Analysis of Paape and Vasishth local coherence paper

Dario Paape and Shravan Vasishth

September 9, 2015

1 Preprocessing

```
library(lme4)

## Loading required package:  Matrix

library(ggplot2)
library(plotrix)
library(languageR)
library(gplots)

##
## Attaching package:  'gplots'
##
## The following object is masked from 'package:plotrix':
##
##    plotCI
##
## The following object is masked from 'package:stats':
##
##    lowess

library(MASS)
library(car)
library(xtable)
#sessionInfo()
```

```
# The bias predictor generated from corpus data
bias <- c(2.58,1.05,1.11,0.5,3.31,14.39,8.66,4.1,3.77,5.75,2.73,9.95,0.64,1.22,3.14,2.06,0.7
bias <- log(bias)
## imputing missing values?
bias<-append(bias,mean(bias),after=5)
```

```
bias<-append(bias,mean(bias),after=7)
bias <- bias-mean(bias)
items<-1:34
bias.data<-data.frame(items=items,bias=bias)

# Load data, name columns, calculate reciprocal reading times, remove test subjects
data<-read.table("lcdata1.txt",header=F)
colnames(data) <- c("subj","expt","item","cond","pos","word","roi","rt")
data$rrt <- -1000/data$rt
data<-subset(data,subj>5)

# Calculates % correct on comprehension questions
quest <- subset(data, substr(pos,0,1)=="?")
colnames(quest)[7] <- 'correct'
quest$correct <- as.numeric(as.character(quest$correct))

means.questions<-with(quest,tapply(correct,IND=cond,mean))
round(100*means.questions,digit=0)
```

```
##   -   a   b   d   e  m1 m10 m11 m12  m2  m3  m4  m5  m6  m7  m8  m9  s1
##  97  82  85  97  95  97  98 100  93  92  88  92  89  98 100  98  98  92
## s10 s11 s12  s2  s3  s4  s5  s6  s7  s8  s9
##  94  93 100 100  91  88  98  80 100  86  94
```

```
mean(round(100*means.questions,digit=0))
```

```
## [1] 93.62069
```

```
means.lcq<-with(subset(quest,expt=="lc"),tapply(correct,IND=cond,mean))
round(100*means.lcq,digit=0)
```

```
##   -   a   b   d   e  m1 m10 m11 m12  m2  m3  m4  m5  m6  m7  m8  m9  s1
##  NA  82  85  97  95  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## s10 s11 s12  s2  s3  s4  s5  s6  s7  s8  s9
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
mean(round(100*means.lcq,digit=0),na.rm=T)
```

```
## [1] 89.75
```

```
# Only comprehension questions on experimental items
eq <- subset(quest, expt=="lc")
eq<-merge(eq,bias.data,by.x=("item"),by.y="items")
eq$correct <- as.numeric(as.character(eq$correct))
eq$coh<-ifelse(eq$cond%in%letters[1:2],1,-1)
eq$len<-ifelse(eq$cond%in%letters[c(1,4)],1,-1)

mq<-glmer(correct ~ coh+len+bias+coh:len+len:bias+coh:bias+(1|subj)+(1|item), data=eq,family
```

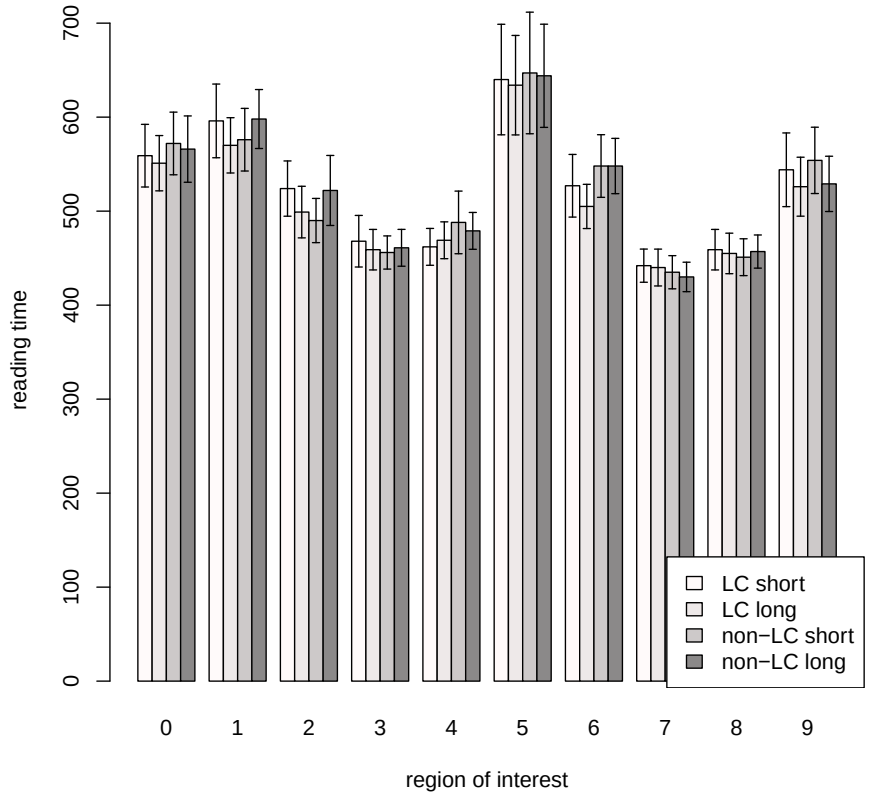
Locally coherent conditions have lower question-response accuracy:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.82	0.38	7.50	0.00
coh	-0.89	0.24	-3.69	0.00
len	0.09	0.23	0.41	0.68
bias	0.02	0.18	0.13	0.90
coh:len	-0.22	0.23	-0.96	0.34
len:bias	0.19	0.14	1.34	0.18
coh:bias	-0.03	0.17	-0.16	0.87

```
# Calculate odds based on model coefficients
odds <- exp(2.83+0.9)
(prob <- odds/(1+odds))

## [1] 0.9765693

odds2 <- exp(2.83-0.9)
(prob2 <- odds2/(1+odds2))

## [1] 0.8732494

## didn't work for SV without warnings:
#data<-drop.levels(subset(data,substr(pos,0,1) != "?"))

lcdata<-subset(data,expt=="lc")

#factor(lcdata£expt)

#unique(lcdata£pos)

## eliminate all non-word rows:
lcdata<-subset(lcdata,pos%in%c(0:20))

data<-lcdata

# Generate a column with reading times for the previous word (there is probably an easier w
data$pos<-as.numeric(as.character(data$pos))
datacopy<-data
datacopy$pos<-datacopy$pos+1
datacopy<-subset(datacopy, select=c(1,2,3,4,5,9))
colnames(datacopy)[6]<-"prev"
datacopy$prev<-scale(datacopy$prev,scale=FALSE)
data<-merge(data,datacopy,by.x=c("subj","expt","item","cond","pos"),by.y=c("subj","expt","it
```

```
#data<-drop.levels((subset(data,expt=="lc")))
critdata<-subset(data,roi%in%c("0","1","2","3","4","5","6","7","8","9"))

critdata2<-merge(critdata,bias.data,by.x=("item"),by.y="items")
## Did not work for SV:
#critdata<-gdata.drop.levels(critdata2)

critdata2$cond<-factor(critdata2$cond)
critdata2$expt<-factor(critdata2$expt)
critdata2$subj<-factor(critdata2$subj)
critdata2$item<-factor(critdata2$item)
critdata2$pos<-factor(critdata2$pos)

#summary(critdata2)

critdata<-critdata2

# Contrast coding
critdata$coh<-ifelse(critdata$cond%in%letters[1:2],1,-1)
critdata$len<-ifelse(critdata$cond%in%letters[c(1,4)],1,-1)

# Nested contrasts
critdata$c1<-ifelse(critdata$cond=="a",1,ifelse(critdata$cond=="b",-1,0))
critdata$c2<-ifelse(critdata$cond=="d",1,ifelse(critdata$cond=="e",-1,0))
critdata$c3<-ifelse(critdata$cond%in%c("a","b"),1,-1)

critdata<-subset(critdata,roi%in%c(0:9))

critdata$roi<-factor(critdata$roi)

summary(critdata$roi)

##    0    1    2    3    4    5    6    7    8    9
## 1280 1280 1280 1280 1280 1280 1280 1280 1280 1280
```

```
# Make a barplot of reading times
library(gplots)
std.error<-function(x){sd(x)/sqrt(length(x))}

means<-round(with(critdata,tapply(rt,IND=list(cond,roi),mean)),digits=0)
lowc<-means-1.96*round(with(critdata,tapply(rt,IND=list(cond,roi),std.error)),digits=0)
highc<-means+1.96*round(with(critdata,tapply(rt,IND=list(cond,roi),std.error)),digits=0)
barplot2(means, plot.ci = TRUE,col=c("snow1","snow2","snow3","snow4"),ci.l=lowc,ci.u=highc,b
legend("bottomright", legend=c("LC short","LC long","non-LC short","non-LC long"),fill=c("sn
```




```
critdata$Condition <- ifelse(critdata$cond=="a","LC short",ifelse(critdata$cond=="b","LC lon
```

	0	1	2	3	4	5	6	7	8	9
LC long	551.00	570.00	499.00	459.00	469.00	634.00	505.00	440.00	455.00	526.00
LC short	559.00	596.00	524.00	468.00	462.00	640.00	527.00	442.00	459.00	544.00
non-LC long	566.00	598.00	522.00	461.00	479.00	644.00	548.00	430.00	457.00	529.00
non-LC short	572.00	576.00	490.00	456.00	488.00	647.00	548.00	435.00	451.00	554.00

```
#round(xtabs(rt ~ Condition + roi, stats:::aggregate.formula(rt ~ Condition + roi,critdata,s
#xtabs(rt ~ Condition + roi,critdata)

# Fit the models
## roi 2:
#m2<-lmer(rrt ~ coh + len + bias +
```

```
#                   coh:len + coh:bias + len:bias +
#                   (1|subj) + (0+coh|subj) + (0+len|subj) +
#                   (0+bias|subj) + (1|item) + (0+coh|item) +
#                   (0+len|item), subset(critdata,
#                                        roi==2 & rrt > -6))

m2<-lmer(rrt ~ coh + len + bias +
                  coh:len + coh:bias + len:bias +
                  coh:len:bias+
                  (len +
                  coh:len + coh:bias ||subj)+
                  (1|item),
         subset(critdata,roi==2 & rrt > -6))

summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## rrt ~ coh + len + bias + coh:len + coh:bias + len:bias + coh:len:bias +
##     ((1 | subj) + (0 + len | subj) + (0 + len:coh | subj) + (0 +
##         coh:bias | subj)) + (1 | item)
##    Data: subset(critdata, roi == 2 & rrt > -6)
##
## REML criterion at convergence: 2094.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4256 -0.5531 -0.0639  0.5101  4.1584
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept) 0.250531 0.50053
##  subj.1   len         0.000619 0.02488
##  subj.2   len:coh     0.002950 0.05432
##  subj.3   coh:bias    0.002683 0.05180
##  item     (Intercept) 0.022723 0.15074
##  Residual             0.245294 0.49527
## Number of obs: 1277, groups:  subj, 40; item, 32
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)  -2.2513110  0.0846695 -26.589
## coh           0.0177283  0.0138607   1.279
## len          -0.0007763  0.0145095  -0.054
## bias         -0.0079712  0.0222743  -0.358
## coh:len       0.0353225  0.0163057   2.166
```

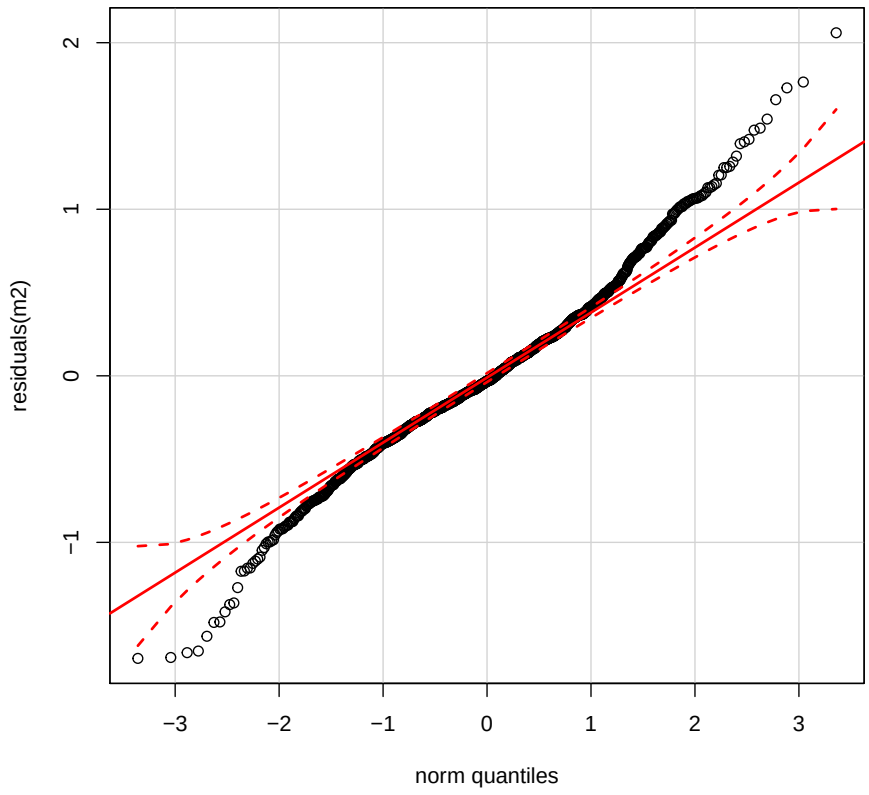
```
## coh:bias      0.0078377  0.0132789   0.590
## len:bias      0.0213262  0.0103249   2.066
## coh:len:bias  0.0014328  0.0110945   0.129
##
## Correlation of Fixed Effects:
##             (Intr) coh    len    bias   coh:ln coh:bs len:bs
## coh          0.000
## len          0.000  0.002
## bias         0.000 -0.001 -0.002
## coh:len      0.000  0.001  0.001  0.000
## coh:bias     0.000  0.000  0.000  0.000 -0.001
## len:bias     0.000  0.000  0.005  0.000 -0.001  0.001
## coh:len:bis  0.000 -0.001 -0.001  0.000  0.001 -0.003  0.000

## coh:len removed from this model:
m2cohlen<-lmer(rrt ~ coh + len + bias +
                  coh:bias + len:bias +
                 coh:len:bias+
                 (len + coh:len+
                 coh:bias ||subj)+
                 (1|item),
         subset(critdata,roi==2 & rrt > -6))

## coh:bias removed:
m2cohbias<-lmer(rrt ~ coh + len + bias +
                 coh:len + len:bias +
                 coh:len:bias+
                 (len +
                 coh:len + coh:bias ||subj)+
                 (1|item),
         subset(critdata,roi==2 & rrt > -6))

r2anova<-anova(m2,m2cohlen)

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

r2chisq<-c(r2anova$Chisq[2],
           r2anova$"Pr(>Chisq)"[2])

r2anovacohbias<-anova(m2,m2cohbias)

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

r2chisq<-c(r2anova$Chisq[2],
           r2anova$"Pr(>Chisq)"[2])
```

At region r2, the interaction coh:len was significant $\chi^2_1 = 4.54, p = 0.03$. The

coh:bias did not reach significant.

Table for paper:

	Estimate	Std. Error	t value
(Intercept)	-2.25	0.08	-26.59
coh	0.02	0.01	1.28
len	-0.00	0.01	-0.05
bias	-0.01	0.02	-0.36
coh:len	0.04	0.02	2.17
coh:bias	0.01	0.01	0.59
len:bias	0.02	0.01	2.07
coh:len:bias	0.00	0.01	0.13

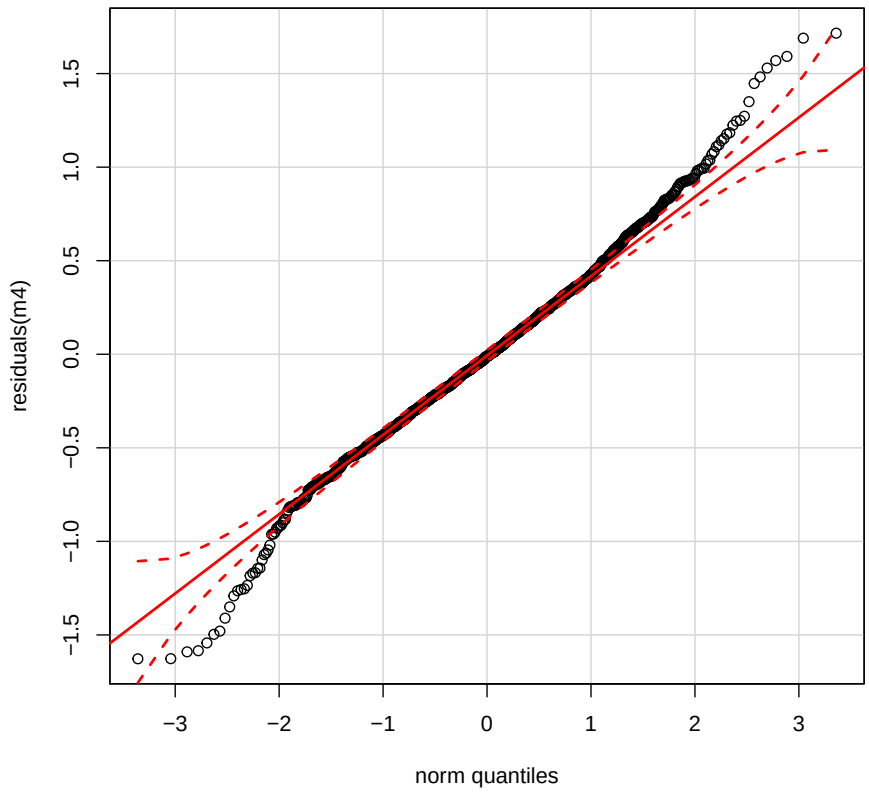



```
#(m4<-lmer(rrt ~ (coh + len + bias)^2 + (1|subj) + (0+coh|subj) + (0+len|subj) + (0+bias|sub

m4<-lmer(rrt ~ coh + len + bias +
```

```
                  coh:len + coh:bias + len:bias +
                  coh:len:bias+
                  (coh:len + coh:bias||subj)+
                  (coh:len ||item),
         subset(critdata,roi==4 & rrt > -6))

m4coh<-lmer(rrt ~ len + bias +
                  coh:len + coh:bias + len:bias +
                  coh:len:bias+
                  (coh:len + coh:bias||subj)+
                  (coh:len ||item),
         subset(critdata,roi==4 & rrt > -6))

r4anova<-anova(m4,m4coh)

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

r4chisq<-c(r4anova$Chisq[2],
           r4anova$"Pr(>Chisq)"[2])
```

At region r4, the coherence effect was significant $\chi^2_1 = 4.43, p = 0.04$.

	Estimate	Std. Error	t value
(Intercept)	-2.37	0.10	-24.26
coh	-0.03	0.01	-2.10
len	-0.02	0.01	-1.61
bias	-0.01	0.02	-0.42
coh:len	-0.00	0.02	-0.28
coh:bias	-0.00	0.01	-0.24
len:bias	0.02	0.01	1.62
coh:len:bias	0.01	0.01	0.57

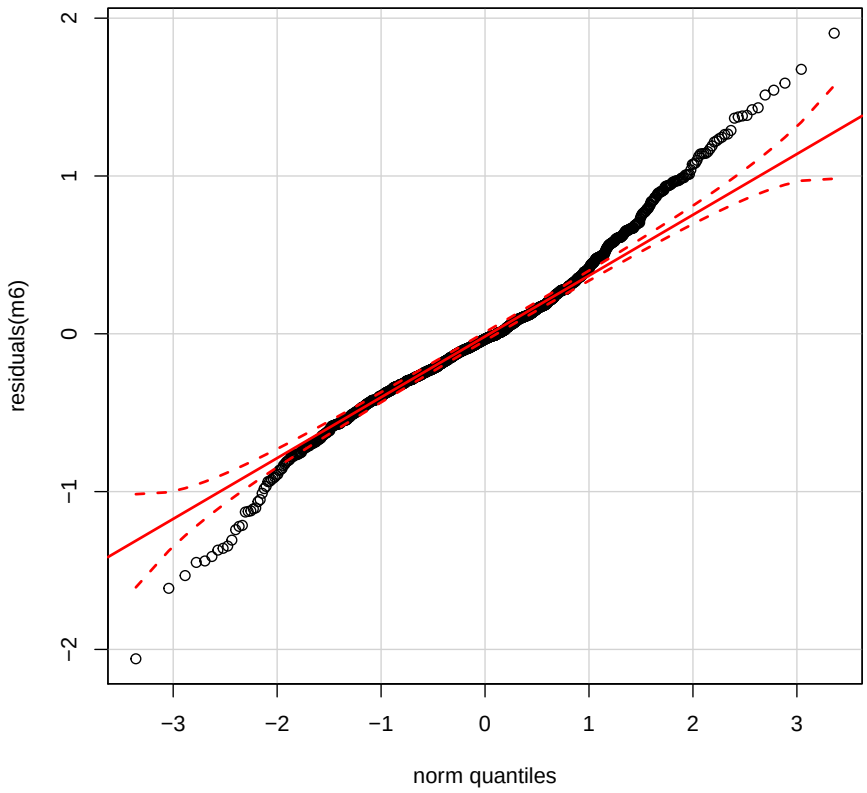



```
# With spillover predictor, for region 6
#(m6<-lmer(rrt ~ (coh + len + bias)^2 + prev + (1|subj) + (0+coh|subj) + (0+len|subj) + (0+

m6<-lmer(rrt ~ coh + len + bias +
                  coh:len + coh:bias + len:bias +
                  coh:len:bias + prev+
                  (prev+coh:len + coh:len:bias||subj)+
                  (len +
                  coh:len ||item),
         subset(critdata,roi==6 & rrt > -6))

m6coh<-lmer(rrt ~ len + bias +
                  coh:len + coh:bias + len:bias +
                  coh:len:bias +
                  (coh:len + coh:len:bias||subj)+
                  (len +
```

```
                  coh:len ||item),
         subset(critdata,roi==6 & rrt > -6))

r6anova<-anova(m6,m6coh)

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

r6chisq<-c(r6anova$Chisq[2],
           r6anova$"Pr(>Chisq)"[2])
```

At region r6, the coherence effect was significant $\chi^2_1 = 104.19, p = 0$.

	Estimate	Std. Error	t value
(Intercept)	-2.11	0.06	-34.81
coh	-0.05	0.01	-3.38
len	0.00	0.02	0.02
bias	0.00	0.02	0.07
prev	0.11	0.04	2.53
coh:len	-0.00	0.02	-0.12
coh:bias	-0.01	0.01	-0.70
len:bias	-0.00	0.01	-0.03
coh:len:bias	-0.00	0.01	-0.20




```
((dim(subset(critdata, roi == 0))-dim(subset(critdata, roi == 0 & rrt > -6)))/dim(subset(cri

## [1] 0.078125 0.000000

# Fit with nested contrasts
#summary(m2a<-lmer(rrt~c1+c2+c3+(c1+c2||subj)+(c3||item),subset(critdata,roi==2 & rrt > -6),
```

2 Stan models

2.1 Region 2

```
library(rstan)

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

```
##
## Attaching package:  'inline'
##
## The following object is masked from 'package:Rcpp':
##
##    registerPlugin
##
## rstan (Version 2.7.0-1, packaged:  2015-07-17 18:12:01 UTC, GitRev:
05c3d0058b6a)
## For execution on a local, multicore CPU with excess RAM we recommend
calling
## rstan_options(auto_write = TRUE)
## options(mc.cores = parallel::detectCores())
```

```
r2<-subset(critdata,roi==2 & rrt>-6)
r4<-subset(critdata,roi==4 & rrt>-6)
r6<-subset(critdata,roi==6 & rrt>-6)

datr2 <- list(mu_prior=c(0,0,0,0,0,0,0,0),
            subject=as.integer(factor(r2$subj)),
            item=as.integer(factor(r2$item)),
            y=r2$rrt, ## dep. var.
            coh = r2$coh,
            len = r2$len,
            bias = r2$bias,
            coh_len = r2$coh*r2$len,
            coh_bias = r2$coh*r2$bias,
            len_bias = r2$len*r2$bias,
            coh_len_bias = r2$coh*r2$len*r2$bias,
            N = nrow(r2),
            I = length(unique(r2$subj)),
            K = length(unique(r2$item))
)

params<-c("beta","sigma_e","sigma_u","sigma_w")

fit_r2 <- stan(file="PaapeVasishthLC2.Stan",
                 iter=2000,
                 warmup=500,
                 data=datr2,
                 pars=params)

## COMPILING THE C++ CODE FOR MODEL 'PaapeVasishthLC2' NOW.
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 1).
```

```
##
## Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 1, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 1, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 1, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 42.044 seconds (Warm-up)
## #                36.5262 seconds (Sampling)
## #                78.5702 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 2, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 2, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 2, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 2, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 2, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 2, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 2, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 2, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 44.2032 seconds (Warm-up)
## #                36.1666 seconds (Sampling)
## #                80.3698 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 3).
##
## Chain 3, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 3, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 3, Iteration:  900 / 2000 [ 45%]  (Sampling)
```

```
## Chain 3, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 3, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 3, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 3, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 3, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 3, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 48.6701 seconds (Warm-up)
## #                35.785 seconds (Sampling)
## #                84.4551 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 4).
##
## Chain 4, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 4, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 4, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 4, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 4, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 4, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 4, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 4, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 4, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 46.916 seconds (Warm-up)
## #                34.9542 seconds (Sampling)
## #                81.8702 seconds (Total)
```

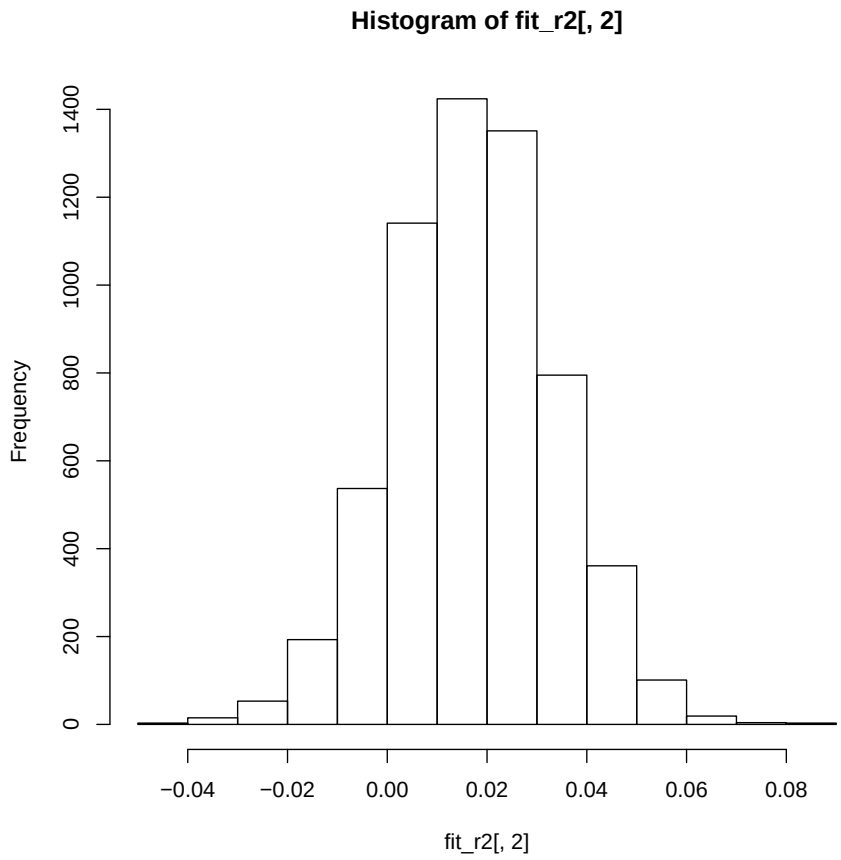
```
print(fit_r2)
```

```
## Inference for Stan model: PaapeVasishthLC2.
## 4 chains, each with iter=2000; warmup=500; thin=1;
## post-warmup draws per chain=1500, total post-warmup draws=6000.
##
##              mean se_mean    sd   2.5%    25%    50%   75% 97.5% n_eff
## beta[1]     -2.24     0.0  0.09  -2.41  -2.30  -2.24 -2.19 -2.07   508
## beta[2]      0.02     0.0  0.02  -0.01   0.01   0.02  0.03  0.05  6000
## beta[3]      0.00     0.0  0.02  -0.03  -0.01   0.00  0.01  0.03  6000
## beta[4]     -0.01     0.0  0.02  -0.06  -0.02  -0.01  0.01  0.04  2205
## beta[5]      0.03     0.0  0.02   0.00   0.02   0.03  0.05  0.07  6000
## beta[6]      0.01     0.0  0.02  -0.02   0.00   0.01  0.02  0.04  6000
## beta[7]      0.02     0.0  0.01   0.00   0.01   0.02  0.03  0.04  6000
## beta[8]      0.00     0.0  0.01  -0.02  -0.01   0.00  0.01  0.02  6000
## sigma_e      0.49     0.0  0.01   0.47   0.49   0.49  0.50  0.52  6000
## sigma_u[1]   0.52     0.0  0.06   0.41   0.48   0.51  0.56  0.66  1091
```

```
## sigma_u[2]   0.02     0.0  0.01   0.00   0.01   0.02  0.03  0.05  4342
## sigma_u[3]   0.03     0.0  0.02   0.00   0.02   0.03  0.05  0.08  2241
## sigma_u[4]   0.02     0.0  0.01   0.00   0.01   0.01  0.02  0.05  3145
## sigma_u[5]   0.05     0.0  0.02   0.01   0.04   0.05  0.07  0.10  1493
## sigma_u[6]   0.05     0.0  0.02   0.01   0.04   0.05  0.06  0.09  1255
## sigma_u[7]   0.02     0.0  0.01   0.00   0.01   0.02  0.03  0.05  2676
## sigma_u[8]   0.02     0.0  0.01   0.00   0.01   0.02  0.03  0.05  2842
## sigma_w[1]   0.16     0.0  0.03   0.11   0.14   0.16  0.18  0.22  2335
## sigma_w[2]   0.04     0.0  0.02   0.00   0.02   0.04  0.06  0.09  1771
## sigma_w[3]   0.02     0.0  0.01   0.00   0.01   0.02  0.03  0.05  3744
## sigma_w[4]   0.02     0.0  0.02   0.00   0.01   0.02  0.03  0.06  2935
## lp__       -13.92     0.6 19.96 -52.84 -27.34 -14.02 -0.38 25.64  1113
##            Rhat
## beta[1]       1
## beta[2]       1
## beta[3]       1
## beta[4]       1
## beta[5]       1
## beta[6]       1
## beta[7]       1
## beta[8]       1
## sigma_e       1
## sigma_u[1]    1
## sigma_u[2]    1
## sigma_u[3]    1
## sigma_u[4]    1
## sigma_u[5]    1
## sigma_u[6]    1
## sigma_u[7]    1
## sigma_u[8]    1
## sigma_w[1]    1
## sigma_w[2]    1
## sigma_w[3]    1
## sigma_w[4]    1
## lp__          1
##
## Samples were drawn using NUTS(diag_e) at Wed Sep  9 12:46:39 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
fit_r2<-as.matrix(fit_r2)

#            coh = r2£coh,
```

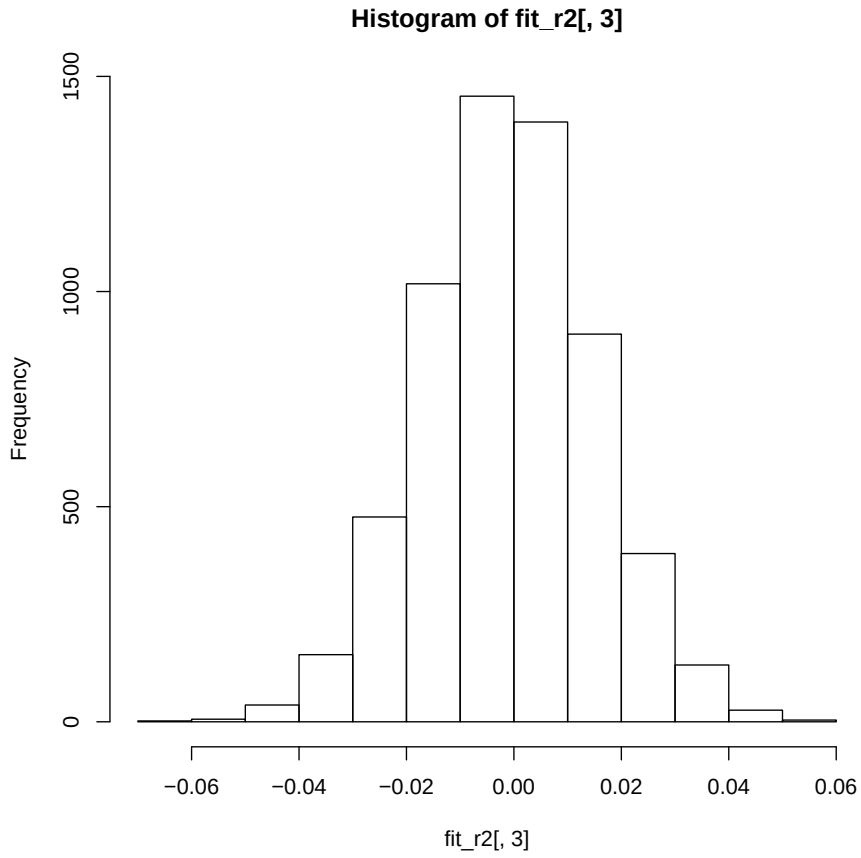
```r
hist(fit_r2[,2])
```




```r
mean(fit_r2[,2]>0)
```

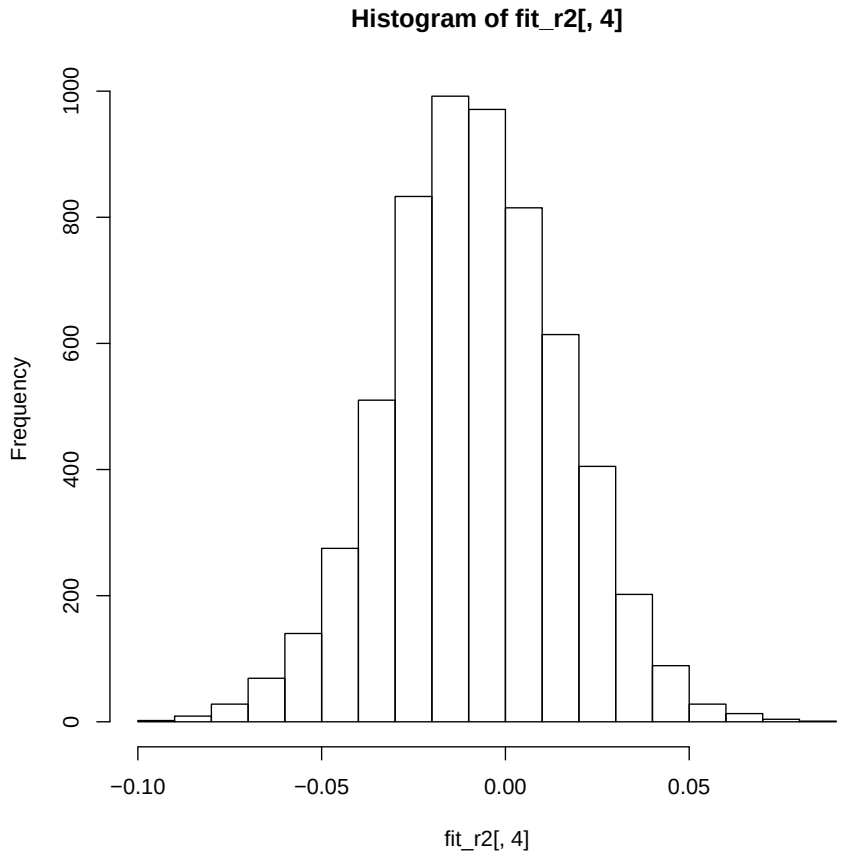
```
## [1] 0.8665
```

```r
#             len = r2$len,
hist(fit_r2[,3])
```



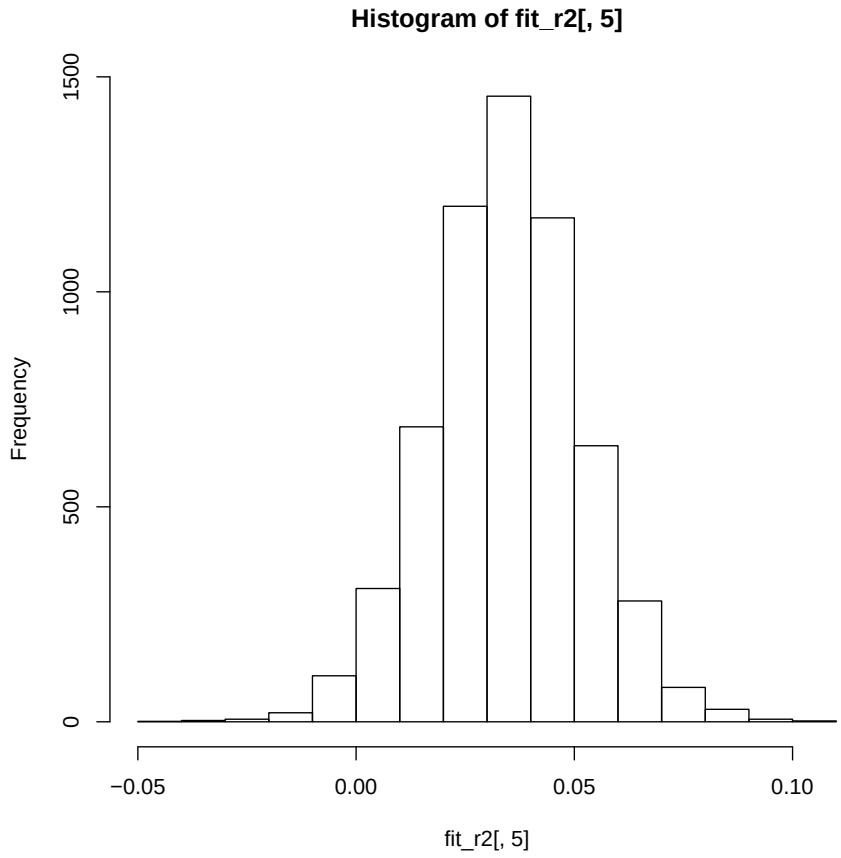

```
mean(fit_r2[,3]>0)

## [1] 0.4748333
```

```
#             bias = r2fbias,
hist(fit_r2[,4])
```

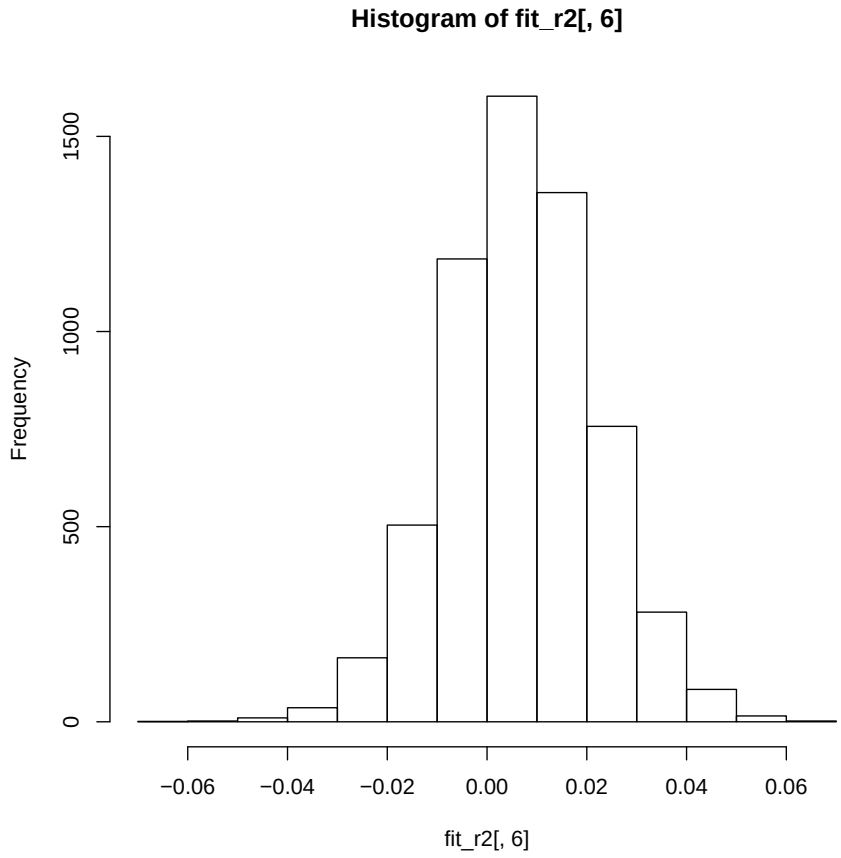



```
mean(fit_r2[,4]>0)

## [1] 0.3618333

quantiles(fit_r2[,4],probs=c(0.025,0.975))

## Error in eval(expr, envir, enclos):  could not find function "quantiles"
```

```
#            coh_len = r2fcoh*r2flen,
hist(fit_r2[,5])
```

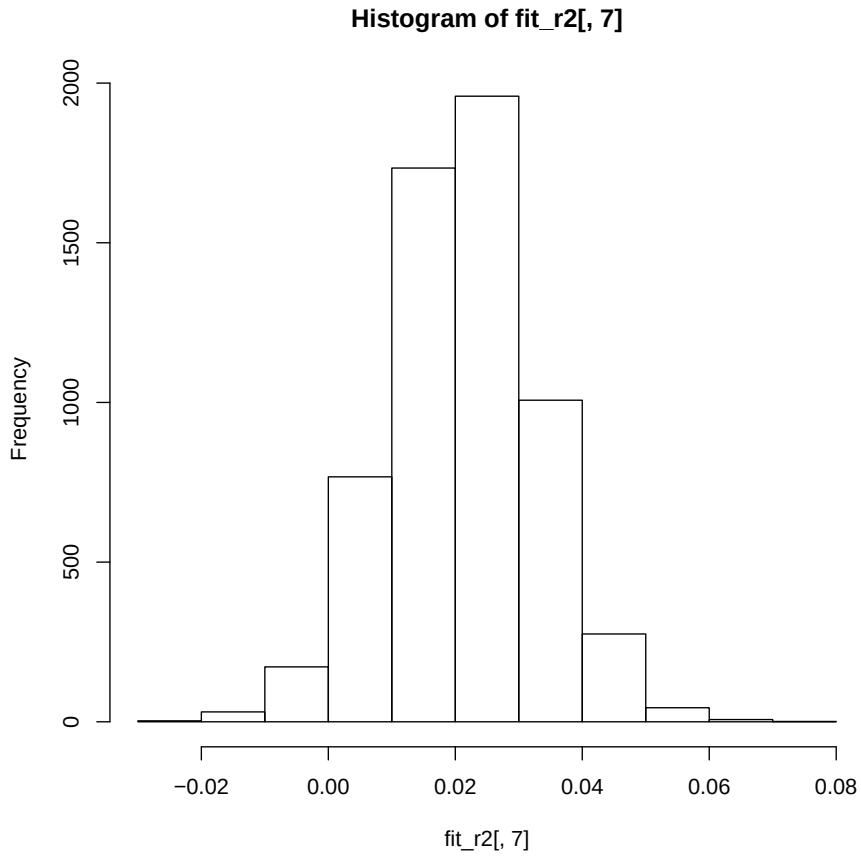



```
r2cohlenHPD<-mean(fit_r2[,5]>0)
#             coh_bias = r2£coh*r2£bias,
hist(fit_r2[,6])
```



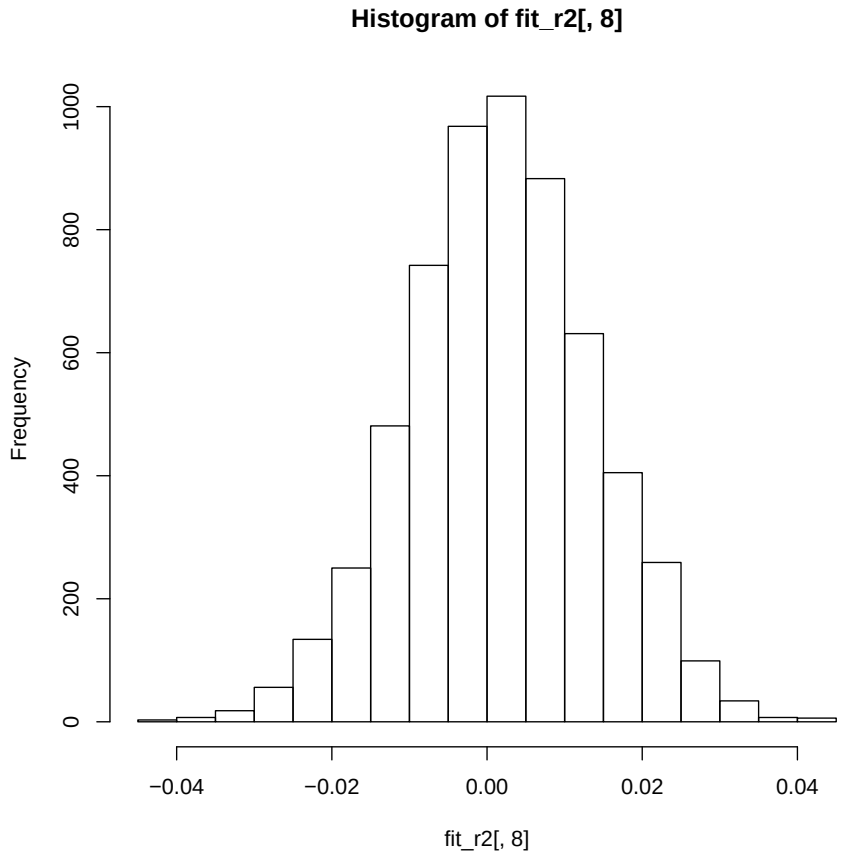

```
mean(fit_r2[,6]>0)

## [1] 0.6828333
```

```
#            len_bias = r2$len*r2$bias,
hist(fit_r2[,7])
```




```
mean(fit_r2[,7]>0)

## [1] 0.9656667
```

```
#            coh_len_bias = r2fcoh*r2flen*r2fbias,
hist(fit_r2[,8])
```




```
mean(fit_r2[,8]>0)

## [1] 0.5568333
```

2.2 Region 4

```
datr4 <- list(mu_prior=c(0,0,0,0,0,0,0,0),
              subject=as.integer(factor(r4$subj)),
              item=as.integer(factor(r4$item)),
              y=r4$rrt, ## dep. var.
              coh = r4$coh,
              len = r4$len,
              bias = r4$bias,
              coh_len = r4$coh*r4$len,
```

```
            coh_bias = r4$coh*r4$bias,
            len_bias = r4$len*r4$bias,
            coh_len_bias = r4$coh*r4$len*r4$bias,
            N = nrow(r4),
            I = length(unique(r4$subj)),
            K = length(unique(r4$item))
)
```

```
fit_r4 <- stan(file="PaapeVasishthLC2.Stan",
                iter=2000,
                warmup=500,
                data=datr4,
                pars=params)

## COMPILING THE C++ CODE FOR MODEL 'PaapeVasishthLC2' NOW.
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 1, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 1, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 1, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 48.2362 seconds (Warm-up)
## #                68.2352 seconds (Sampling)
## #                116.471 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 2, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 2, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 2, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 2, Iteration: 1300 / 2000 [ 65%]  (Sampling)
```

```
## Chain 2, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 2, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 2, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 2, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 46.9677 seconds (Warm-up)
## #                34.8261 seconds (Sampling)
## #                81.7938 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 3).
##
## Chain 3, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 3, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 3, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 3, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 3, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 3, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 3, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 3, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 3, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 39.7022 seconds (Warm-up)
## #                34.523 seconds (Sampling)
## #                74.2253 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC2' NOW (CHAIN 4).
##
## Chain 4, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 4, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 4, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 4, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 4, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 4, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 4, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 4, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 4, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 42.1685 seconds (Warm-up)
## #                34.4739 seconds (Sampling)
## #                76.6423 seconds (Total)
```

```
print(fit_r4)
```

```
## Inference for Stan model: PaapeVasishthLC2.
## 4 chains, each with iter=2000; warmup=500; thin=1;
## post-warmup draws per chain=1500, total post-warmup draws=6000.
##
##             mean se_mean    sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta[1]    -2.38    0.00  0.10  -2.56 -2.44 -2.38 -2.31 -2.18   533 1.00
## beta[2]    -0.03    0.00  0.02  -0.06 -0.04 -0.03 -0.02  0.00  6000 1.00
## beta[3]    -0.02    0.00  0.01  -0.05 -0.03 -0.02 -0.01  0.01  6000 1.00
## beta[4]    -0.01    0.00  0.03  -0.06 -0.03 -0.01  0.01  0.04  2091 1.00
## beta[5]     0.00    0.00  0.02  -0.04 -0.02  0.00  0.01  0.03  6000 1.00
## beta[6]     0.00    0.00  0.02  -0.03 -0.01  0.00  0.01  0.03  5089 1.00
## beta[7]     0.02    0.00  0.01  -0.01  0.01  0.02  0.02  0.04  6000 1.00
## beta[8]     0.01    0.00  0.01  -0.02  0.00  0.01  0.02  0.03  6000 1.00
## sigma_e     0.48    0.00  0.01   0.46  0.48  0.48  0.49  0.50  6000 1.00
## sigma_u[1]  0.60    0.00  0.07   0.48  0.55  0.60  0.65  0.77   820 1.01
## sigma_u[2]  0.03    0.00  0.02   0.00  0.01  0.02  0.04  0.07  2437 1.00
## sigma_u[3]  0.02    0.00  0.01   0.00  0.01  0.02  0.03  0.05  3366 1.00
## sigma_u[4]  0.02    0.00  0.01   0.00  0.01  0.01  0.02  0.04  2829 1.00
## sigma_u[5]  0.05    0.00  0.02   0.00  0.03  0.05  0.06  0.09  1319 1.00
## sigma_u[6]  0.07    0.00  0.02   0.04  0.06  0.07  0.08  0.10  2812 1.00
## sigma_u[7]  0.02    0.00  0.01   0.00  0.01  0.02  0.03  0.05  2762 1.00
## sigma_u[8]  0.02    0.00  0.01   0.00  0.01  0.01  0.02  0.04  3297 1.00
## sigma_w[1]  0.18    0.00  0.03   0.13  0.16  0.18  0.20  0.25  2144 1.00
## sigma_w[2]  0.02    0.00  0.02   0.00  0.01  0.02  0.03  0.06  3031 1.00
## sigma_w[3]  0.02    0.00  0.02   0.00  0.01  0.02  0.03  0.06  3411 1.00
## sigma_w[4]  0.03    0.00  0.02   0.00  0.02  0.03  0.05  0.07  2274 1.00
## lp__       14.65    0.53 19.69 -25.37  1.42 15.06 27.95 53.17  1406 1.00
##
## Samples were drawn using NUTS(diag_e) at Wed Sep  9 12:52:53 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
#fit_r4<-as.matrix(fit_r4)
```

2.3 Region 6

```
datr6 <- list(mu_prior=c(0,0,0,0,0,0,0,0),
              subject=as.integer(factor(r6$subj)),
              item=as.integer(factor(r6$item)),
              y=r6$rrt, ## dep. var.
              coh = r6$coh,
```

```
            len = r6$len,
            bias = r6$bias,
            coh_len = r6$coh*r6$len,
            coh_bias = r6$coh*r6$bias,
            len_bias = r6$len*r6$bias,
            coh_len_bias = r6$coh*r6$len*r6$bias,
            N = nrow(r6),
            I = length(unique(r6$subj)),
            K = length(unique(r6$item))
)
```

```
fit_r6 <- stan(file="PaapeVasishthLC.Stan",
                iter=2000,
                warmup=500,
                data=datr6,
                pars=params)

## COMPILING THE C++ CODE FOR MODEL 'PaapeVasishthLC' NOW.
##
## SAMPLING FOR MODEL 'PaapeVasishthLC' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 1, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 1, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 1, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 56.7654 seconds (Warm-up)
## #                41.5824 seconds (Sampling)
## #                98.3478 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 2, Iteration:  700 / 2000 [ 35%]  (Sampling)
```

```
## Chain 2, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 2, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 2, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 2, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 2, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 2, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 2, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 48.9656 seconds (Warm-up)
## #                42.021 seconds (Sampling)
## #                90.9866 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC' NOW (CHAIN 3).
##
## Chain 3, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 3, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 3, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 3, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 3, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 3, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 3, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 3, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 3, Iteration: 2000 / 2000 [100%]  (Sampling)
## #  Elapsed Time: 53.7938 seconds (Warm-up)
## #                42.315 seconds (Sampling)
## #                96.1089 seconds (Total)
##
##
## SAMPLING FOR MODEL 'PaapeVasishthLC' NOW (CHAIN 4).
##
## Chain 4, Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4, Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4, Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4, Iteration:  501 / 2000 [ 25%]  (Sampling)
## Chain 4, Iteration:  700 / 2000 [ 35%]  (Sampling)
## Chain 4, Iteration:  900 / 2000 [ 45%]  (Sampling)
## Chain 4, Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 4, Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 4, Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 4, Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 4, Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 4, Iteration: 2000 / 2000 [100%]  (Sampling)
```

```
## #  Elapsed Time: 55.2205 seconds (Warm-up)
## #                42.4696 seconds (Sampling)
## #                97.6902 seconds (Total)
```

```
print(fit_r6)
```

```
## Inference for Stan model: PaapeVasishthLC.
## 4 chains, each with iter=2000; warmup=500; thin=1;
## post-warmup draws per chain=1500, total post-warmup draws=6000.
##
##               mean se_mean    sd    2.5%     25%     50%     75%  97.5%
## beta[1]      -2.15    0.00  0.08   -2.31   -2.20   -2.15   -2.10  -1.99
## beta[2]      -0.04    0.00  0.02   -0.08   -0.06   -0.04   -0.03  -0.01
## beta[3]      -0.01    0.00  0.02   -0.04   -0.02   -0.01    0.01   0.03
## beta[4]       0.00    0.00  0.02   -0.04   -0.01    0.00    0.01   0.03
## beta[5]       0.01    0.00  0.02   -0.03   -0.01    0.01    0.02   0.05
## beta[6]       0.00    0.00  0.02   -0.03   -0.01    0.00    0.01   0.03
## beta[7]       0.00    0.00  0.02   -0.03   -0.01    0.00    0.01   0.03
## beta[8]       0.00    0.00  0.02   -0.04   -0.01    0.00    0.01   0.03
## sigma_e       0.51    0.00  0.01    0.49    0.50    0.51    0.52   0.53
## sigma_u[1]    0.47    0.00  0.06    0.38    0.43    0.47    0.50   0.59
## sigma_u[2]    0.03    0.00  0.02    0.00    0.01    0.03    0.04   0.07
## sigma_u[3]    0.03    0.00  0.02    0.00    0.01    0.03    0.04   0.07
## sigma_u[4]    0.02    0.00  0.01    0.00    0.01    0.01    0.02   0.05
## sigma_u[5]    0.05    0.00  0.02    0.00    0.03    0.05    0.06   0.10
## sigma_u[6]    0.02    0.00  0.01    0.00    0.01    0.02    0.03   0.05
## sigma_u[7]    0.02    0.00  0.01    0.00    0.01    0.02    0.03   0.05
## sigma_u[8]    0.02    0.00  0.02    0.00    0.01    0.02    0.03   0.06
## sigma_w[1]    0.08    0.00  0.03    0.01    0.06    0.08    0.10   0.13
## sigma_w[2]    0.03    0.00  0.02    0.00    0.02    0.03    0.05   0.08
## sigma_w[3]    0.03    0.00  0.02    0.00    0.01    0.03    0.05   0.08
## sigma_w[4]    0.03    0.00  0.03    0.00    0.01    0.03    0.05   0.10
## sigma_w[5]    0.07    0.00  0.03    0.01    0.05    0.07    0.09   0.12
## sigma_w[6]    0.02    0.00  0.02    0.00    0.01    0.02    0.03   0.07
## sigma_w[7]    0.03    0.00  0.02    0.00    0.01    0.03    0.04   0.07
## sigma_w[8]    0.03    0.00  0.02    0.00    0.01    0.02    0.04   0.09
## lp__       -141.43    0.59 22.39 -186.62 -156.35 -140.96 -126.15 -98.53
##            n_eff Rhat
## beta[1]      366 1.01
## beta[2]     6000 1.00
## beta[3]     4643 1.00
## beta[4]     2587 1.00
## beta[5]     4655 1.00
## beta[6]     4128 1.00
## beta[7]     4424 1.00
## beta[8]     3712 1.00
```

```
## sigma_e     6000 1.00
## sigma_u[1]   712 1.00
## sigma_u[2]  1874 1.00
## sigma_u[3]  2342 1.00
## sigma_u[4]  2630 1.00
## sigma_u[5]  1279 1.00
## sigma_u[6]  2469 1.00
## sigma_u[7]  2600 1.00
## sigma_u[8]  2020 1.00
## sigma_w[1]   617 1.01
## sigma_w[2]  1770 1.00
## sigma_w[3]  1705 1.00
## sigma_w[4]   790 1.00
## sigma_w[5]   653 1.01
## sigma_w[6]  2099 1.00
## sigma_w[7]  2068 1.00
## sigma_w[8]   994 1.00
## lp__        1459 1.00
##
## Samples were drawn using NUTS(diag_e) at Wed Sep  9 12:59:40 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```