Two-stage exams: Study 1

George Kinnear

20/06/2020

Contents

Data

Import and combine the datasets.

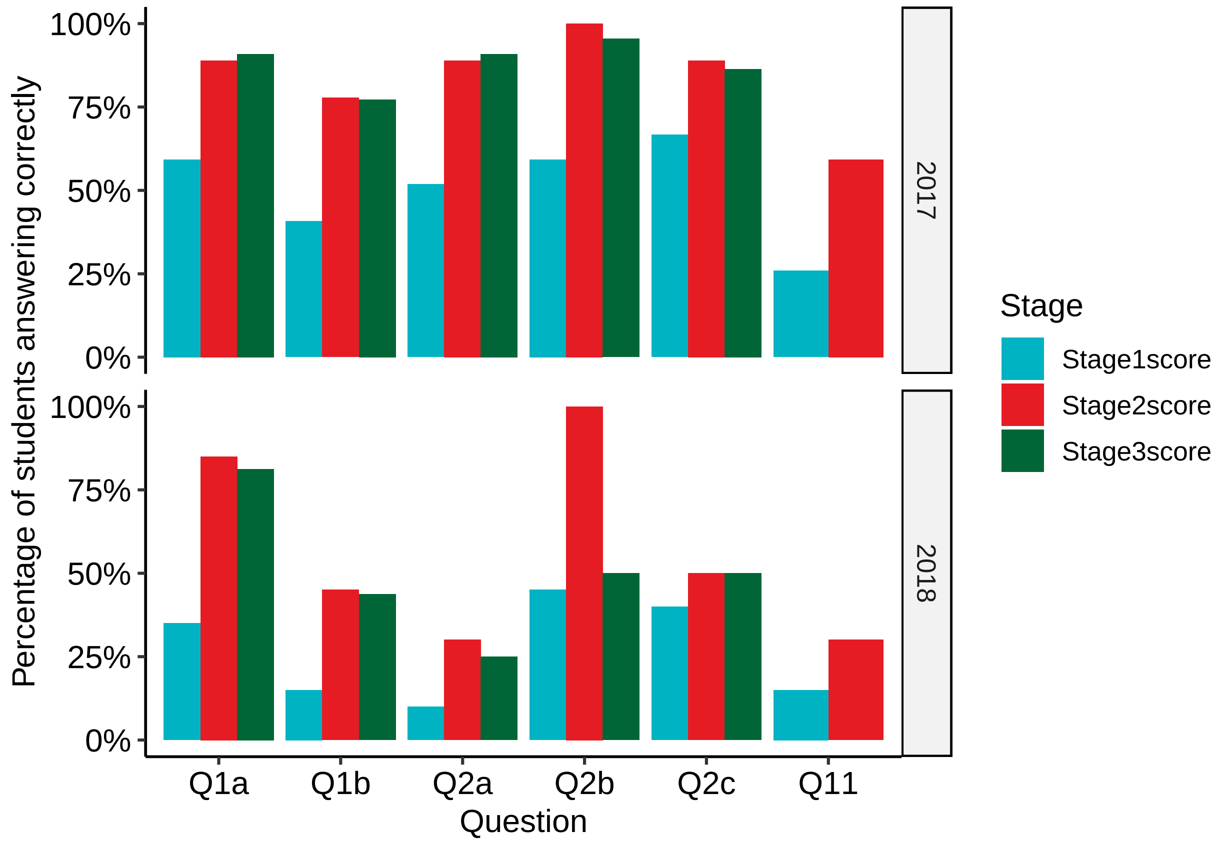
```
data2017 = read.csv('study1_2017.csv', header=T)
data2018 = read.csv('study1_2018.csv', header=T)

data1718 = bind_rows("2017"=data2017, "2018"=data2018, .id="Year") %>%
  select(-c(X)) %>%
  mutate(Q = fct_relevel(Q, c("Q1a", "Q1b", "Q2a", "Q2b", "Q2c", "Q11")))


S1234data = subset(data1718,
                   select=c('Q','Student','Stage1score','Stage2score','Stage3score','Stage4score')) %>%
  filter(!is.na(Stage2score))


S123data = subset(data1718,select=c('Q','Student','Stage1score','Stage2score','Stage3score','ZippGroup'
S123data = S123data[complete.cases(S123data),]
```

Check that it makes sense to combine the two datasets

Viewing both sets of data to check that the items performed similarly in both years.

```
ld <- gather(data = subset(data1718,select=c('Year','Q','Student','Stage1score','Stage2score','Stage3sco
             key = stage,
             value = score,
             Stage1score, Stage2score, Stage3score)
ld <- ld[complete.cases(ld),]
```

```r
# Code from here:
# https://www.r-bloggers.com/building-barplots-with-error-bars/

plotData <- aggregate(ld$score,
                      by = list(Year = ld$Year, Q = ld$Q, Stage = ld$stage),
                      FUN = function(x) c(mean = mean(x), sd = sd(x),
                                          n = length(x)))
plotData <- do.call(data.frame, plotData)
plotData$se <- plotData$x.sd / sqrt(plotData$x.n)

colnames(plotData) <- c("Year","Q", "Stage", "mean", "sd", "n", "se")

limits <- aes(ymax = plotData$mean + plotData$se,
              ymin = plotData$mean - plotData$se)

p <- ggplot(data = plotData, aes(x = factor(Q), y = mean,
                                 fill = factor(Stage))) +
  facet_grid(Year ~ .)+
  geom_bar(stat = "identity",
           position = position_dodge(0.9)) +
  labs(x = "Question",
       y = "Percentage of students answering correctly",
       fill = "Stage") +
  scale_fill_manual(values=heathers) + #viridis_pal(1,0,1)(4)) +
  scale_y_continuous(labels = scales::percent)
p
```



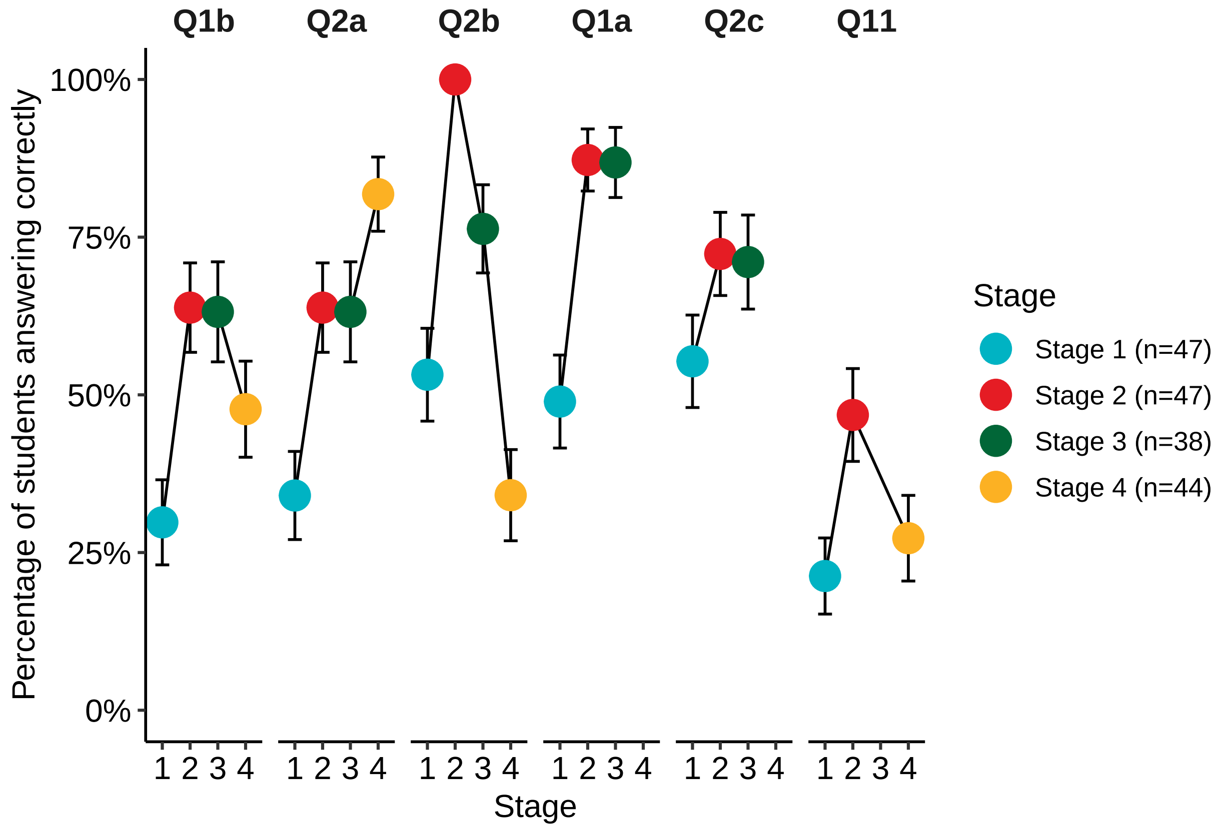
Mean at each stage (as bars)

```
ld <- gather(data = S1234data,
             key = stage,
             value = score,
             Stage1score, Stage2score, Stage3score, Stage4score)
ld <- ld[complete.cases(ld),]

plotData <- aggregate(ld$score,
                      by = list(Q = ld$Q, Stage = ld$stage),
                      FUN = function(x) c(mean = mean(x), sd = sd(x),
                                          n = length(x)))
plotData <- do.call(data.frame, plotData)
plotData$se <- plotData$x.sd / sqrt(plotData$x.n)

colnames(plotData) <- c("Q", "Stage", "mean", "sd", "n", "se")

plotData = plotData %>%
  mutate(
    Q = fct_relevel(Q, c( "Q1b", "Q2a", "Q2b","Q1a","Q2c",  "Q11")),
    Stage = gsub(".*(\\d).*","\\1",Stage)  # alternatively: Stage = parse_number(Stage)
  )
plotCounts = plotData %>% filter(Q=="Q1b") %>% group_by(Stage) %>% select(Stage,n) %>%
  mutate(scale_lab = paste0("Stage ",Stage, " (n=",n,")"))
```

```r
limits <- aes(ymax = plotData$mean + plotData$se,
              ymin = plotData$mean - plotData$se)

p <- ggplot(data = plotData, aes(x = factor(Q), y = mean,
                                 fill = factor(Stage), label=n)) +
  geom_bar(stat = "identity",
             position = position_dodge(0.9))+
  geom_errorbar(limits, position = position_dodge(0.9),
                width = 0.25)  +
  labs(x = "Question",
       y = "Percentage of students answering correctly",
       fill = "Stage") +
  scale_fill_manual(values=heathers, labels=plotCounts$scale_lab) +
  scale_y_continuous(labels = scales::percent)
ggsave("Figs/Study1_S1234_means.pdf",width=20,height=10,units="cm",dpi=300)
```

Mean at each stage (as points)

```r
ggplot(data=plotData,aes(x=Stage,y=mean,group=Q))+
  geom_line()+
  geom_errorbar(limits, position = position_dodge(0.9),
                width = 0.5)  +
  geom_point(position = position_dodge(0.9),size=5,aes(color=Stage))+
  scale_color_manual(values=heathers, labels=plotCounts$scale_lab) +
  labs(x = "Stage",
       y = "Percentage of students answering correctly",
       fill = "Stage") +
  scale_y_continuous(labels = scales::percent, limits=c(0,1))+
  facet_grid(cols=vars(Q))+
  theme(strip.background = element_rect(fill=NA,colour = NA),
        strip.text = element_text(size=12, face="bold"))
```




```
ggsave("Figs/Study1_S1234_means_pts.pdf",width=20,height=10,units="cm",dpi=300)
```

Mean at each stage (table, with standard errors)

```
tab = plotData %>%
  mutate(
    entry = paste0(sprintf("%2.0f", mean*100), " (", sprintf("%2.1f", se*100), ")"),
    Stage = gsub(".*(\\d).*","\\1",Stage)
  ) %>%
  group_by(Q,Stage) %>%
  select(Q,Stage,entry) %>%
  spread(Q,entry)

tab %>% knitr::kable()
```

Stage	Q1b	Q2a	Q2b	Q1a	Q2c	Q11
1	30 (6.7)	34 (7.0)	53 (7.4)	49 (7.4)	55 (7.3)	21 (6.0)
2	64 (7.1)	64 (7.1)	100 (0.0)	87 (4.9)	72 (6.6)	47 (7.4)
3	63 (7.9)	63 (7.9)	76 (7.0)	87 (5.6)	71 (7.5)	NA
4	48 (7.6)	82 (5.9)	34 (7.2)	NA	NA	27 (6.8)

```
#tab %>% knitr::kable(format="latex",booktabs=T)
```

Forming the Zipp tables

This constructs the data in Table 3 of the paper.

Stages 1-3 only

```
zipptab = S123data %>%
  group_by(ZippGroup) %>%
  summarize( numcorrect = sum(Stage3score),
             numingroup = n()) %>%
  mutate( pc = numcorrect/numingroup,
          entry = paste0(sprintf("%2.1f", pc*100), " (", numcorrect, "/",numingroup,")"))
zipptab %>% knitr::kable()
```

ZippGroup	numcorrect	numingroup	pc	entry
1	10	34	0.2941176	29.4 (10/34)
2	47	64	0.7343750	73.4 (47/64)
3	2	5	0.4000000	40.0 (2/5)
4	78	87	0.8965517	89.7 (78/87)

Stages 1-4

```
S124data = subset(data1718,select=c('Q','Student','Stage1score','Stage2score','Stage4score','ZippGroup'
S124data = S124data[complete.cases(S124data),]

zipptab124 = S124data %>%
  group_by(ZippGroup) %>%
  summarize( numcorrect = sum(Stage4score),
             numingroup = n()) %>%
  mutate( pc = numcorrect/numingroup,
          entry = paste0(sprintf("%2.1f", pc*100), " (", numcorrect, "/",numingroup,")"))
zipptab124 %>% knitr::kable()
```

ZippGroup	numcorrect	numingroup	pc	entry
1	17	54	0.3148148	31.5 (17/54)
2	30	61	0.4918033	49.2 (30/61)
3	3	4	0.7500000	75.0 (3/4)
4	34	57	0.5964912	59.6 (34/57)

Group dynamics: Stage 1 vs Stage 2

Here we look at the relative performance in the groups across the first two stages.

```
groupCorrectness = data1718 %>%
  mutate(
    Stage2group = paste0(Year,"_",Stage2group)
  ) %>%
```

```
  group_by(Stage2group,Q) %>%
  summarise(
    GpSize = n(),
    S1sum = sum(Stage1score),
    S1avg = S1sum/GpSize,
    S2 = max(Stage2score)
  )
groupCorrectness %>% ungroup() %>% gt()
```

Stage2group	Q	GpSize	S1sum	S1avg	S2
2017_1	Q1a	3	2	0.6666667	1
2017_1	Q1b	3	1	0.3333333	1
2017_1	Q2a	3	1	0.3333333	1
2017_1	Q2b	3	1	0.3333333	1
2017_1	Q2c	3	3	1.0000000	1
2017_1	Q11	3	0	0.0000000	0
2017_2	Q1a	3	2	0.6666667	1
2017_2	Q1b	3	1	0.3333333	0
2017_2	Q2a	3	1	0.3333333	1
2017_2	Q2b	3	3	1.0000000	1
2017_2	Q2c	3	1	0.3333333	1
2017_2	Q11	3	1	0.3333333	1
2017_3	Q1a	3	0	0.0000000	0
2017_3	Q1b	3	0	0.0000000	1
2017_3	Q2a	3	2	0.6666667	1
2017_3	Q2b	3	2	0.6666667	1
2017_3	Q2c	3	1	0.3333333	1
2017_3	Q11	3	0	0.0000000	0
2017_4	Q1a	3	2	0.6666667	1
2017_4	Q1b	3	0	0.0000000	0
2017_4	Q2a	3	1	0.3333333	1
2017_4	Q2b	3	2	0.6666667	1
2017_4	Q2c	3	2	0.6666667	1
2017_4	Q11	3	1	0.3333333	1
2017_5	Q1a	3	2	0.6666667	1
2017_5	Q1b	3	1	0.3333333	1
2017_5	Q2a	3	3	1.0000000	1
2017_5	Q2b	3	2	0.6666667	1
2017_5	Q2c	3	3	1.0000000	1
2017_5	Q11	3	0	0.0000000	0
2017_6	Q1a	3	2	0.6666667	1
2017_6	Q1b	3	1	0.3333333	1
2017_6	Q2a	3	1	0.3333333	1
2017_6	Q2b	3	1	0.3333333	1
2017_6	Q2c	3	2	0.6666667	1
2017_6	Q11	3	2	0.6666667	1
2017_7	Q1a	3	3	1.0000000	1
2017_7	Q1b	3	3	1.0000000	1
2017_7	Q2a	3	2	0.6666667	1
2017_7	Q2b	3	2	0.6666667	1
2017_7	Q2c	3	3	1.0000000	1
2017_7	Q11	3	1	0.3333333	1
2017_8	Q1a	3	2	0.6666667	1

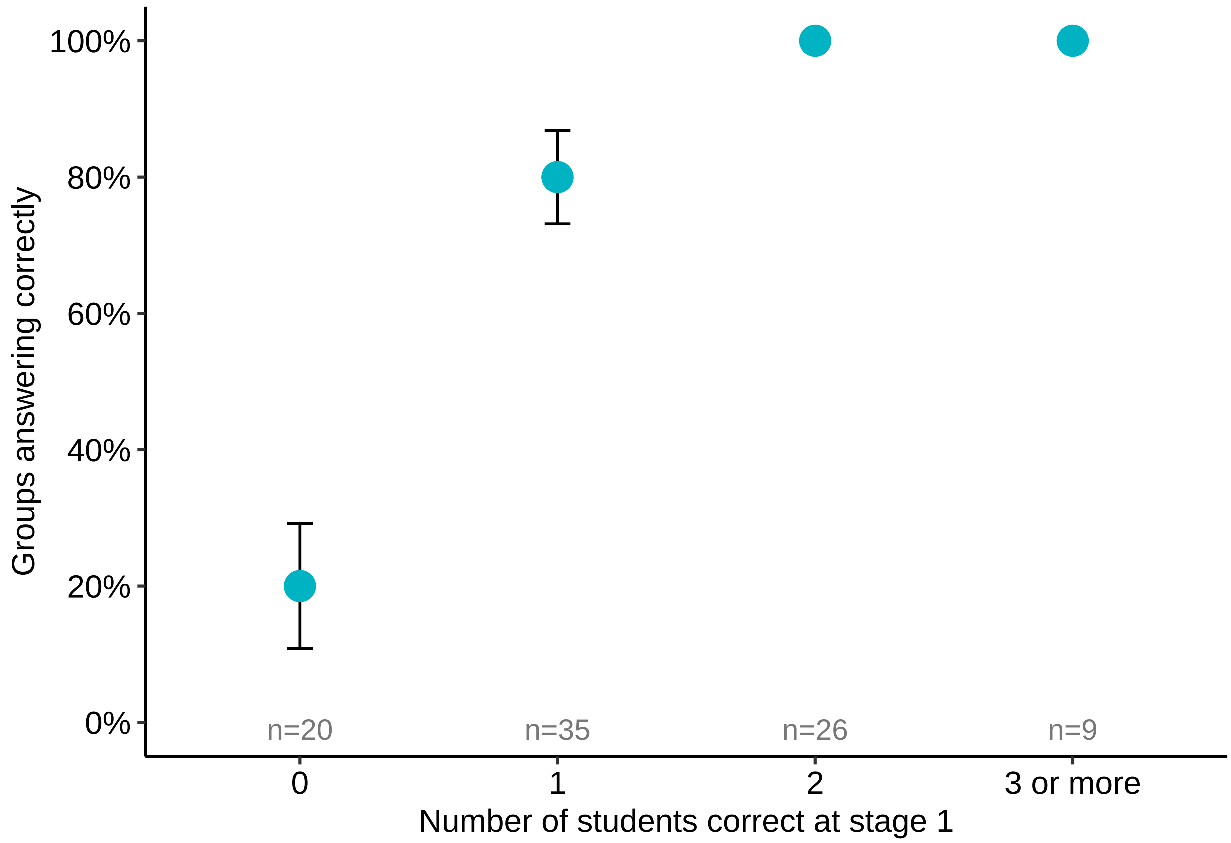
2017_8	Q1b	3	2	0.6666667	1
2017_8	Q2a	3	1	0.3333333	0
2017_8	Q2b	3	2	0.6666667	1
2017_8	Q2c	3	1	0.3333333	0
2017_8	Q11	3	0	0.0000000	1
2017_9	Q1a	3	1	0.3333333	1
2017_9	Q1b	3	2	0.6666667	1
2017_9	Q2a	3	2	0.6666667	1
2017_9	Q2b	3	1	0.3333333	1
2017_9	Q2c	3	2	0.6666667	1
2017_9	Q11	3	2	0.6666667	1
2018_1	Q1a	4	0	0.0000000	1
2018_1	Q1b	4	0	0.0000000	0
2018_1	Q2a	4	1	0.2500000	0
2018_1	Q2b	4	2	0.5000000	1
2018_1	Q2c	4	3	0.7500000	1
2018_1	Q11	4	0	0.0000000	0
2018_2	Q1a	4	2	0.5000000	1
2018_2	Q1b	4	1	0.2500000	1
2018_2	Q2a	4	0	0.0000000	0
2018_2	Q2b	4	2	0.5000000	1
2018_2	Q2c	4	3	0.7500000	1
2018_2	Q11	4	0	0.0000000	0
2018_3	Q1a	2	2	1.0000000	1
2018_3	Q1b	2	1	0.5000000	1
2018_3	Q2a	2	0	0.0000000	1
2018_3	Q2b	2	2	1.0000000	1
2018_3	Q2c	2	1	0.5000000	1
2018_3	Q11	2	1	0.5000000	1
2018_4	Q1a	3	1	0.3333333	1
2018_4	Q1b	3	1	0.3333333	1
2018_4	Q2a	3	0	0.0000000	0
2018_4	Q2b	3	1	0.3333333	1
2018_4	Q2c	3	0	0.0000000	0
2018_4	Q11	3	1	0.3333333	0
2018_5	Q1a	3	1	0.3333333	0
2018_5	Q1b	3	0	0.0000000	0
2018_5	Q2a	3	0	0.0000000	0
2018_5	Q2b	3	1	0.3333333	1
2018_5	Q2c	3	0	0.0000000	0
2018_5	Q11	3	0	0.0000000	0
2018_6	Q1a	4	1	0.2500000	1
2018_6	Q1b	4	0	0.0000000	0
2018_6	Q2a	4	1	0.2500000	1
2018_6	Q2b	4	1	0.2500000	1
2018_6	Q2c	4	1	0.2500000	0
2018_6	Q11	4	1	0.2500000	1

```
groupPerfS12 = groupCorrectness %>%
  mutate(
    tot_group = cut(S1sum,breaks=c(-Inf,0.5,1.5,2.5,Inf),labels=c("0","1","2","3 or more"))
  ) %>%
  group_by(tot_group) %>%
```

```r
  summarize(
    S2avg = mean(S2),
    S2se = sd(S2)/sqrt(n()),
    S2n = n()
  )
groupPerfS12 %>% knitr::kable()
```

tot_group	S2avg	S2se	S2n
0	0.2	0.0917663	20
1	0.8	0.0685994	35
2	1.0	0.0000000	26
3 or more	1.0	0.0000000	9

```r
ggplot(groupPerfS12,aes(x=tot_group,y=S2avg,label=S2n))+
  geom_errorbar(aes(ymax = groupPerfS12$S2avg + groupPerfS12$S2se,
                    ymin = groupPerfS12$S2avg - groupPerfS12$S2se),
                position = position_dodge(0.9),
                width = 0.1)+
  geom_point(size=5,colour=heathers[1])+
  scale_y_continuous(labels = scales::percent,breaks=seq(0,1,by=.2))+
  scale_color_manual(values=heathers) +
  coord_cartesian(ylim=c(0,1),clip="off")+
  geom_text(position = position_dodge(width = 0.9),
            aes(y=-0.01, label=paste0("n=",groupPerfS12$S2n)),
            angle=0,
            color="#777777")+
  labs(x = "Number of students correct at stage 1",
       y = "Groups answering correctly",
       colour = "Stage 2 attempt") +
  theme(strip.background = element_rect(fill=NA,colour = NA),
        strip.text = element_text(size=12, face="bold"))
```




```r
ggsave("Figs/Study1_S12_collab.pdf",width=15,height=7,units="cm",dpi=300)
```

Group dynamics

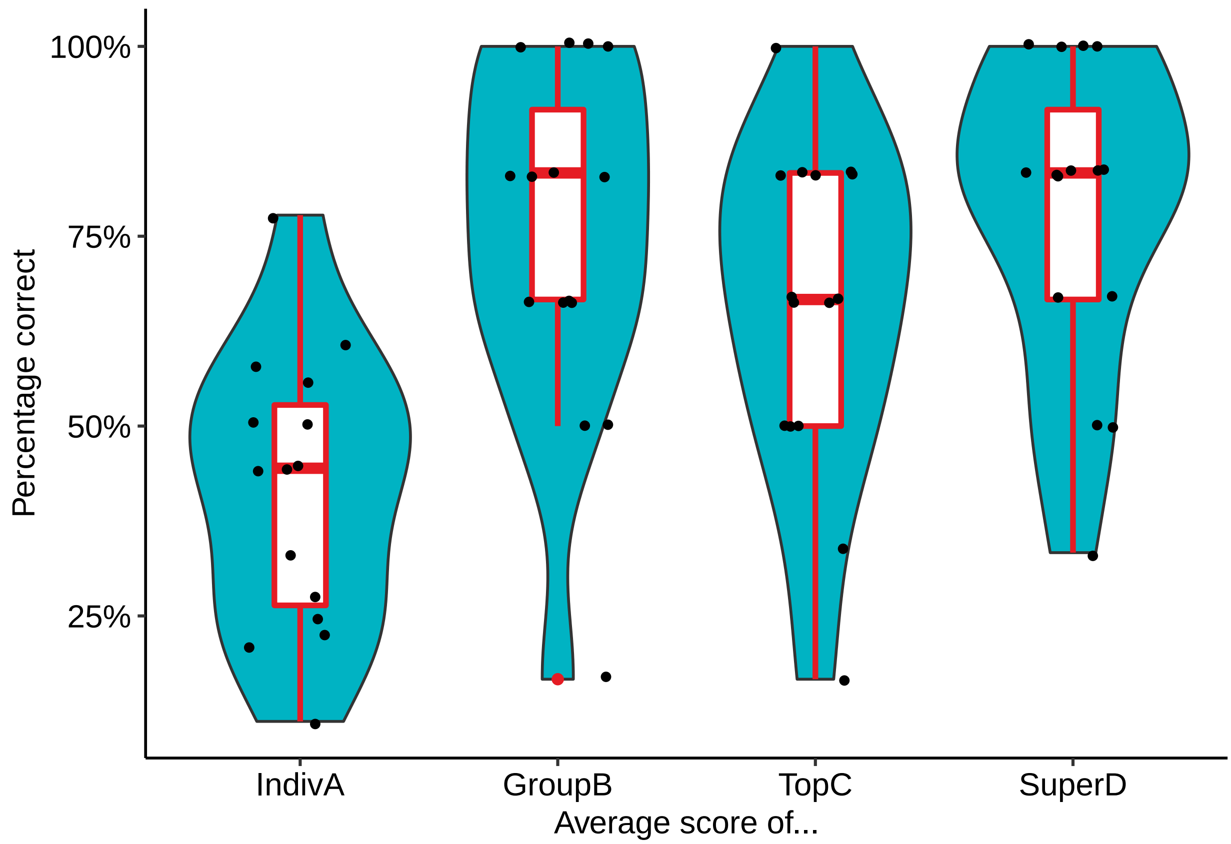
This replicates the analysis of Levy et al. (2018), producing Fig 2 of the paper. There is extra detail here, with the various measures like 'collaborative efficiency' shown for each group and also plotted.

```r
S12data_scored = data1718 %>%
  dplyr::select(Year,Q,Stage1score,Stage2score,Student,Stage2group) %>%
  mutate(
    Group = paste0(Year,"_",Stage2group)
  ) %>%
  dplyr::select(-Stage2group)

S1superandtop = S12data_scored %>%
  group_by(Group,Q) %>%
  mutate(
    superstudent = max(Stage1score)
  ) %>%
  group_by(Group,Student) %>%
  mutate(
    topstudent = sum(Stage1score)/n() # the Student's mean score on the n() Questions
  ) %>%
  group_by(Group) %>%
  summarise(
    superstudent = sum(superstudent)/n(),
```

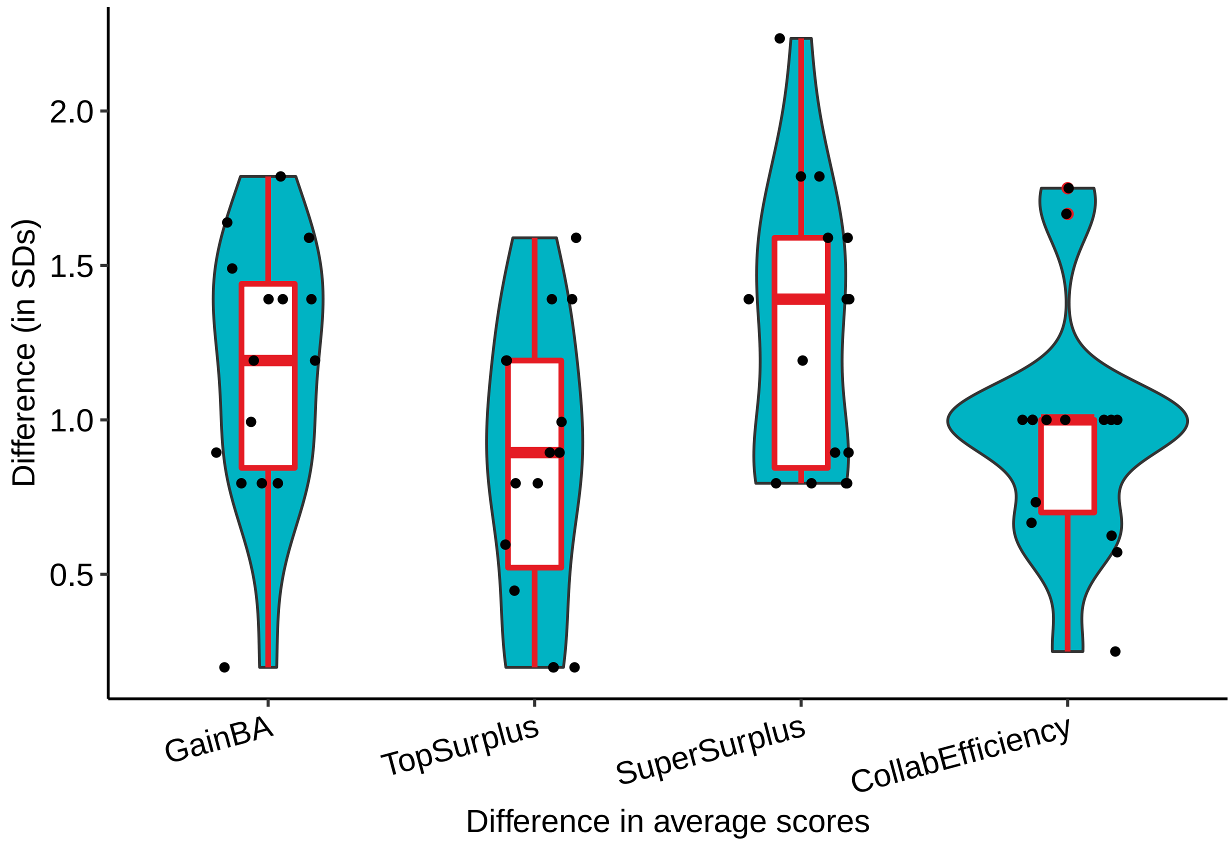
```r
    topstudent = max(topstudent)
  )

LevyA = S12data_scored %>%
  group_by(Student) %>%
  summarise(
    Stage1pc = sum(Stage1score)/n()
  ) %>%
  summarise(
    S1mean = mean(Stage1pc),
    S1sd = sd(Stage1pc),
    S1n = n()
  )

LevyAsd =  LevyA$S1sd[[1]]

LevyByGroup = groupCorrectness %>%
  mutate(
    Group = Stage2group
  ) %>%
  left_join(S1superandtop) %>%
  group_by(Group) %>%
  summarise(
    n = max(GpSize),
    IndivA = mean(S1avg),
    GroupB = mean(S2),
    TopC = max(topstudent),
    SuperD = max(superstudent),
    GainBA = (GroupB-IndivA)/LevyAsd,
    TopSurplus = (TopC-IndivA)/LevyAsd,
    SuperSurplus = (SuperD-IndivA)/LevyAsd,
    CollabEfficiency = GainBA / SuperSurplus
  ) %>%
  ungroup()
LevyByGroup %>% knitr::kable()
```

Group	n	IndivA	GroupB	TopC	SuperD	GainBA	TopSurplus	SuperSurplus	CollabEfficiency
2017_1	3	0.4444444	0.8333333	0.8333333	0.8333333	1.3907167	1.3907167	1.3907167	1.0000000
2017_2	3	0.5000000	0.8333333	0.8333333	1.0000000	1.1920428	1.1920428	1.7880643	0.6666667
2017_3	3	0.2777778	0.6666667	0.5000000	0.5000000	1.3907167	0.7946952	0.7946952	1.7500000
2017_4	3	0.4444444	0.8333333	0.8333333	0.8333333	1.3907167	1.3907167	1.3907167	1.0000000
2017_5	3	0.6111111	0.8333333	0.6666667	0.8333333	0.7946952	0.1986738	0.7946952	1.0000000
2017_6	3	0.5000000	1.0000000	0.6666667	1.0000000	1.7880643	0.5960214	1.7880643	1.0000000
2017_7	3	0.7777778	1.0000000	1.0000000	1.0000000	0.7946952	0.7946952	0.7946952	1.0000000
2017_8	3	0.4444444	0.6666667	0.5000000	0.8333333	0.7946952	0.1986738	1.3907167	0.5714286
2017_9	3	0.5555556	1.0000000	0.8333333	1.0000000	1.5893905	0.9933690	1.5893905	1.0000000
2018_1	4	0.2500000	0.5000000	0.5000000	0.5000000	0.8940321	0.8940321	0.8940321	1.0000000
2018_2	4	0.3333333	0.6666667	0.6666667	0.6666667	1.1920428	1.1920428	1.1920428	1.0000000
2018_3	2	0.5833333	1.0000000	0.8333333	0.8333333	1.4900536	0.8940321	0.8940321	1.6666667
2018_4	3	0.2222222	0.5000000	0.6666667	0.6666667	0.9933690	1.5893905	1.5893905	0.6250000
2018_5	3	0.1111111	0.1666667	0.1666667	0.3333333	0.1986738	0.1986738	0.7946952	0.2500000
2018_6	4	0.2083333	0.6666667	0.3333333	0.8333333	1.6390589	0.4470161	2.2350803	0.7333333

```
ggplot(stack(LevyByGroup %>% select(IndivA,GroupB,TopC,SuperD)), aes(x = ind, y = values)) +
  geom_violin(fill=heathers[1]) +
  geom_boxplot(width=0.2,color=heathers[2],lwd=1) +
  geom_jitter(shape=16, position=position_jitter(0.2)) +
  labs(x = "Average score of...",
       y = "Percentage correct") +
  scale_y_continuous(labels = scales::percent)
```




```
ggsave("Figs/Study1_LevyABCD.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study1_LevyABCD_small.pdf",width=10,height=7,units="cm",dpi=300)

ggplot(stack(LevyByGroup %>% select(GainBA,TopSurplus,SuperSurplus,CollabEfficiency)), aes(x = ind, y =
  geom_violin(fill=heathers[1]) +
  geom_boxplot(width=0.2,color=heathers[2],lwd=1) +
  geom_jitter(shape=16, position=position_jitter(0.2)) +
  labs(x = "Difference in average scores",
       y = "Difference (in SDs)")+
  theme(axis.text.x = element_text(angle = 15, hjust = 1))
```




```r
ggsave("Figs/Study1_LevyDiffs.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study1_LevyDiffs_small.pdf",width=10,height=7,units="cm",dpi=300)
```

```r
LevyByGroup %>%
  summarise(
    CollabEfficiency_m = mean(CollabEfficiency),
    CollabEfficiency_sd = sd(CollabEfficiency)
  ) %>% knitr::kable()
```

CollabEfficiency_m	CollabEfficiency_sd
0.950873	0.3817033

Bayesian analysis

Here we look at (and compare) the proportions in the 4 Zipp groups.

Using model code for the Bayesian First Aid alternative to the test of proportions.

```r
require(rjags)

source("DBDA2E-utilities.R")
```

```
##
## *******************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
```

```
## ***********************************************************************
```

```
source("DBDAderivatives.R")

myData = S123data %>%
  select(ZippGroup, Stage3score)

params = c(2,2)
# The model string written in the JAGS language
model_string <- paste0("model {
for(i in 1:length(x)) {
  x[i] ~ dbinom(theta[i], n[i])
  theta[i] ~ dbeta(",params[1],", ",params[2],")
  x_pred[i] ~ dbinom(theta[i], n[i])
}
}")

# Running the model
modelS3 <- jags.model(textConnection(model_string), data = list(x = zipptab$numcorrect, n = zipptab$num
                    n.chains = 3, n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 4
##    Unobserved stochastic nodes: 8
##    Total graph size: 17
##
## Initializing model
```

```
samplesS3 <- coda.samples(modelS3, c("theta", "x_pred"), n.iter=5000)

# Inspecting the posterior
#plot(samples)
#summary(samples)
```

You can extract the mcmc samples as a matrix and compare the thetas of the groups. For example, the following shows the median and 95% credible interval for the difference between Group 1 and Group 2.

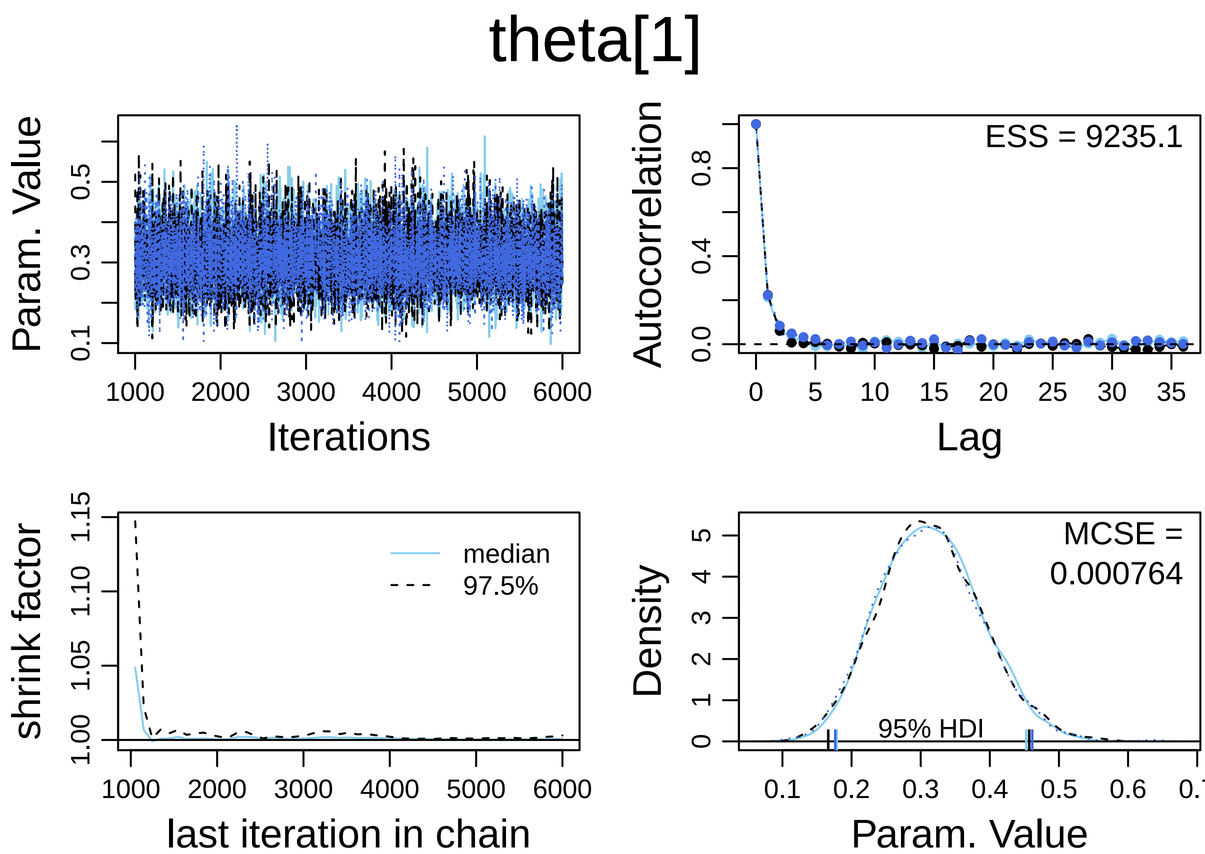
```
samp_mat <- as.matrix(samplesS3)
print(quantile(samp_mat[, "theta[2]"] - samp_mat[, "theta[1]"], c(0.025, 0.5, 0.975)))
```

```
##      2.5%       50%     97.5%
## 0.2234247 0.4088187 0.5783900
```

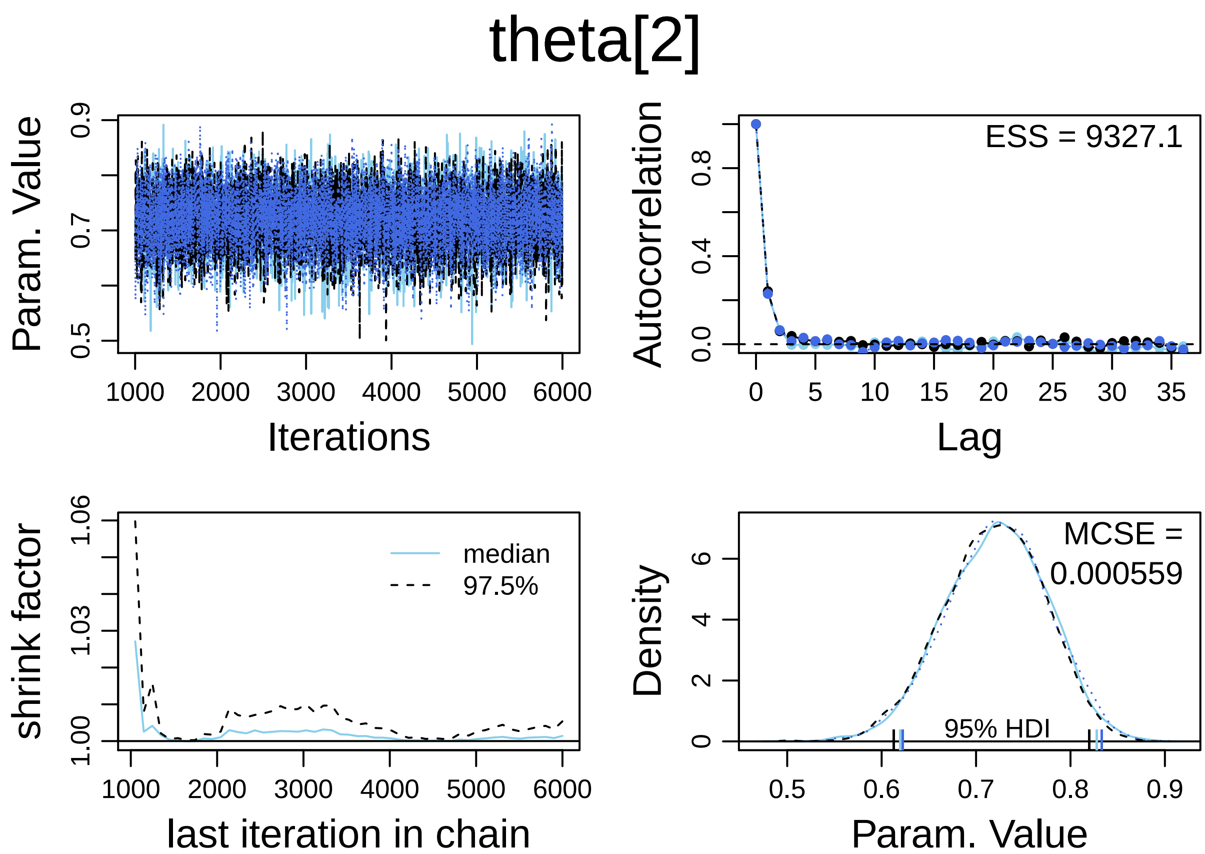
```
print(quantile(samp_mat[, "theta[4]"] - samp_mat[, "theta[3]"], c(0.025, 0.5, 0.975)))
```

```
##      2.5%       50%     97.5%
## 0.1143593 0.4390870 0.7283634
```

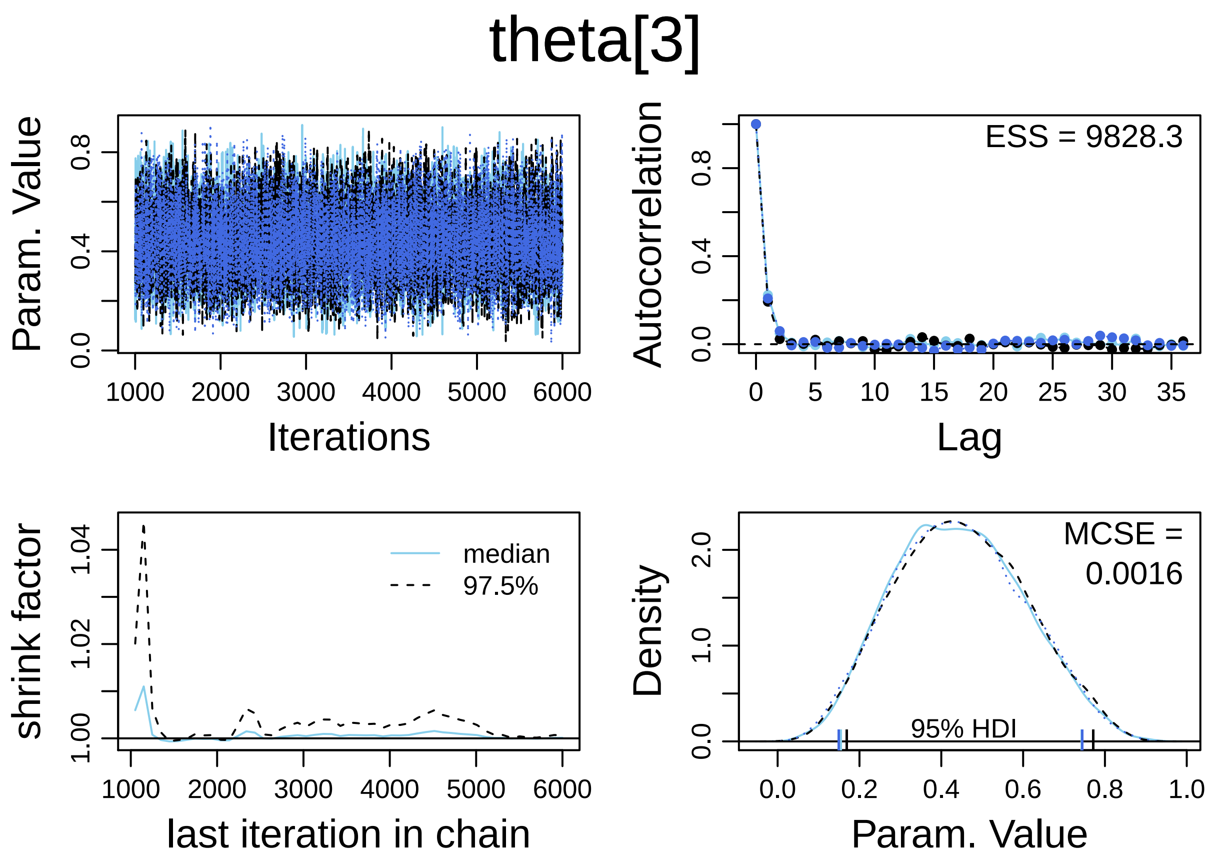
```
diagMCMC(samplesS3, parName = "theta[1]", saveName="Figs/Study1_S3props", saveType = "pdf")
```



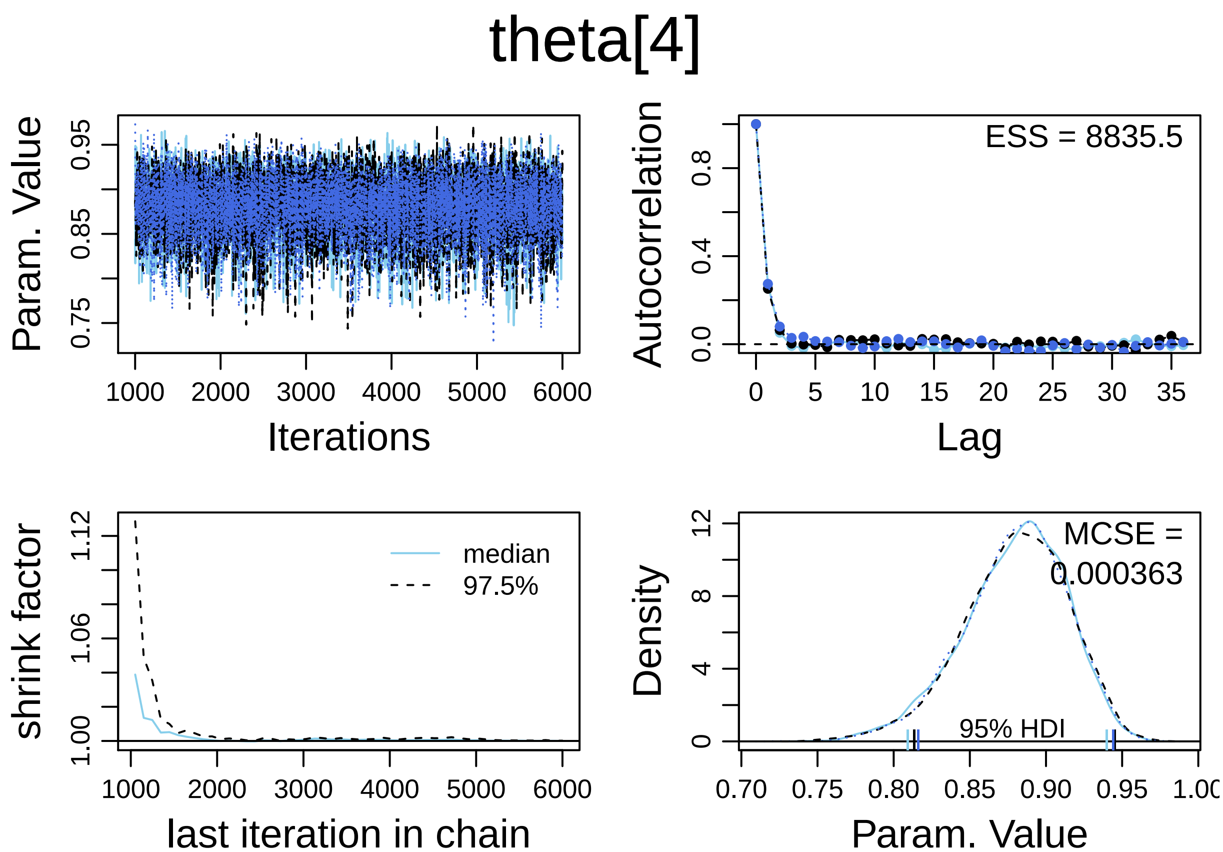

```r
diagMCMC(samplesS3, parName = "theta[2]", saveName="Figs/Study1_S3props", saveType = "pdf")
```



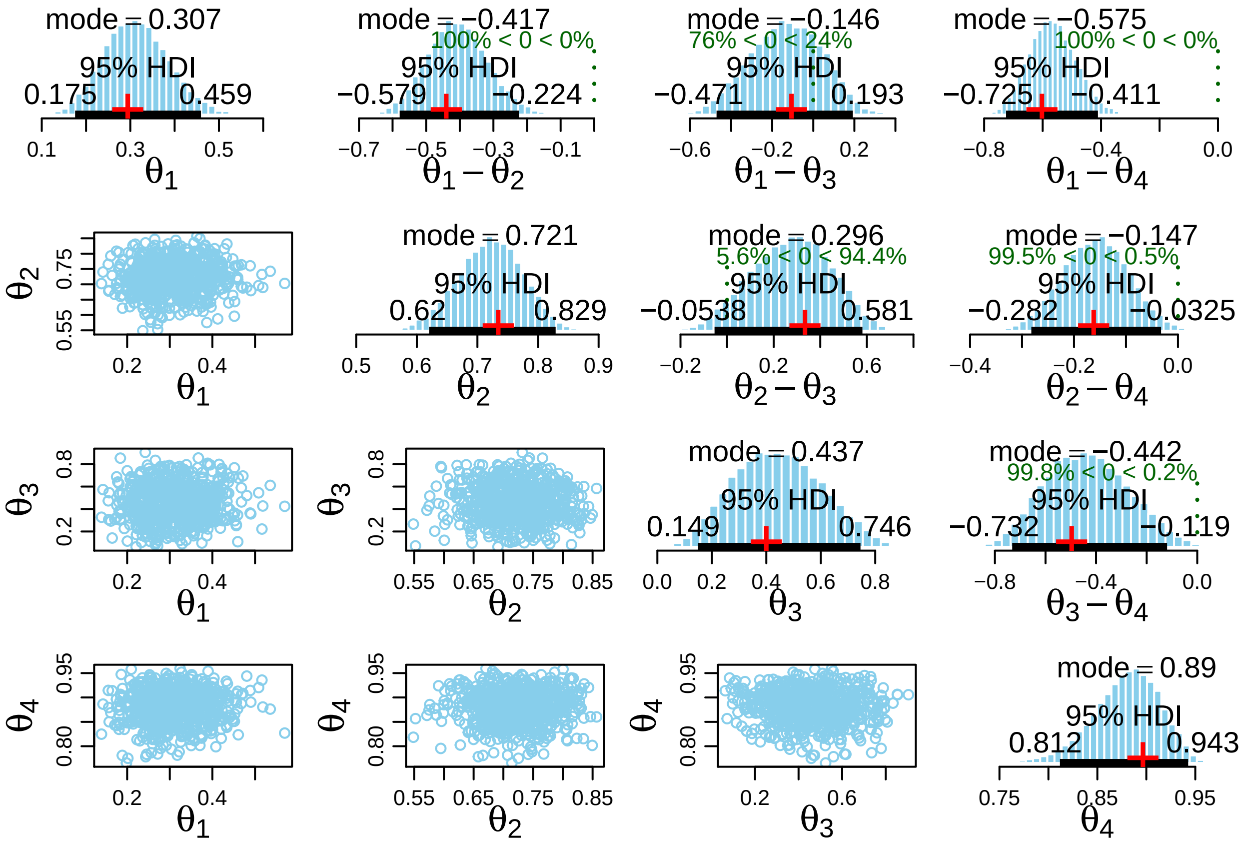

```r
diagMCMC(samplesS3, parName = "theta[3]", saveName="Figs/Study1_S3props", saveType = "pdf")
```



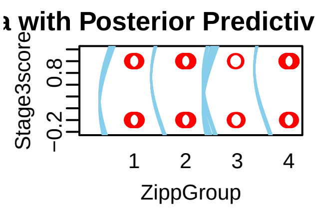

```r
diagMCMC(samplesS3, parName = "theta[4]", saveName="Figs/Study1_S3props", saveType = "pdf")
```




```
plotMCMC( samplesS3, data=myData, yName="Stage3score", sName="ZippGroup", compVal=NULL, compValDiff=0.0
          saveName="Figs/Study1_S3props", saveType = "pdf")
```



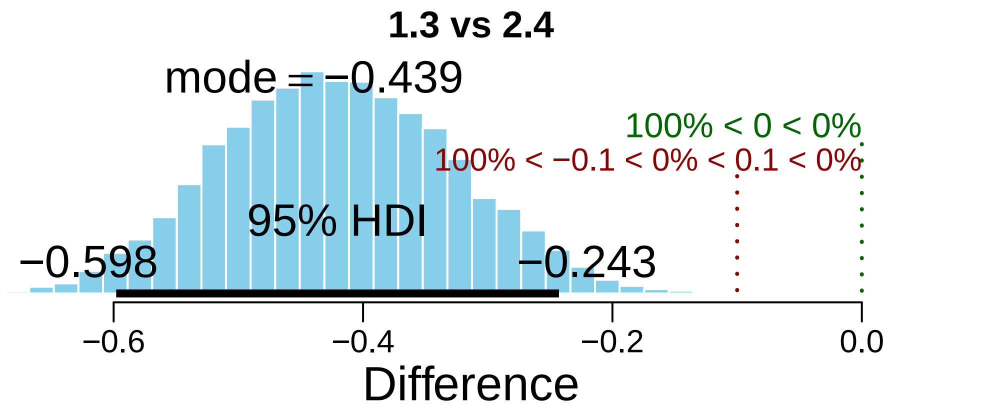

```
contrasts = list(
  list( c(1,3), c(2,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(3,4), compVal=0.0, ROPE=c(-0.1,0.1))
)
plotMCMCwithContrasts( samplesS3, datFrm=data.frame(myData), yName="Stage3score", xName="ZippGroup", co
          saveName="Figs/Study1_S3props", saveType = "pdf")
```

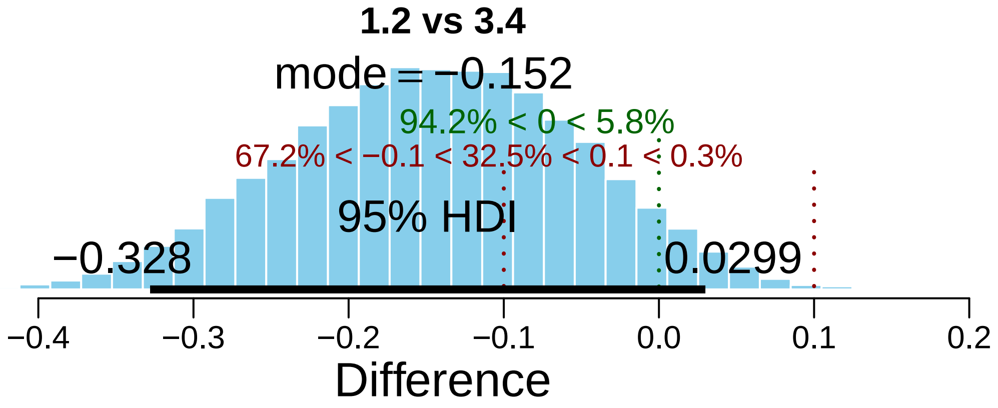



```
## [1] 1
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 2
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
```

```
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 3
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 4
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
```

```
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
```





For Stage 4

```
myData124 = S124data %>%
  select(ZippGroup, Stage4score)

params = c(2,2)
# The model string written in the JAGS language
model_string <- paste0("model {
 for(i in 1:length(x)) {
   x[i] ~ dbinom(theta[i], n[i])
   theta[i] ~ dbeta(",params[1],", ",params[2],")
   x_pred[i] ~ dbinom(theta[i], n[i])
 }
 }")

# Running the model
modelS4 <- jags.model(textConnection(model_string), data = list(x = zipptab124$numcorrect, n = zipptab1
                      n.chains = 3, n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 4
##    Unobserved stochastic nodes: 8
##    Total graph size: 17
```

```
##
## Initializing model
```

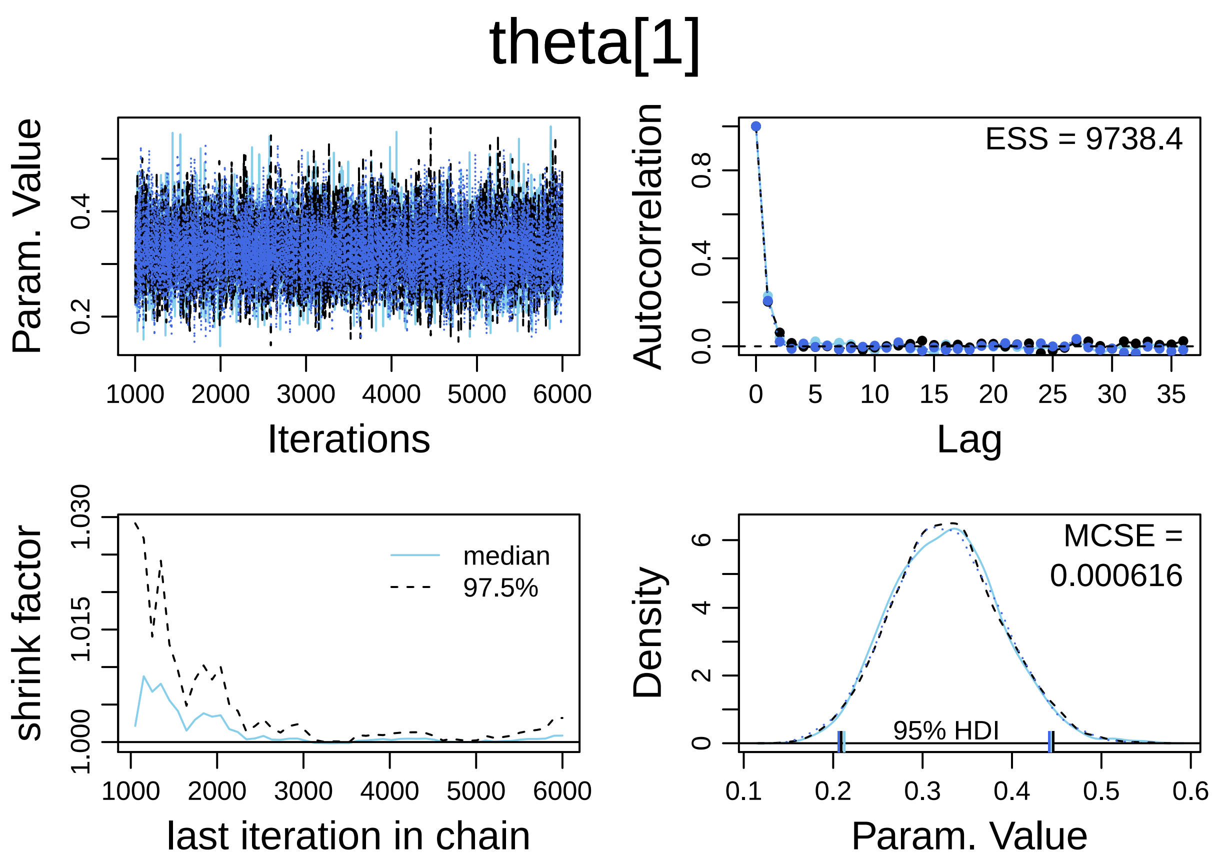
```r
samplesS4 <- coda.samples(modelS4, c("theta", "x_pred"), n.iter=5000)

samp_mat <- as.matrix(samplesS4)
print(quantile(samp_mat[, "theta[2]"] - samp_mat[, "theta[1]"], c(0.025, 0.5, 0.975)))
```

```
##         2.5%          50%        97.5%
## -0.006556672  0.166641171  0.332032899
```

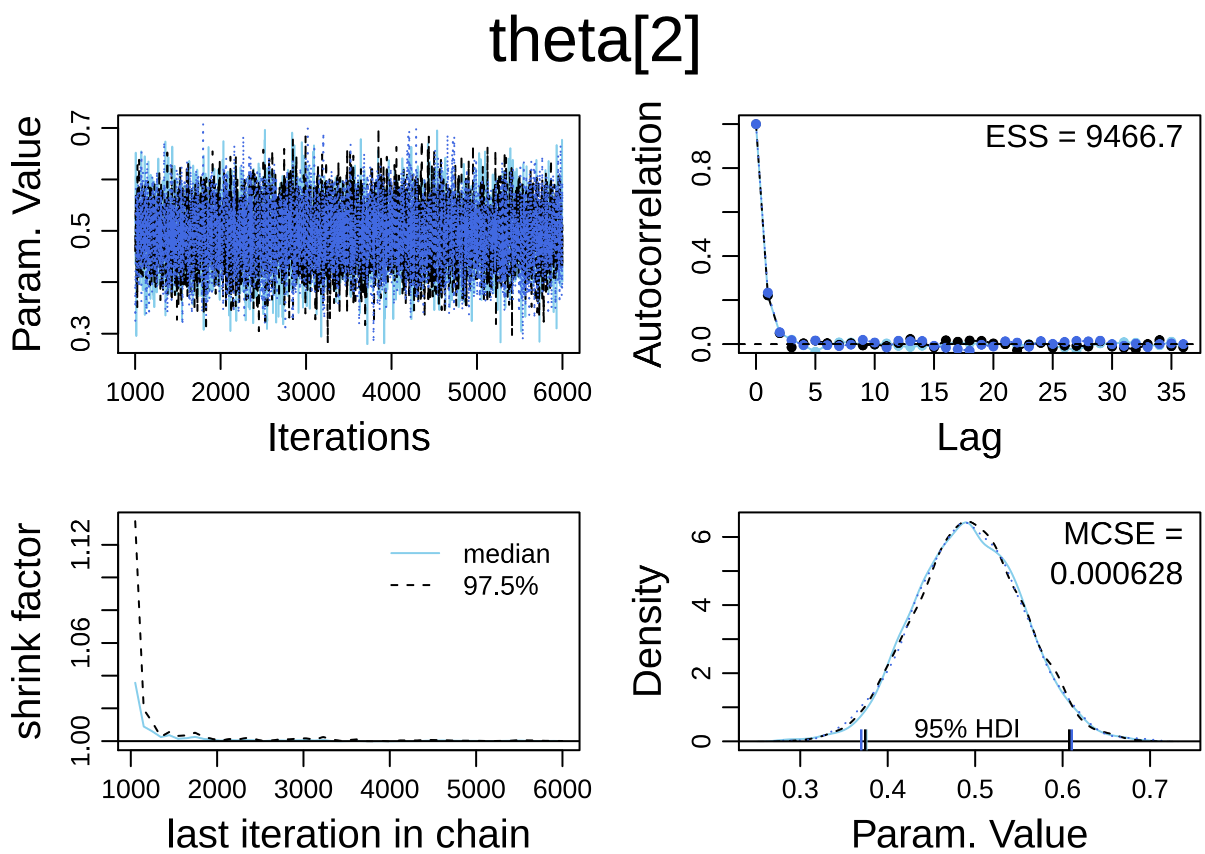
```r
print(quantile(samp_mat[, "theta[4]"] - samp_mat[, "theta[3]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.34018328 -0.04419645  0.31970151
```

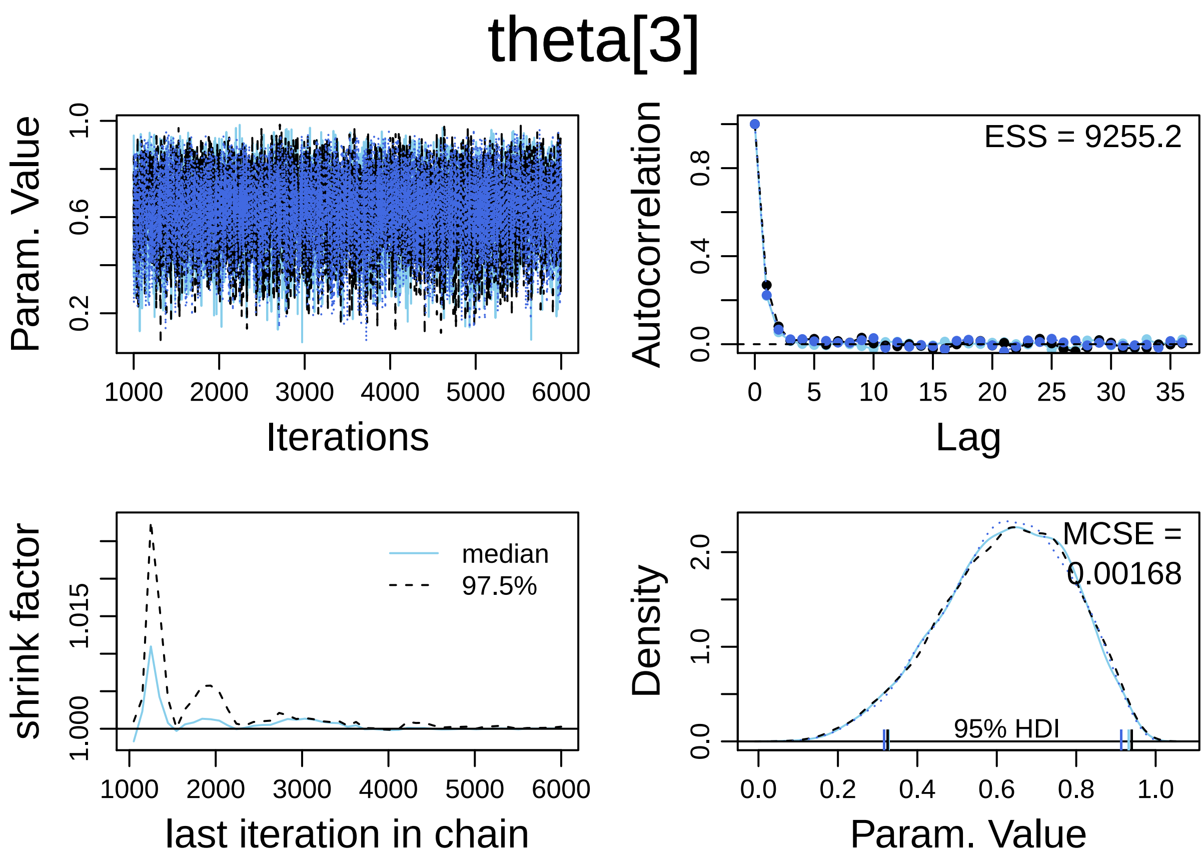
```r
diagMCMC(samplesS4, parName = "theta[1]", saveName="Figs/Study1_S4props", saveType = "pdf")
```



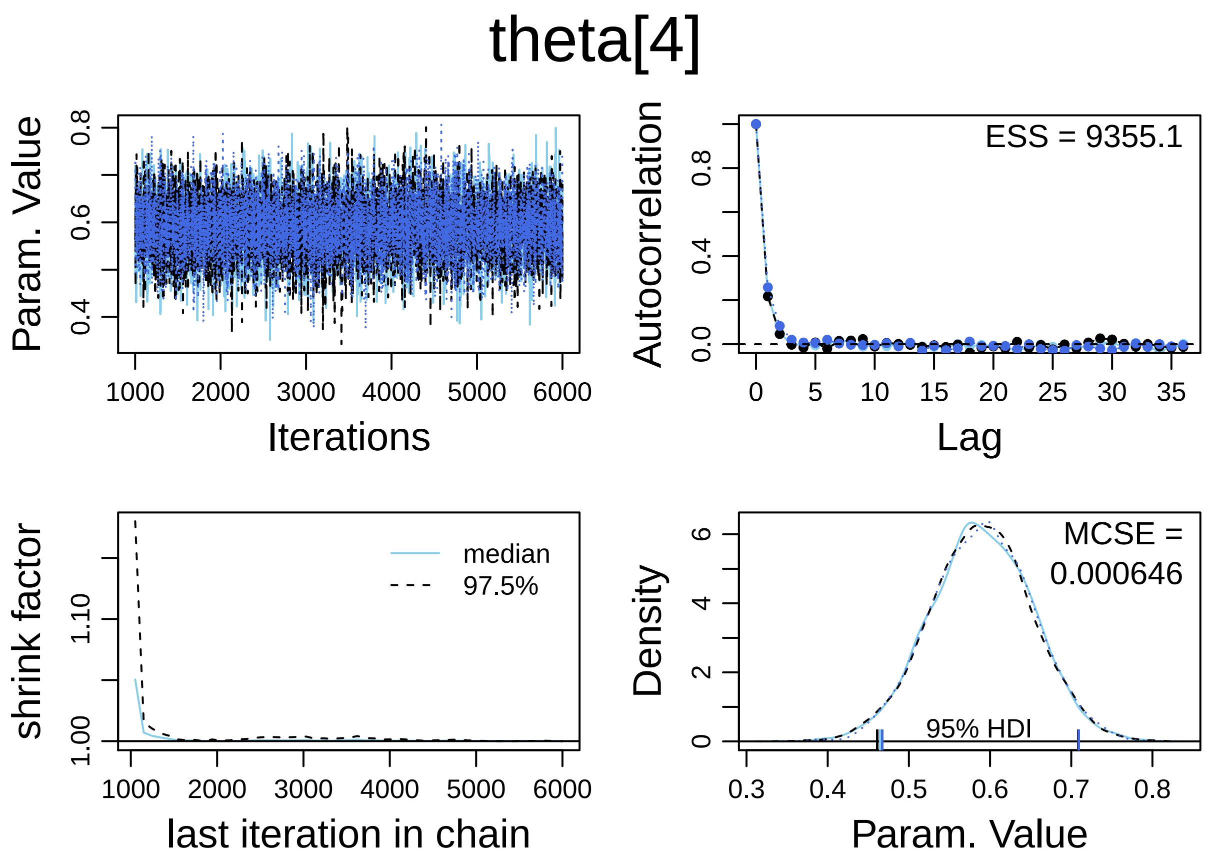

```r
diagMCMC(samplesS4, parName = "theta[2]", saveName="Figs/Study1_S4props", saveType = "pdf")
```



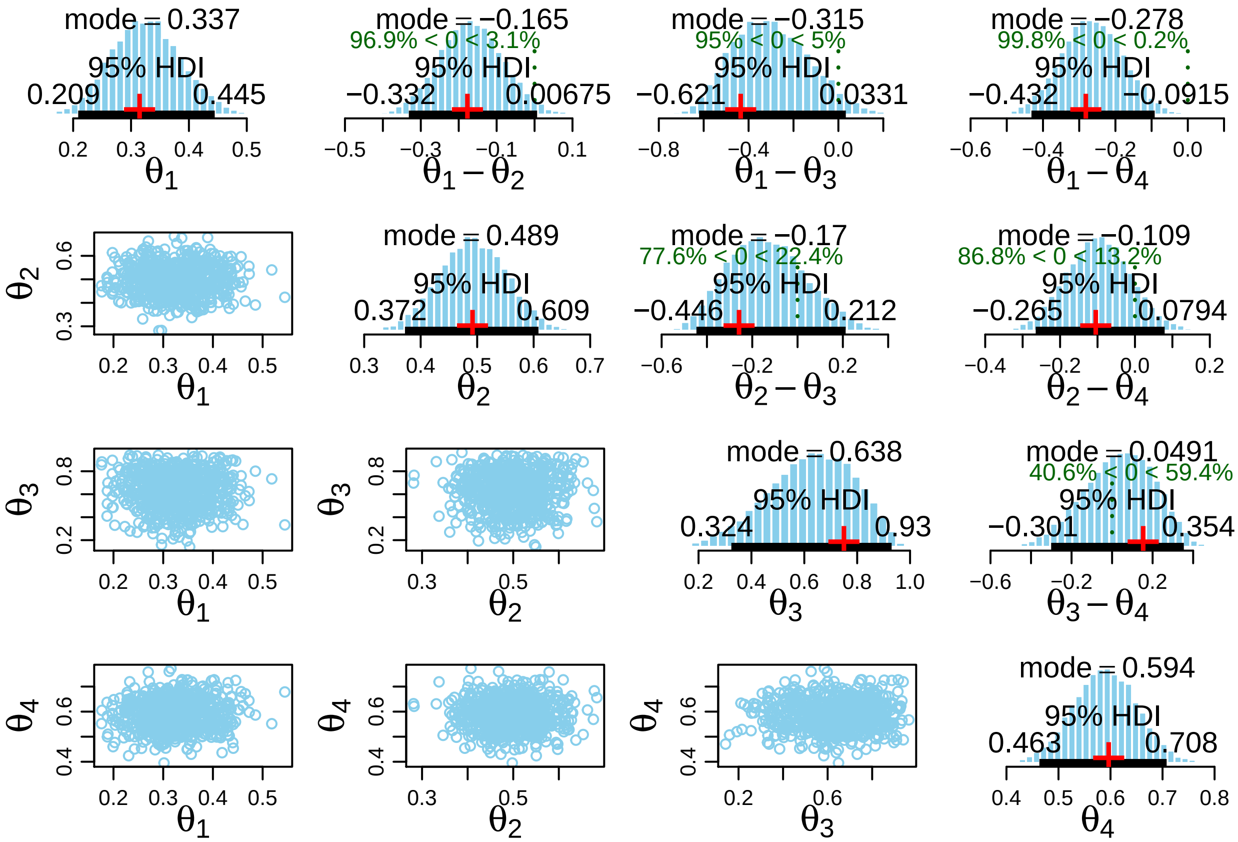

```
diagMCMC(samplesS4, parName = "theta[3]", saveName="Figs/Study1_S4props", saveType = "pdf")
```



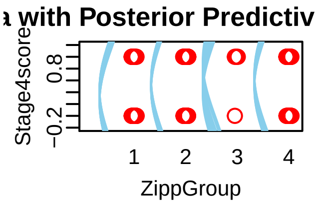

```
diagMCMC(samplesS4, parName = "theta[4]", saveName="Figs/Study1_S4props", saveType = "pdf")
```




```
plotMCMC( samplesS4, data=myData124, yName="Stage4score", sName="ZippGroup", compVal=NULL, compValDiff=
          saveName="Figs/Study1_S4props", saveType = "pdf")
```



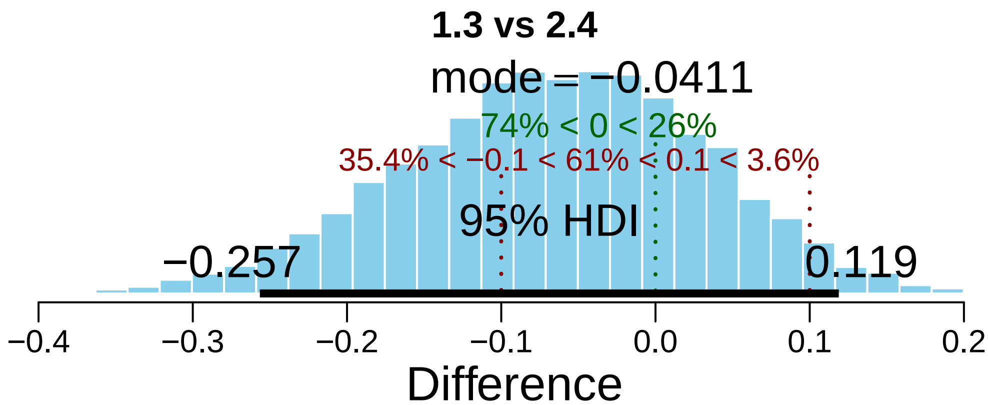
```
contrasts = list(
  list( c(1,3), c(2,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(3,4), compVal=0.0, ROPE=c(-0.1,0.1))
)
plotMCMCwithContrasts( samplesS4, datFrm=data.frame(myData124), yName="Stage4score", xName="ZippGroup",
                       saveName="Figs/Study1_S4props", saveType = "pdf")
```

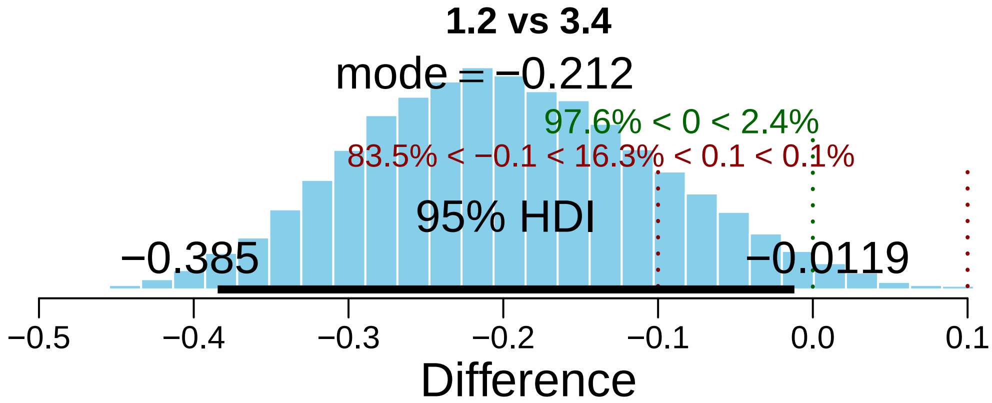



```
## [1] 1
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 2
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
```

```
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 3
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 4
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
```

```
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
```




1.2 vs 3.4
mode = −0.212
97.6% < 0 < 2.4%
83.5% < −0.1 < 16.3% < 0.1 < 0.1%
95% HDI
−0.385
−0.0119
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
Difference