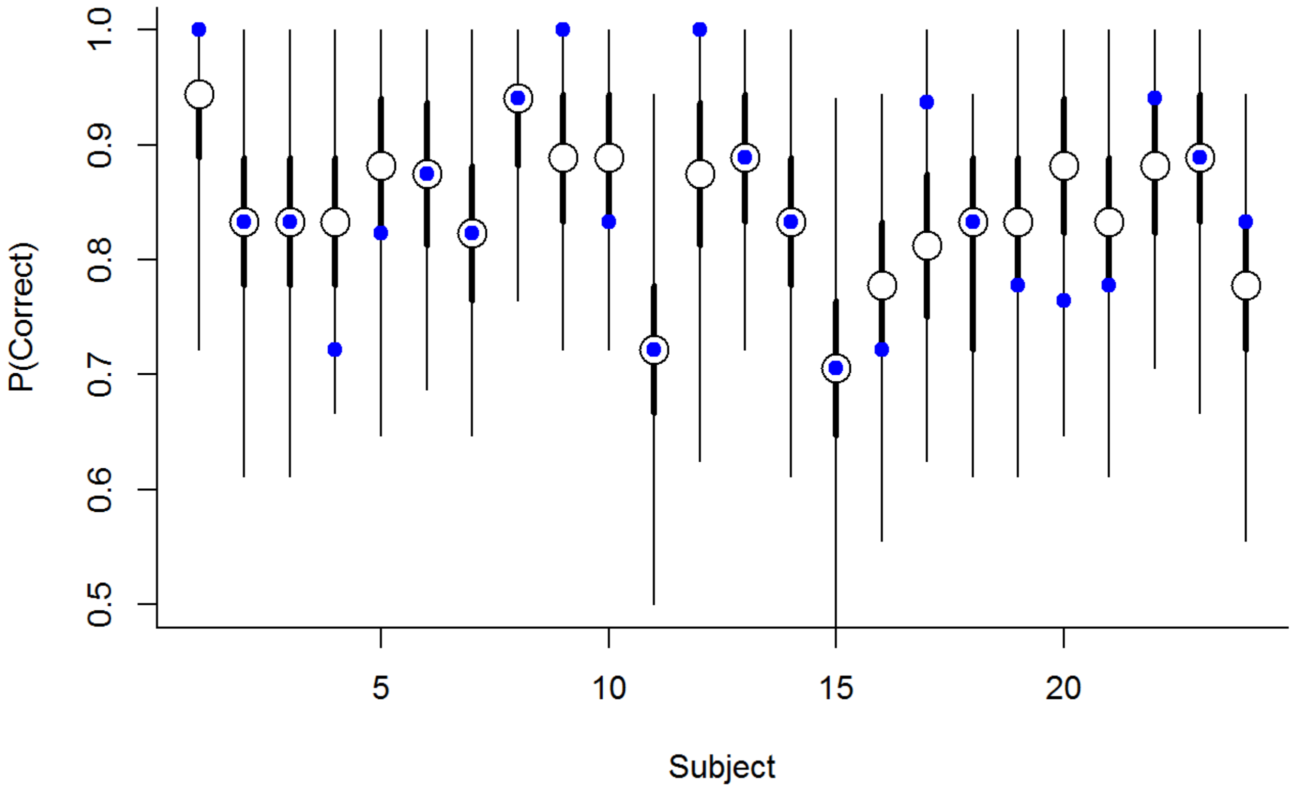
To examine how well the model performs, we can examine how well it retrodicts the data to which the model was fitted.

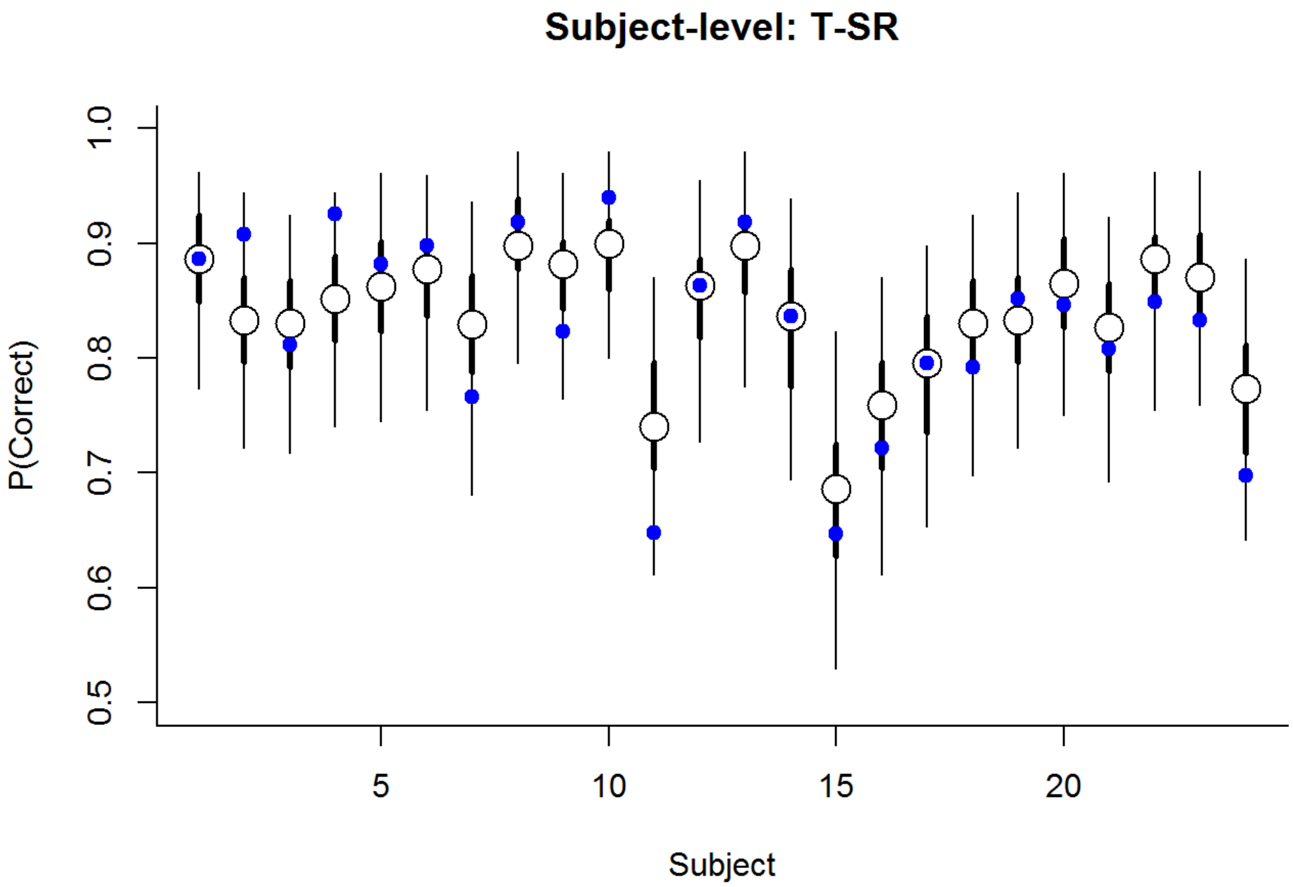


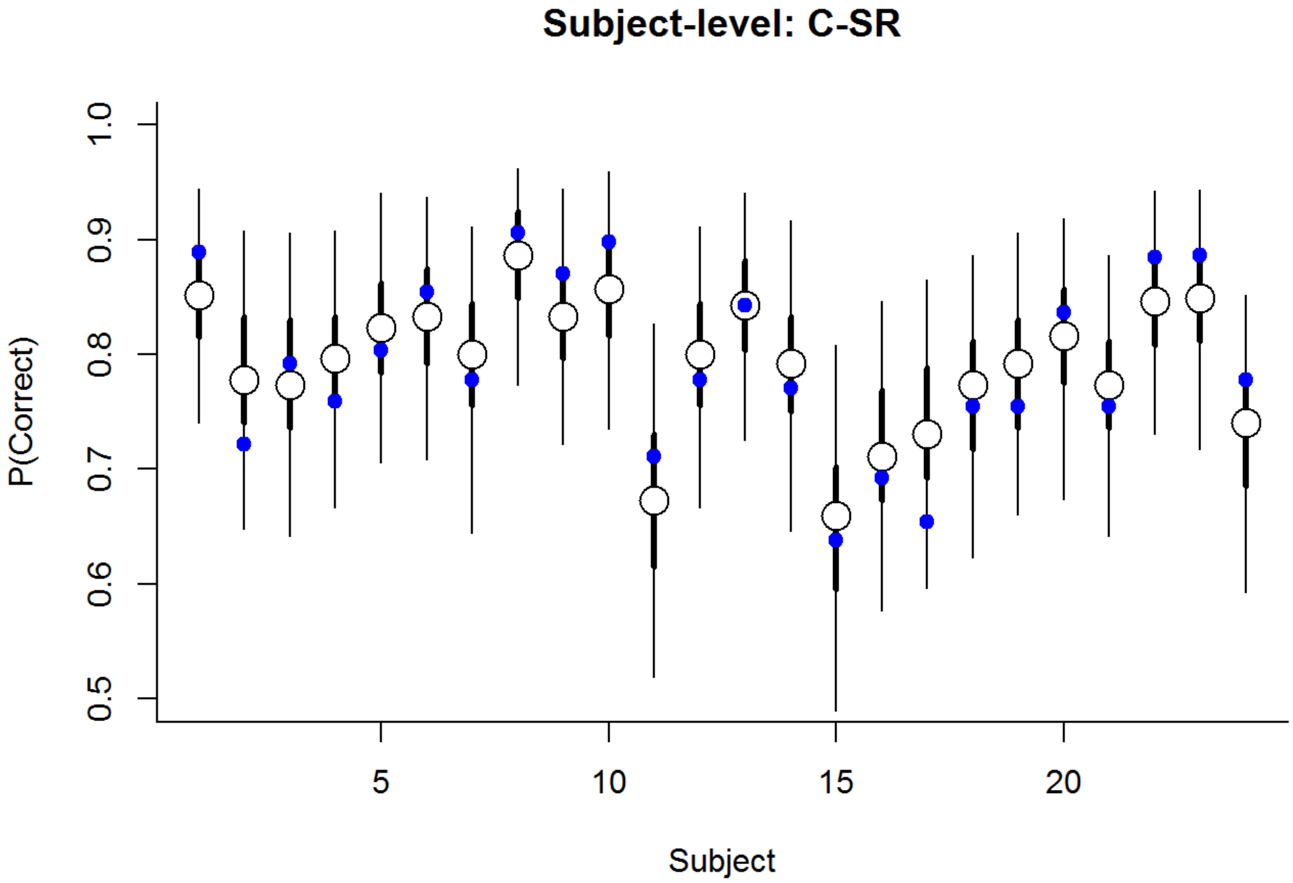
Subject-level: T-BS



Subject-level: C-BS


P(Correct)
0.5
0.6
0.7
0.8
0.9
1.0
5
10
15
20
Subject


Subject-level: T-SR


Subject-level: C-SR

The model does quite well at the group-level, but (unsurprisingly) is less precise at the subject level. Finally, assuming that the posteriors are well approximated by normal distributions, we can use the mean and standard deviation of the posterior approximations for the intercept and RIF coefficient to determine the subsequent priors for the replication analyses.

```
## Mean and standard deviation for intercept:
```

```
## 1.76 , 0.11
```

```
## Mean and standard deviation for RIF coefficient:
```

```
## -0.29 , 0.09
```