Two-stage exams: Study 3

George Kinnear

20/06/2020

Contents

Data

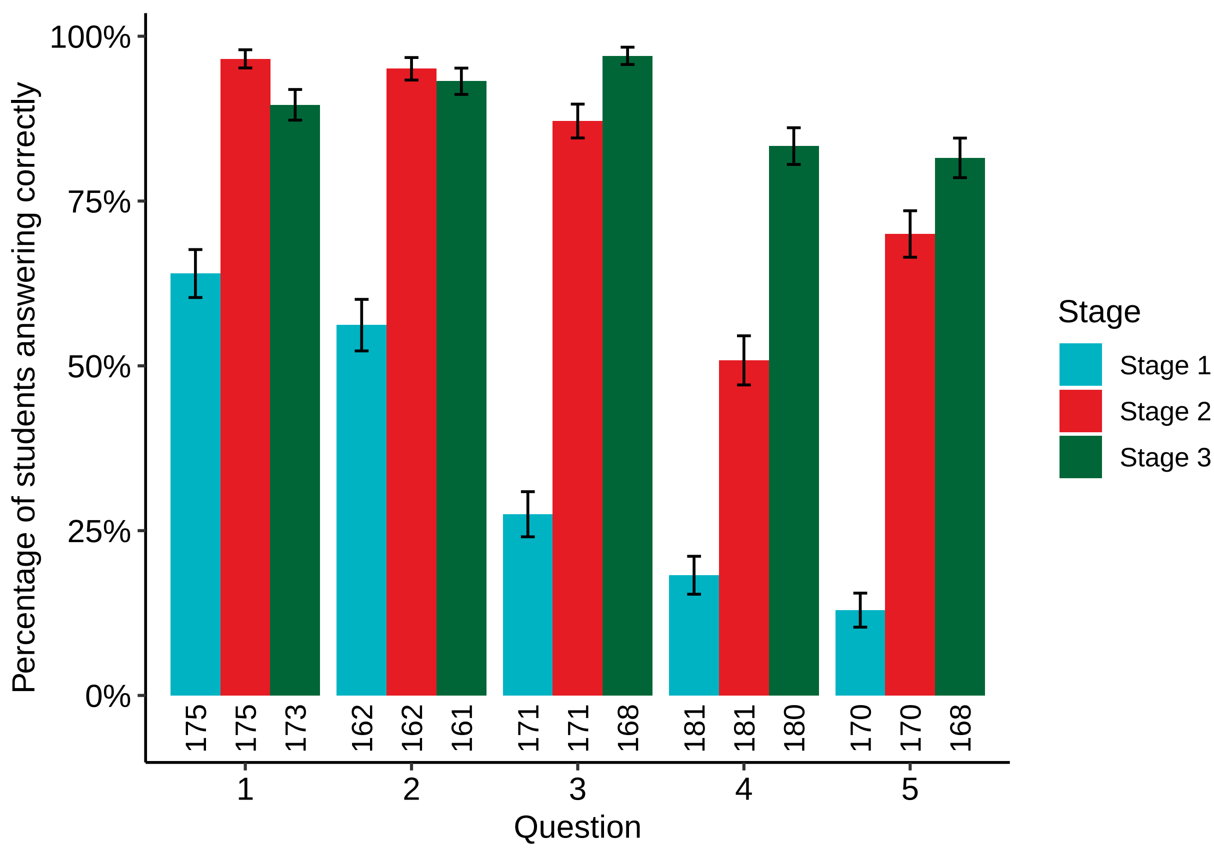
Import the dataset.

```
data = read.csv('Study3_data.csv', header=T) %>%
  dplyr::select(-c(X))


S123data = subset(data,
                  select=c('Q','Student','Stage1score','Stage2score','Stage3score')) %>%
  filter(!is.na(Stage2score))
```

Mean at each stage (as bars)

```
ld <- gather(data = S123data,
             key = stage,
             value = score,
             Stage1score, Stage2score, Stage3score)
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")
```

```r
plotData = plotData %>%
  mutate(
    Stage = gsub(".*(\\d).*","\\1",Stage)  # alternatively: Stage = parse_number(Stage)
  )
plotCounts = plotData %>% group_by(Q,Stage) %>% select(Stage,n) %>%
  mutate(scale_lab = paste0("Stage ",Stage, " (n=",n,")"))

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_text(position = position_dodge(width = 0.9),aes(y=-0.05),angle=90) +
  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=paste0("Stage ",c(1:4))) + #plotCounts$scale_lab) +
  #scale_fill_grey() +
  scale_y_continuous(labels = scales::percent)
p
```



```
ggsave("Figs/Study3_S123_means.pdf",width=20,height=10,units="cm",dpi=300)
```

A look at the data (this only shows the first few rows, but for a sanity check the full table could be consulted):

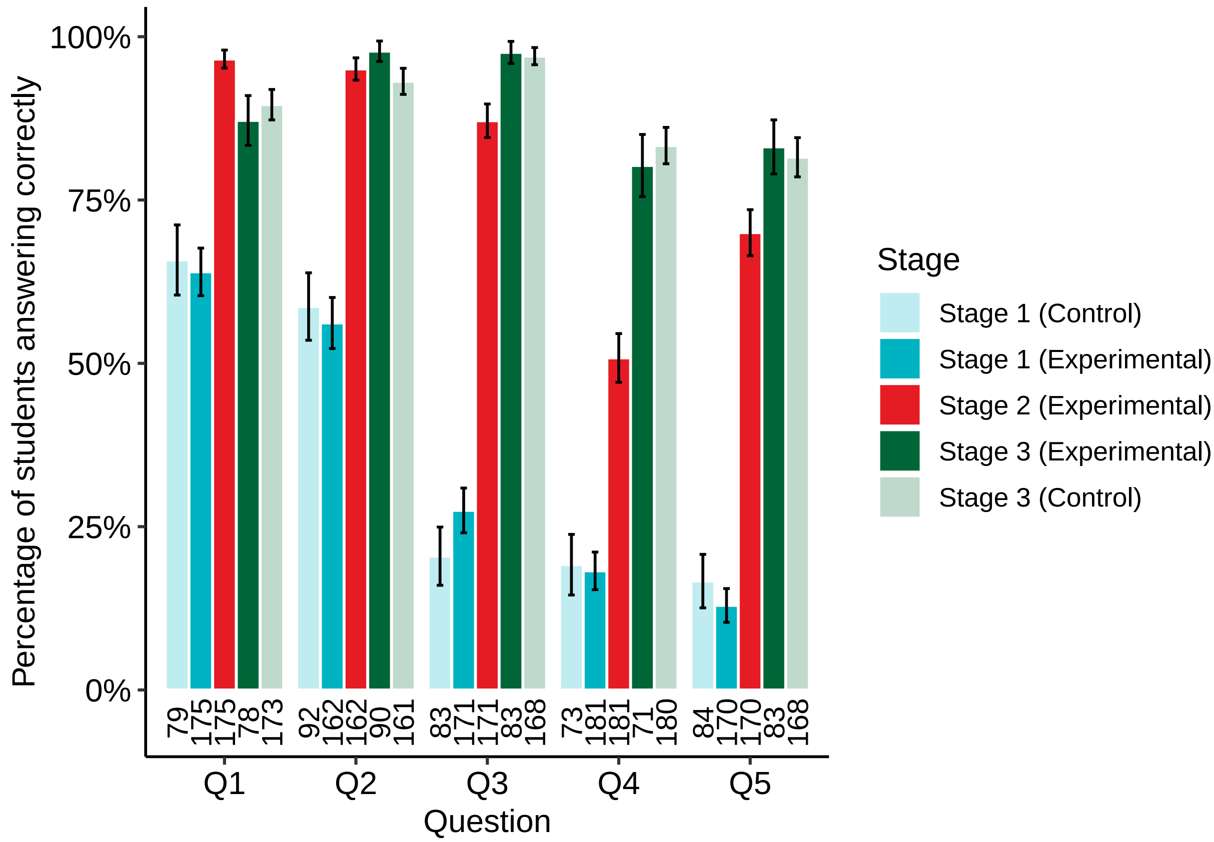
```
S123data %>%
  group_by(Student) %>%
  mutate(
    Stage1sum = sum(Stage1score),
    Stage2sum = sum(Stage2score),
    Stage3sum = sum(Stage3score),
    qs = str_length(paste0(Stage1score, collapse=""))
  ) %>%
  ungroup() %>%
  mutate(
    S1max = max(Stage1sum)
  ) %>%
  arrange(-qs) %>%
  head() %>%
  knitr::kable()
```

Q	Student	Stage1score	Stage2score	Stage3score	Stage1sum	Stage2sum	Stage3sum	qs	S1max
1	058ce7a0	1	1	1	3	4	4	4	3
1	3966c674	1	1	1	1	3	2	4	3
1	e1f2a406	0	1	1	2	2	4	4	3
1	3ad8b6e0	1	1	1	2	3	3	4	3
1	2d9b726e	0	1	1	1	4	2	4	3
1	3e9bfda8	1	1	1	2	3	3	4	3

Bar plot

```
barPlotData = data %>%
  dplyr::select(c('Q','Student','ZippGroup','Stage1score','Stage2score','Stage3score')) %>%
  gather('Stage1score','Stage2score','Stage3score',key="Stage", value="Score") %>%
  drop_na() %>%
  mutate(
    Stage=parse_number(Stage),
    expt = case_when(
      str_sub(ZippGroup,1,1)=="E" ~ "E",
      TRUE ~ "C"
    ),
    bar = case_when(
      Stage==1 & expt=="E" ~ "1E",
      Stage==1 & expt=="C" ~ "1C",
      Stage==2 ~ "2E",
      Stage==3 & expt=="E" ~ "3E",
      Stage==3 & expt=="C" ~ "3C"
    )
  ) %>%
  group_by(Q,bar) %>%
  summarise(
    mean=mean(Score,na.rm=TRUE),
    sd=sd(Score,na.rm=TRUE),
    n=n(),
```

```r
    se=sd/sqrt(n)
  ) %>%
  ungroup() %>%
  mutate(
    Q=paste0("Q",Q),
    Stage=parse_number(bar),
    Expt=str_sub(bar,2,2),
    ConditionOrder = case_when(
      bar=="1C" ~ 1,
      bar=="1E" ~ 2,
      bar=="2E" ~ 3,
      bar=="3E" ~ 4,
      bar=="3C" ~ 5
    ),
    bar2=fct_reorder(bar,ConditionOrder)
  ) %>% arrange(Q,ConditionOrder)


barLabels = c("Stage 1 (Control)", "Stage 1 (Experimental)",
              "Stage 2 (Experimental)",
              "Stage 3 (Experimental)", "Stage 3 (Control)")
barColoursAlpha = c(alpha(heathers[1],.25),heathers[1],heathers[2],heathers[3],alpha(heathers[3],.25))
barColoursAlpha = c("#bfecf0",heathers[1],heathers[2],heathers[3],"#bfd9cd")

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

ggplot(data = barPlotData, aes(x = factor(Q), y = mean,
                                    fill = factor(bar), label=n)) +
  geom_bar(stat = "identity",
           position = position_dodge(0.9),color="white")+
  geom_text(position = position_dodge(width = 0.9),aes(y=-0.05),angle=90) +
  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=barColoursAlpha, labels=barLabels) +
  scale_y_continuous(labels = scales::percent)
```

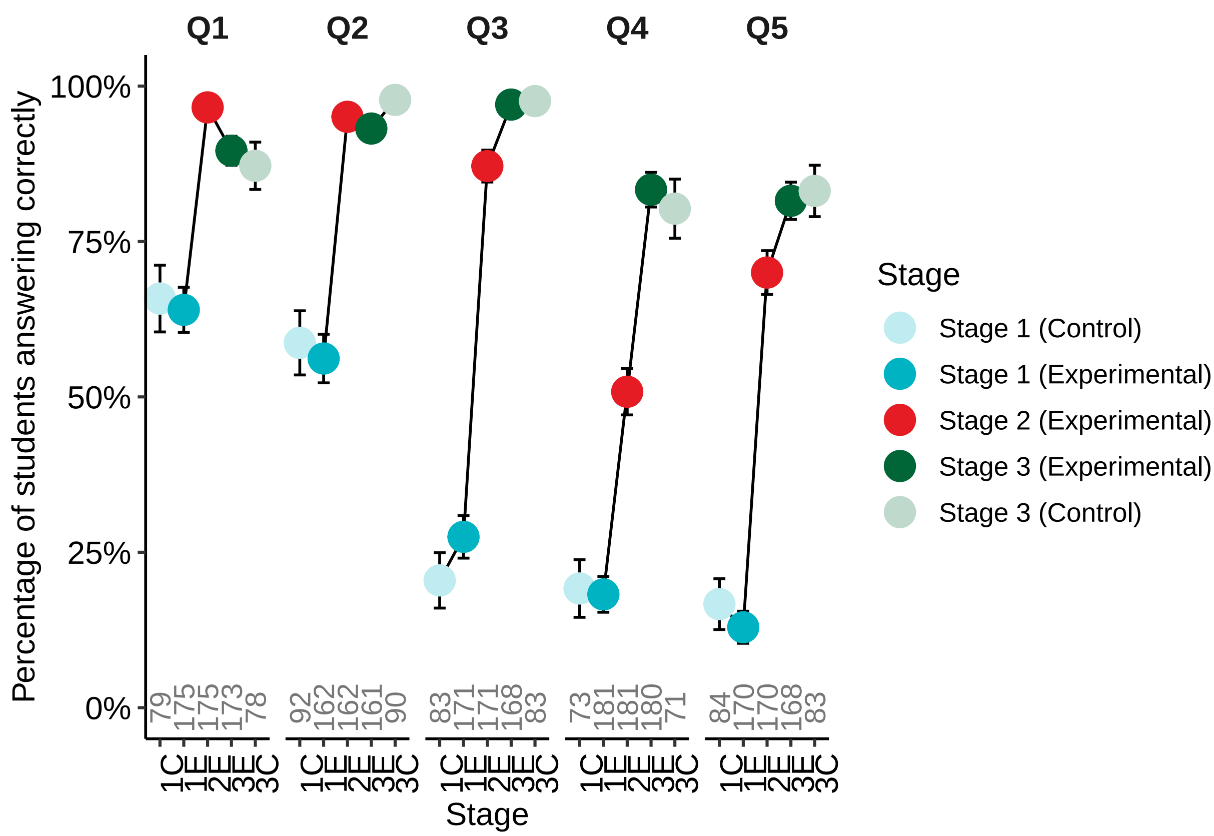



```
ggsave("Figs/Study3_S123_means.pdf",width=20,height=10,units="cm",dpi=300)
```

Mean at each stage (as points)

```
ggplot(data=barPlotData,aes(x=bar2,y=mean,group=Q,label=n))+
  geom_line()+
  geom_errorbar(aes(ymax = barPlotData$mean + barPlotData$se,
                    ymin = barPlotData$mean - barPlotData$se),
                position = position_dodge(0.9),
                width = 0.5)  +
  geom_point(position = position_dodge(0.9),size=5,
             aes(color=bar2))+
  facet_grid(cols=vars(Q))+
  labs(x = "Stage",
       y = "Percentage of students answering correctly",
       color = "Stage") +
  scale_y_continuous(labels = scales::percent)+
  scale_color_manual(values=barColoursAlpha, labels=barLabels)+
  coord_cartesian(ylim=c(0,1),clip="off")+
  geom_text(position = position_dodge(width = 0.9),
            aes(y=0, label=paste0("",barPlotData$n)),
            angle=90,
            color="#777777") +
  theme(strip.background = element_rect(fill=NA,colour = NA),
        strip.text = element_text(size=12, face="bold"),
```

```
      axis.text.x = element_text(angle=90))
```




```
ggsave("Figs/Study3_S123_means_pts.pdf",width=20,height=10,units="cm",dpi=300)
```

Mean at each stage (table, with standard errors)

```
tab = barPlotData %>%
  mutate(
    entry = paste0(sprintf("%2.0f", mean*100), " (", sprintf("%2.1f", se*100), ")")
  ) %>%
  group_by(Q,bar) %>%
  select(Q,bar,entry) %>%
  spread(Q,entry)

tab$bar = barLabels
tab %>% knitr::kable()
```

bar	Q1	Q2	Q3	Q4	Q5
Stage 1 (Control)	66 (5.4)	59 (5.2)	20 (4.5)	19 (4.6)	17 (4.1)
Stage 1 (Experimental)	64 (3.6)	56 (3.9)	27 (3.4)	18 (2.9)	13 (2.6)
Stage 2 (Experimental)	97 (1.4)	95 (1.7)	87 (2.6)	51 (3.7)	70 (3.5)
Stage 3 (Experimental)	87 (3.8)	98 (1.6)	98 (1.7)	80 (4.8)	83 (4.1)
Stage 3 (Control)	90 (2.3)	93 (2.0)	97 (1.3)	83 (2.8)	82 (3.0)

Forming the Zipp tables

This constructs the data in Table 5 of the paper. Note that this only has data for the experimental condition – the data for the full Table 5 (i.e. including the control condition) appears in the final section of this script, where the experimental analysis takes place.

Table 5, first attempt

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

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
E1	141	168	0.8392857	83.9 (141/168)
E2	330	379	0.8707124	87.1 (330/379)
E3	7	7	1.0000000	100.0 (7/7)
E4	277	296	0.9358108	93.6 (277/296)

Group dynamics: Stage 1 vs Stage 2

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

```
groupCorrectness = data %>%
  group_by(Stage2group,Q) %>%
  summarise(
    GpSize = n(),
    S1sum = sum(Stage1score),
    S1avg = S1sum/GpSize,
    S2 = max(Stage2score),
    S2pc = ceiling(max(Stage2scorePC)) ## round up so that it's 1 if they were correct on second attemp
  ) %>%
  filter(
    !is.na(S2)
  )
groupCorrectness %>% ungroup() %>% knitr::kable()
```

Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
1	1	4	3	0.7500000	1	1
1	3	4	1	0.2500000	1	1
1	4	4	1	0.2500000	1	1
1	5	4	0	0.0000000	1	1
2	1	3	2	0.6666667	1	1
2	3	3	0	0.0000000	1	1
2	4	3	0	0.0000000	0	0
3	1	4	2	0.5000000	1	1

Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
3	2	4	2	0.5000000	1	1
3	4	4	1	0.2500000	0	0
4	2	4	1	0.2500000	1	1
4	3	4	1	0.2500000	1	1
4	5	4	0	0.0000000	1	1
5	1	4	3	0.7500000	1	1
5	2	4	2	0.5000000	1	1
5	4	4	0	0.0000000	0	1
5	5	4	3	0.7500000	1	1
6	1	4	4	1.0000000	1	1
6	3	4	1	0.2500000	1	1
6	5	4	0	0.0000000	0	1
7	1	4	3	0.7500000	1	1
7	3	4	1	0.2500000	1	1
7	4	4	0	0.0000000	0	0
8	1	4	2	0.5000000	1	1
8	3	4	1	0.2500000	1	1
8	4	4	0	0.0000000	0	1
8	5	4	0	0.0000000	1	1
9	1	4	4	1.0000000	1	1
9	2	4	2	0.5000000	1	1
9	5	4	0	0.0000000	1	1
10	1	4	3	0.7500000	1	1
10	3	4	1	0.2500000	1	1
10	4	4	0	0.0000000	0	0
10	5	4	0	0.0000000	1	1
12	1	4	3	0.7500000	1	1
12	2	4	2	0.5000000	1	1
12	5	4	0	0.0000000	0	0
13	2	4	2	0.5000000	0	1
13	3	4	1	0.2500000	1	1
13	4	4	0	0.0000000	0	0
14	1	4	4	1.0000000	1	1
14	3	4	1	0.2500000	1	1
14	4	4	1	0.2500000	1	1
15	1	3	0	0.0000000	0	1
15	2	3	1	0.3333333	1	1
15	4	3	1	0.3333333	1	1
16	1	4	1	0.2500000	1	1
16	2	4	4	1.0000000	1	1
16	4	4	0	0.0000000	0	0
17	1	4	3	0.7500000	1	1
17	2	4	1	0.2500000	1	1
17	4	4	0	0.0000000	0	0
17	5	4	0	0.0000000	1	1
18	1	2	2	1.0000000	1	1
18	2	2	2	1.0000000	1	1
18	5	2	0	0.0000000	0	0
19	2	3	1	0.3333333	1	1
19	3	3	2	0.6666667	1	1
19	5	3	2	0.6666667	1	1
21	2	3	2	0.6666667	1	1

Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
21	3	3	2	0.6666667	1	1
21	4	3	0	0.0000000	0	0
22	2	4	0	0.0000000	1	1
22	3	4	4	1.0000000	1	1
22	4	4	2	0.5000000	1	1
23	1	3	1	0.3333333	1	1
23	2	3	3	1.0000000	1	1
23	5	3	2	0.6666667	1	1
24	2	4	1	0.2500000	1	1
24	3	4	0	0.0000000	0	0
24	5	4	0	0.0000000	1	1
25	2	4	2	0.5000000	1	1
25	3	4	3	0.7500000	1	1
25	4	4	0	0.0000000	0	1
26	1	4	3	0.7500000	1	1
26	3	4	1	0.2500000	1	1
26	4	4	1	0.2500000	1	1
26	5	4	0	0.0000000	0	1
28	2	4	3	0.7500000	1	1
28	3	4	1	0.2500000	1	1
28	4	4	2	0.5000000	1	1
28	5	4	0	0.0000000	0	1
29	1	4	3	0.7500000	1	1
29	2	4	4	1.0000000	1	1
29	4	4	0	0.0000000	0	1
29	5	4	0	0.0000000	0	1
30	1	4	3	0.7500000	1	1
30	3	4	1	0.2500000	1	1
30	4	4	1	0.2500000	1	1
30	5	4	3	0.7500000	1	1
31	2	3	1	0.3333333	1	1
31	3	3	1	0.3333333	1	1
31	4	3	1	0.3333333	1	1
31	5	3	1	0.3333333	1	1
32	2	4	2	0.5000000	1	1
32	3	4	1	0.2500000	1	1
32	4	4	1	0.2500000	1	1
33	1	4	2	0.5000000	1	1
33	2	4	2	0.5000000	1	1
33	4	4	1	0.2500000	1	1
33	5	4	0	0.0000000	1	1
34	1	4	3	0.7500000	1	1
34	2	4	1	0.2500000	1	1
34	4	4	1	0.2500000	1	1
35	1	4	3	0.7500000	1	1
35	3	4	2	0.5000000	1	1
35	4	4	1	0.2500000	1	1
35	5	4	2	0.5000000	1	1
36	2	4	0	0.0000000	0	0
36	3	4	1	0.2500000	1	1
36	5	4	0	0.0000000	0	0
37	2	4	1	0.2500000	1	1

Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
37	3	4	1	0.2500000	1	1
37	5	4	0	0.0000000	1	1
38	2	2	1	0.5000000	1	1
38	3	2	0	0.0000000	0	1
38	5	2	0	0.0000000	0	0
39	1	4	3	0.7500000	1	1
39	3	4	2	0.5000000	1	1
39	4	4	1	0.2500000	1	1
40	2	4	2	0.5000000	1	1
40	3	4	1	0.2500000	1	1
40	4	4	0	0.0000000	0	0
40	5	4	1	0.2500000	1	1
41	2	4	4	1.0000000	1	1
41	3	4	2	0.5000000	1	1
41	4	4	1	0.2500000	1	1
41	5	4	0	0.0000000	1	1
42	1	4	2	0.5000000	1	1
42	3	4	0	0.0000000	1	1
42	4	4	0	0.0000000	0	1
43	1	4	1	0.2500000	1	1
43	3	4	0	0.0000000	1	1
43	5	4	0	0.0000000	0	0
44	1	3	2	0.6666667	1	1
44	2	3	2	0.6666667	1	1
44	5	3	0	0.0000000	1	1
45	1	4	2	0.5000000	1	1
45	3	4	1	0.2500000	1	1
45	4	4	2	0.5000000	1	1
46	2	4	2	0.5000000	1	1
46	3	4	0	0.0000000	0	0
46	4	4	0	0.0000000	0	1
46	5	4	0	0.0000000	1	1
47	1	3	2	0.6666667	1	1
47	3	3	1	0.3333333	1	1
47	4	3	0	0.0000000	0	1
47	5	3	1	0.3333333	1	1
48	1	4	2	0.5000000	1	1
48	3	4	0	0.0000000	1	1
48	5	4	0	0.0000000	1	1
49	1	4	3	0.7500000	1	1
49	3	4	1	0.2500000	0	0
49	5	4	0	0.0000000	1	1
50	1	4	2	0.5000000	1	1
50	2	4	2	0.5000000	1	1
50	5	4	2	0.5000000	1	1
52	1	4	3	0.7500000	1	1
52	2	4	4	1.0000000	1	1
52	4	4	0	0.0000000	0	0
53	1	4	2	0.5000000	1	1
53	2	4	2	0.5000000	1	1
53	4	4	0	0.0000000	0	0
54	1	3	2	0.6666667	1	1

Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
54	2	3	1	0.3333333	1	1
54	4	3	1	0.3333333	1	1
55	1	4	4	1.0000000	1	1
55	2	4	2	0.5000000	1	1
55	4	4	0	0.0000000	0	0
55	5	4	0	0.0000000	0	0
56	1	4	2	0.5000000	1	1
56	3	4	0	0.0000000	1	1
56	5	4	1	0.2500000	1	1
57	1	4	1	0.2500000	1	1
57	2	4	3	0.7500000	1	1
57	4	4	2	0.5000000	0	1
57	5	4	0	0.0000000	0	1
58	1	4	3	0.7500000	1	1
58	3	4	3	0.7500000	1	1
58	4	4	1	0.2500000	1	1
59	2	4	2	0.5000000	1	1
59	3	4	1	0.2500000	1	1
59	4	4	2	0.5000000	1	1
59	5	4	0	0.0000000	1	1
60	1	3	3	1.0000000	1	1
60	2	3	2	0.6666667	1	1
60	5	3	0	0.0000000	0	0
61	2	4	3	0.7500000	1	1
61	3	4	2	0.5000000	1	1
61	4	4	1	0.2500000	1	1
62	1	3	1	0.3333333	0	1
62	2	3	3	1.0000000	1	1
62	5	3	1	0.3333333	1	1
63	2	4	3	0.7500000	1	1
63	3	4	2	0.5000000	1	1
63	4	4	1	0.2500000	1	1
63	5	4	0	0.0000000	0	1
64	1	4	2	0.5000000	1	1
64	3	4	0	0.0000000	1	1
64	5	4	1	0.2500000	1	1
65	2	4	3	0.7500000	1	1
65	3	4	2	0.5000000	1	1
65	4	4	0	0.0000000	0	0
66	1	4	3	0.7500000	1	1
66	3	4	1	0.2500000	1	1
66	4	4	0	0.0000000	0	0
66	5	4	0	0.0000000	1	1
67	1	3	1	0.3333333	1	1
67	2	3	2	0.6666667	1	1
67	4	3	1	0.3333333	1	1
67	5	3	0	0.0000000	1	1
68	1	4	3	0.7500000	1	1
68	3	4	0	0.0000000	1	1
68	4	4	3	0.7500000	1	1
69	1	2	2	1.0000000	1	1
69	3	2	0	0.0000000	1	1

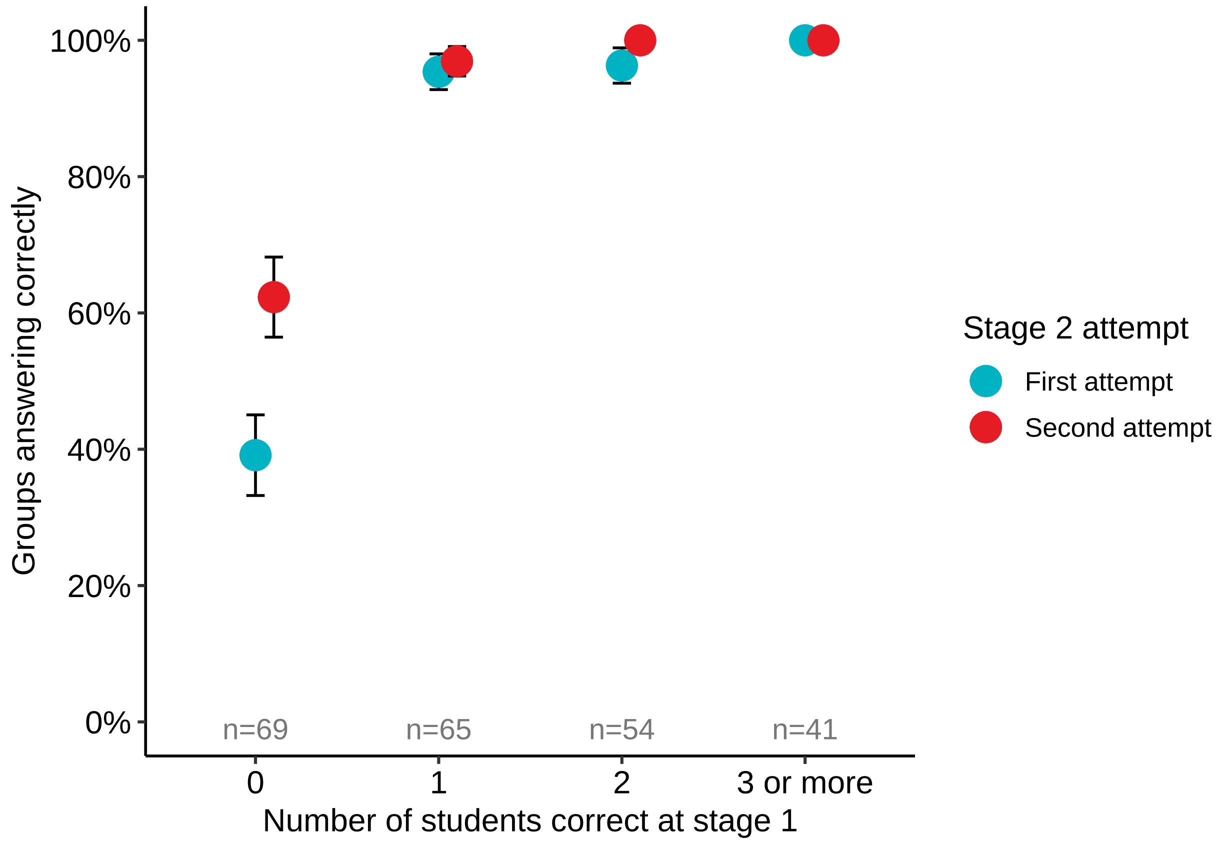
Stage2group	Q	GpSize	S1sum	S1avg	S2	S2pc
69	5	2	1	0.5000000	1	1
70	2	4	3	0.7500000	1	1
70	3	4	0	0.0000000	0	1
70	4	4	0	0.0000000	0	0
70	5	4	0	0.0000000	0	0
71	1	4	2	0.5000000	1	1
71	2	4	3	0.7500000	1	1
71	4	4	1	0.2500000	1	1
71	5	4	1	0.2500000	1	1
72	1	4	2	0.5000000	1	1
72	3	4	0	0.0000000	0	0
72	4	4	1	0.2500000	1	1
72	5	4	0	0.0000000	1	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) %>%
  summarize(
    S2avg = mean(S2),
    S2se = sd(S2)/sqrt(n()),
    S2n = n(),
    S2Pavg = mean(S2pc),
    S2Pse = sd(S2pc)/sqrt(n())
  )
groupPerfS12 %>% knitr::kable()
```

tot_group	S2avg	S2se	S2n	S2Pavg	S2Pse
0	0.3913043	0.0591838	69	0.6231884	0.0587648
1	0.9538462	0.0262273	65	0.9692308	0.0215865
2	0.9629630	0.0259409	54	1.0000000	0.0000000
3 or more	1.0000000	0.0000000	41	1.0000000	0.0000000

```
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(aes(colour="First attempt"),size=5)+
  geom_errorbar(aes(ymax = groupPerfS12$S2Pavg + groupPerfS12$S2Pse,
                    ymin = groupPerfS12$S2Pavg - groupPerfS12$S2Pse),
                position = position_nudge(x=0.1),
                width = 0.1)+
  geom_point(aes(y=S2Pavg,colour="Second attempt"),position=position_nudge(x=0.1),size=5)+
  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"))
```




```
ggsave("Figs/Study3_S12_collab.pdf",width=15,height=7,units="cm",dpi=300)
```

Group dynamics

This replicates the analysis of Levy et al. (2018), producing Fig 7 of the paper. There is extra detail here, with the various measures like 'collaborative efficiency' shown for each group and also plotted.

Find the top scoring student in each group, and the "super" score (max score across all students in the group, by question)

```
S12data_scored = data %>%
  dplyr::select(Q,Stage1score,Stage2score,Student,Stage2group) %>%
  mutate(
    Group = Stage2group
  ) %>%
  dplyr::select(-Stage2group)

S1superandtop = S12data_scored %>%
  group_by(Group,Q) %>%
```

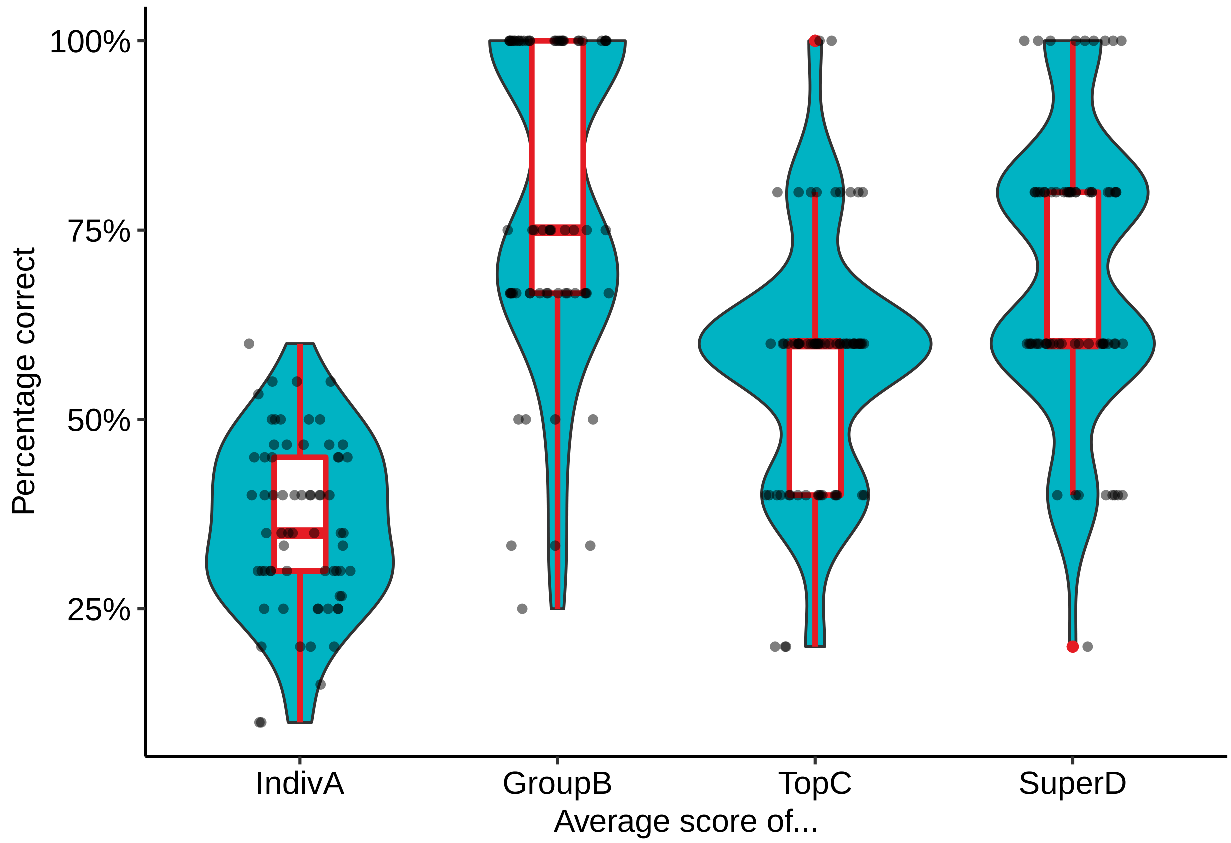
```r
  mutate(
    superstudent = max(Stage1score)
  ) %>%
  group_by(Group,Student) %>%
  mutate(
    topstudent = sum(Stage1score)/n() # the Student's mean score on the n() Questions
  ) %>%
  group_by(Group) %>%
  summarise(
    superstudent = sum(superstudent)/n(),
    topstudent = max(topstudent)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


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


LevyAsd =  LevyA$S1sd[[1]]

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

LevyByGroup = groupCorrectness %>%
  mutate(
    Group = Stage2group
  ) %>%
  left_join(S1superandtop) %>%
#  left_join(LevyA %>% select(S1sd)) %>%
  group_by(Group) %>%
  summarise(
    n = max(GpSize),
    IndivA = mean(S1avg),
    GroupB = mean(S2,na.rm=TRUE),
    TopC = max(topstudent),
    SuperD = max(superstudent),
    GainBA = (GroupB-IndivA)/LevyAsd,
    TopSurplus = (TopC-IndivA)/LevyAsd,
    SuperSurplus = (SuperD-IndivA)/LevyAsd,
```

```r
    CollabEfficiency = GainBA / na_if(SuperSurplus,0)
  ) %>%
  ungroup()

LevyByGroup %>% knitr::kable(digits = 2)
```

Group	n	IndivA	GroupB	TopC	SuperD	GainBA	TopSurplus	SuperSurplus	CollabEfficiency
1	4	0.35	1.00	0.8	0.8	3.02	2.09	2.09	1.44
2	3	0.27	0.67	0.4	0.4	1.86	0.62	0.62	3.00
3	4	0.30	0.67	0.6	0.8	1.70	1.39	2.32	0.73
4	4	0.30	1.00	0.6	0.8	3.25	1.39	2.32	1.40
5	4	0.45	0.75	0.8	0.8	1.39	1.63	1.63	0.86
6	4	0.35	0.67	0.6	0.6	1.47	1.16	1.16	1.27
7	4	0.30	0.67	0.6	0.8	1.70	1.39	2.32	0.73
8	4	0.25	0.75	0.4	0.6	2.32	0.70	1.63	1.43
9	4	0.40	1.00	0.4	0.6	2.79	0.00	0.93	3.00
10	4	0.30	0.75	0.6	0.6	2.09	1.39	1.39	1.50
12	4	0.25	0.67	0.4	0.4	1.93	0.70	0.70	2.78
13	4	0.25	0.33	0.4	0.6	0.39	0.70	1.63	0.24
14	4	0.45	1.00	0.6	0.8	2.55	0.70	1.63	1.57
15	3	0.20	0.67	0.6	0.6	2.17	1.86	1.86	1.17
16	4	0.35	0.67	0.6	0.6	1.47	1.16	1.16	1.27
17	4	0.20	0.75	0.4	0.4	2.55	0.93	0.93	2.75
18	2	0.40	0.67	0.4	0.4	1.24	0.00	0.00	NA
19	3	0.47	1.00	0.6	0.8	2.48	0.62	1.55	1.60
21	3	0.47	0.67	0.6	0.6	0.93	0.62	0.62	1.50
22	4	0.50	1.00	0.6	0.8	2.32	0.46	1.39	1.67
23	3	0.53	1.00	1.0	1.0	2.17	2.17	2.17	1.00
24	4	0.15	0.67	0.4	0.4	2.40	1.16	1.16	2.07
25	4	0.40	0.67	0.6	0.6	1.24	0.93	0.93	1.33
26	4	0.45	0.75	0.6	0.8	1.39	0.70	1.63	0.86
28	4	0.40	0.75	0.8	0.8	1.63	1.86	1.86	0.88
29	4	0.40	0.50	0.6	0.6	0.46	0.93	0.93	0.50
30	4	0.50	1.00	0.8	1.0	2.32	1.39	2.32	1.00
31	3	0.40	1.00	0.8	1.0	2.79	1.86	2.79	1.00
32	4	0.30	1.00	0.6	0.8	3.25	1.39	2.32	1.40
33	4	0.30	1.00	0.8	0.8	3.25	2.32	2.32	1.40
34	4	0.25	1.00	0.4	0.6	3.48	0.70	1.63	2.14
35	4	0.60	1.00	0.8	1.0	1.86	0.93	1.86	1.00
36	4	0.20	0.33	0.4	0.6	0.62	0.93	1.86	0.33
37	4	0.30	1.00	0.6	0.8	3.25	1.39	2.32	1.40
38	2	0.30	0.33	0.4	0.6	0.15	0.46	1.39	0.11
39	4	0.55	1.00	0.8	1.0	2.09	1.16	2.09	1.00
40	4	0.35	0.75	0.6	0.8	1.86	1.16	2.09	0.89
41	4	0.50	1.00	0.8	0.8	2.32	1.39	1.39	1.67
42	4	0.10	0.67	0.2	0.2	2.63	0.46	0.46	5.67
43	4	0.10	0.67	0.2	0.4	2.63	0.46	1.39	1.89
44	3	0.33	1.00	0.6	0.6	3.10	1.24	1.24	2.50
45	4	0.40	1.00	0.6	1.0	2.79	0.93	2.79	1.00
46	4	0.20	0.50	0.2	0.4	1.39	0.00	0.93	1.50
47	3	0.47	0.75	0.6	0.8	1.32	0.62	1.55	0.85
48	4	0.30	1.00	0.6	0.6	3.25	1.39	1.39	2.33
49	4	0.40	0.67	0.6	0.8	1.24	0.93	1.86	0.67

Group	n	IndivA	GroupB	TopC	SuperD	GainBA	TopSurplus	SuperSurplus	CollabEfficiency
50	4	0.40	1.00	0.6	1.0	2.79	0.93	2.79	1.00
52	4	0.40	0.67	0.6	0.6	1.24	0.93	0.93	1.33
53	4	0.30	0.67	0.4	0.6	1.70	0.46	1.39	1.22
54	3	0.33	1.00	0.4	0.8	3.10	0.31	2.17	1.43
55	4	0.40	0.50	0.6	0.6	0.46	0.93	0.93	0.50
56	4	0.35	1.00	0.6	0.6	3.02	1.16	1.16	2.60
57	4	0.45	0.50	0.6	0.8	0.23	0.70	1.63	0.14
58	4	0.55	1.00	0.6	1.0	2.09	0.23	2.09	1.00
59	4	0.25	1.00	0.6	0.6	3.48	1.63	1.63	2.14
60	3	0.47	0.67	0.6	0.6	0.93	0.62	0.62	1.50
61	4	0.55	1.00	1.0	1.0	2.09	2.09	2.09	1.00
62	3	0.47	0.67	0.6	0.8	0.93	0.62	1.55	0.60
63	4	0.50	0.75	0.6	0.8	1.16	0.46	1.39	0.83
64	4	0.30	1.00	0.4	0.8	3.25	0.46	2.32	1.40
65	4	0.45	0.67	0.6	0.6	1.01	0.70	0.70	1.44
66	4	0.35	0.75	0.6	0.6	1.86	1.16	1.16	1.60
67	3	0.27	1.00	0.4	0.6	3.41	0.62	1.55	2.20
68	4	0.45	1.00	0.6	0.8	2.55	0.70	1.63	1.57
69	2	0.50	1.00	0.6	0.6	2.32	0.46	0.46	5.00
70	4	0.25	0.25	0.4	0.4	0.00	0.70	0.70	0.00
71	4	0.35	1.00	0.6	0.8	3.02	1.16	2.09	1.44
72	4	0.25	0.75	0.4	0.6	2.32	0.70	1.63	1.43

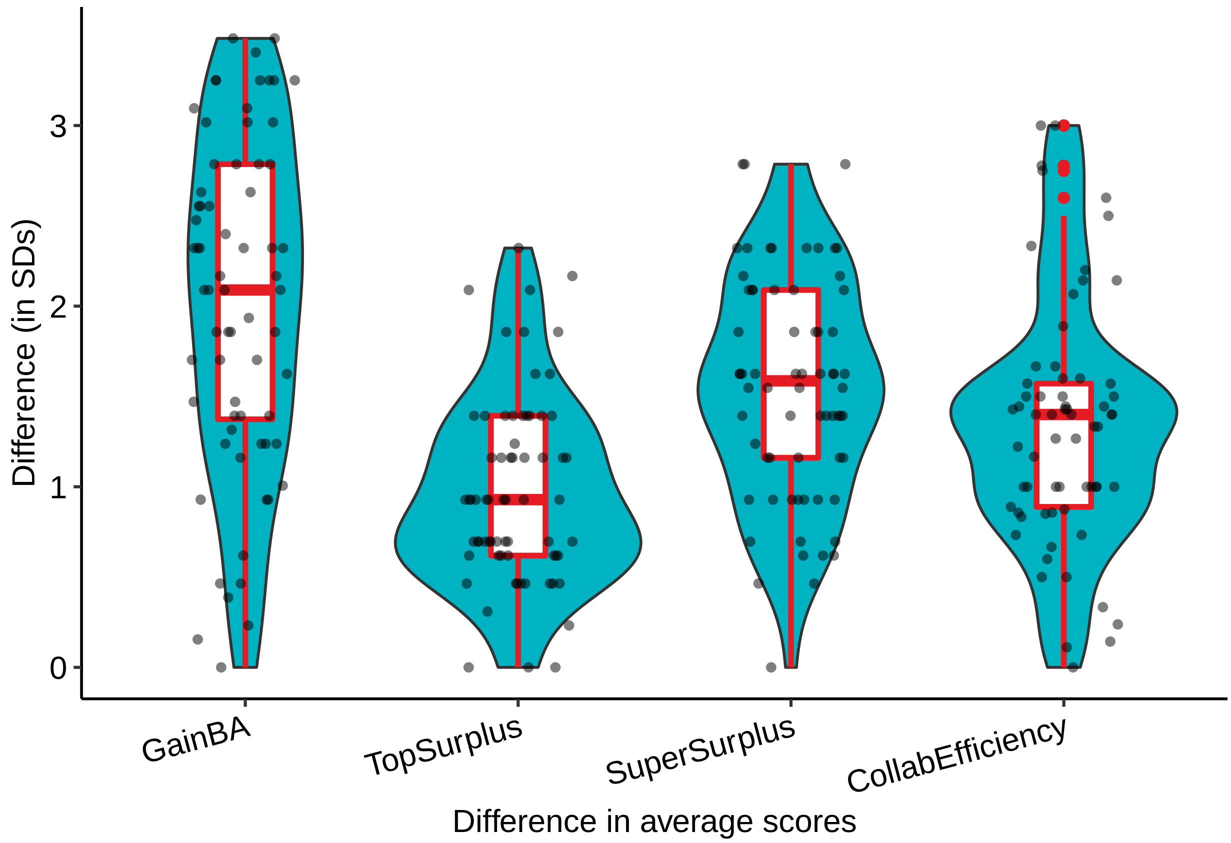
```r
ggplot(stack(LevyByGroup %>% select(IndivA,GroupB,TopC,SuperD)), aes(x = ind, y = values)) +
  geom_violin(fill=heathers[1]) +
  geom_boxplot(width=0.2,color=heathers[2],lwd=1) +
  geom_jitter(shape=16, position=position_jitter(0.2),alpha=0.5) +
  labs(x = "Average score of...",
       y = "Percentage correct") +
  scale_y_continuous(labels = scales::percent)
```




```
ggsave("Figs/Study3_LevyABCD.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study3_LevyABCD_small.pdf",width=10,height=7,units="cm",dpi=300)

LevyByGroup %>% select(GainBA,TopSurplus,SuperSurplus,CollabEfficiency) %>%
  filter(CollabEfficiency>4)
```

```
## # A tibble: 2 x 4
##   GainBA TopSurplus SuperSurplus CollabEfficiency
##    <dbl>      <dbl>        <dbl>            <dbl>
## 1   2.63      0.464        0.464             5.67
## 2   2.32      0.464        0.464             5.
```

```
ggplot(stack(LevyByGroup %>% select(GainBA,TopSurplus,SuperSurplus,CollabEfficiency) %>%
               mutate(CollabEfficiency = ifelse(CollabEfficiency<4,CollabEfficiency,NA_real_))), aes(x =
  geom_violin(fill=heathers[1]) +
  geom_boxplot(width=0.2,color=heathers[2],lwd=1) +
  geom_jitter(shape=16, position=position_jitter(0.2),alpha=0.5) +
  labs(x = "Difference in average scores",
       y = "Difference (in SDs)")+
  theme(axis.text.x = element_text(angle = 15, hjust = 1))
```




```r
ggsave("Figs/Study3_LevyDiffs.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study3_LevyDiffs_small.pdf",width=10,height=7,units="cm",dpi=300)


LevyByGroup %>%
  summarise(
    CollabEfficiency_m = mean(CollabEfficiency, na.rm=TRUE),
    CollabEfficiency_sd = sd(CollabEfficiency, na.rm=TRUE),
    n=n()
  ) %>% knitr::kable(digits = 2)
```

CollabEfficiency_m	CollabEfficiency_sd	n
1.46	0.96	68

```r
LevyByGroup %>%
  filter(CollabEfficiency>1) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   154
```

Bayesian analysis

Here we look at (and compare) the proportions in the 6 groups shown in Table 5 of the paper.

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

```
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)


myData = S123data %>%
  select(ZippGroup, Stage3score) %>%
  mutate(
    ZippGroup = as.numeric(str_sub(ZippGroup,2,2))
  )

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)
```

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.

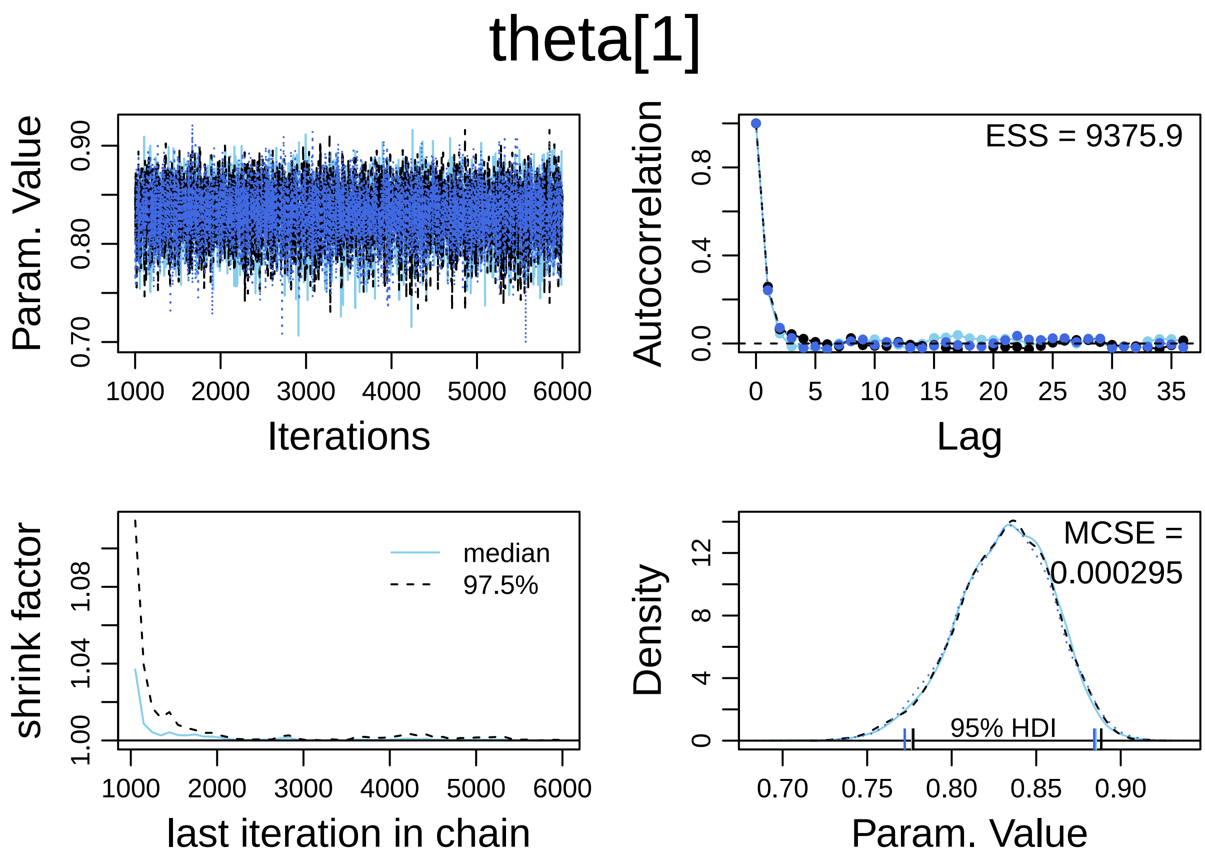
```r
samp_mat <- as.matrix(samplesS3)
print(quantile(samp_mat[, "theta[2]"] - samp_mat[, "theta[1]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.02627627  0.03463602  0.10322755
```

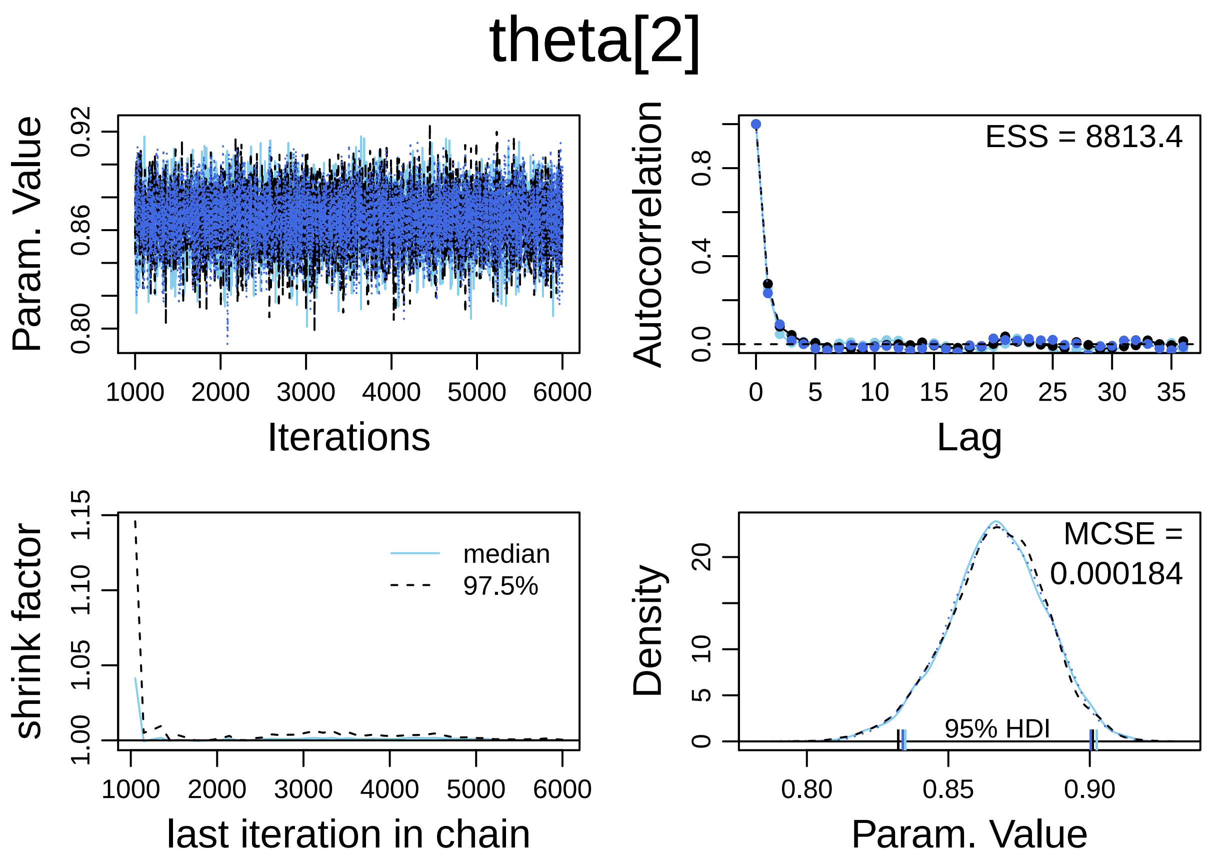
```r
print(quantile(samp_mat[, "theta[4]"] - samp_mat[, "theta[3]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.04878312  0.09406176  0.37668576
```

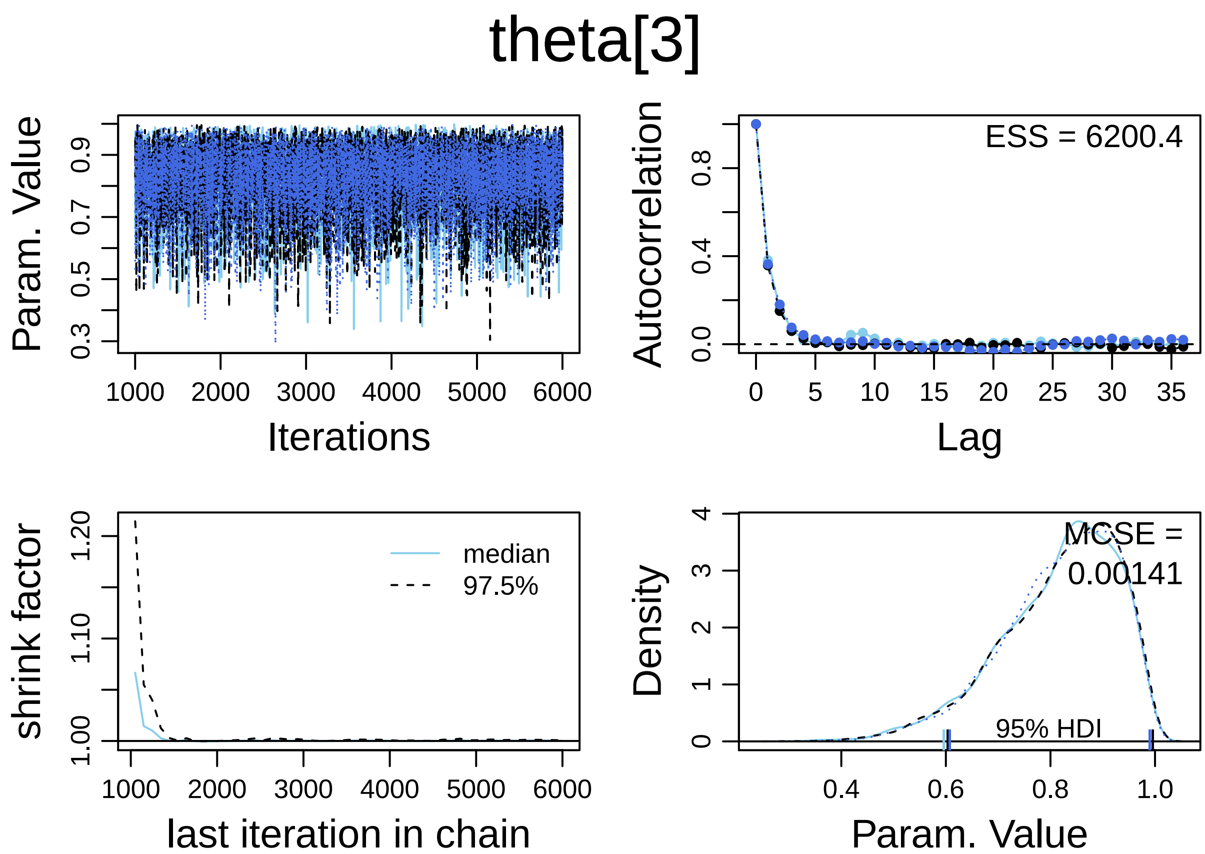
```r
diagMCMC(samplesS3, parName = "theta[1]", saveName="Figs/Study3_S3props", saveType = "pdf")
```



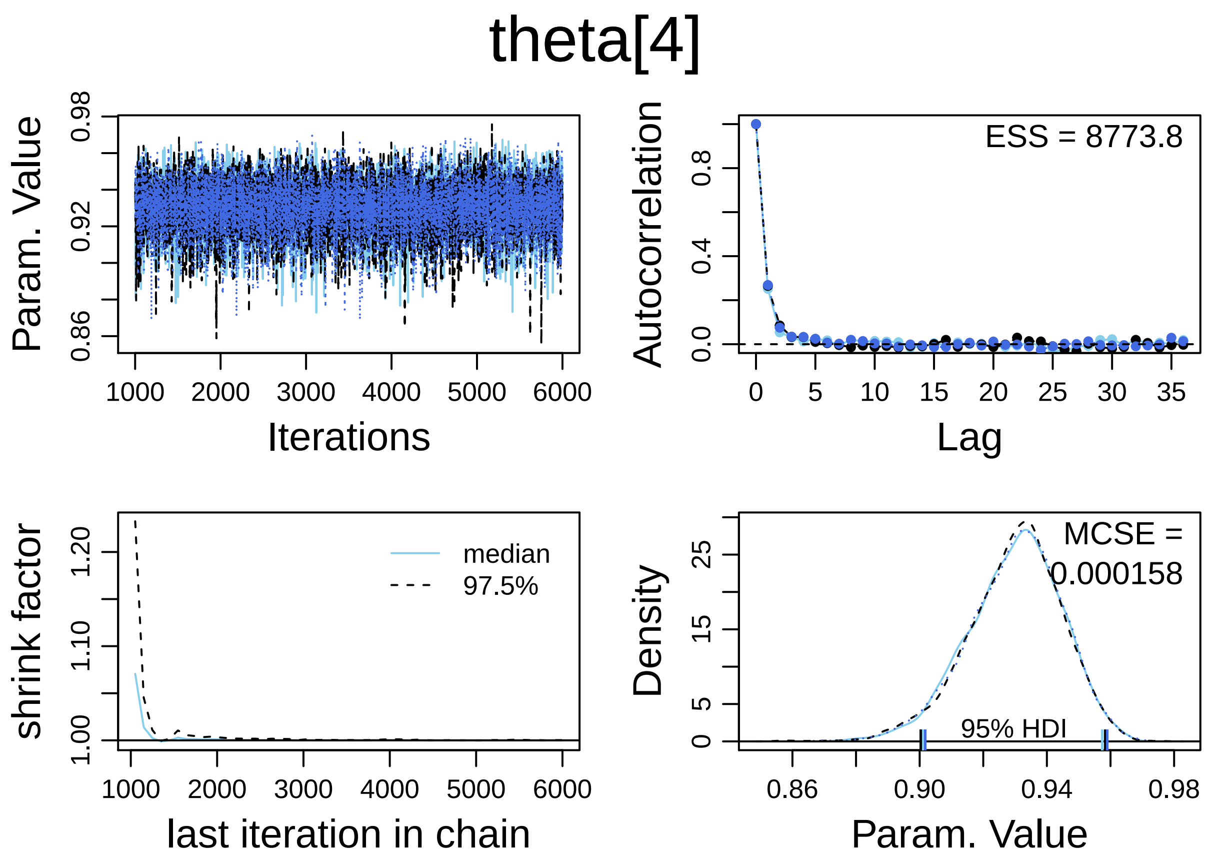

```r
diagMCMC(samplesS3, parName = "theta[2]", saveName="Figs/Study3_S3props", saveType = "pdf")
```



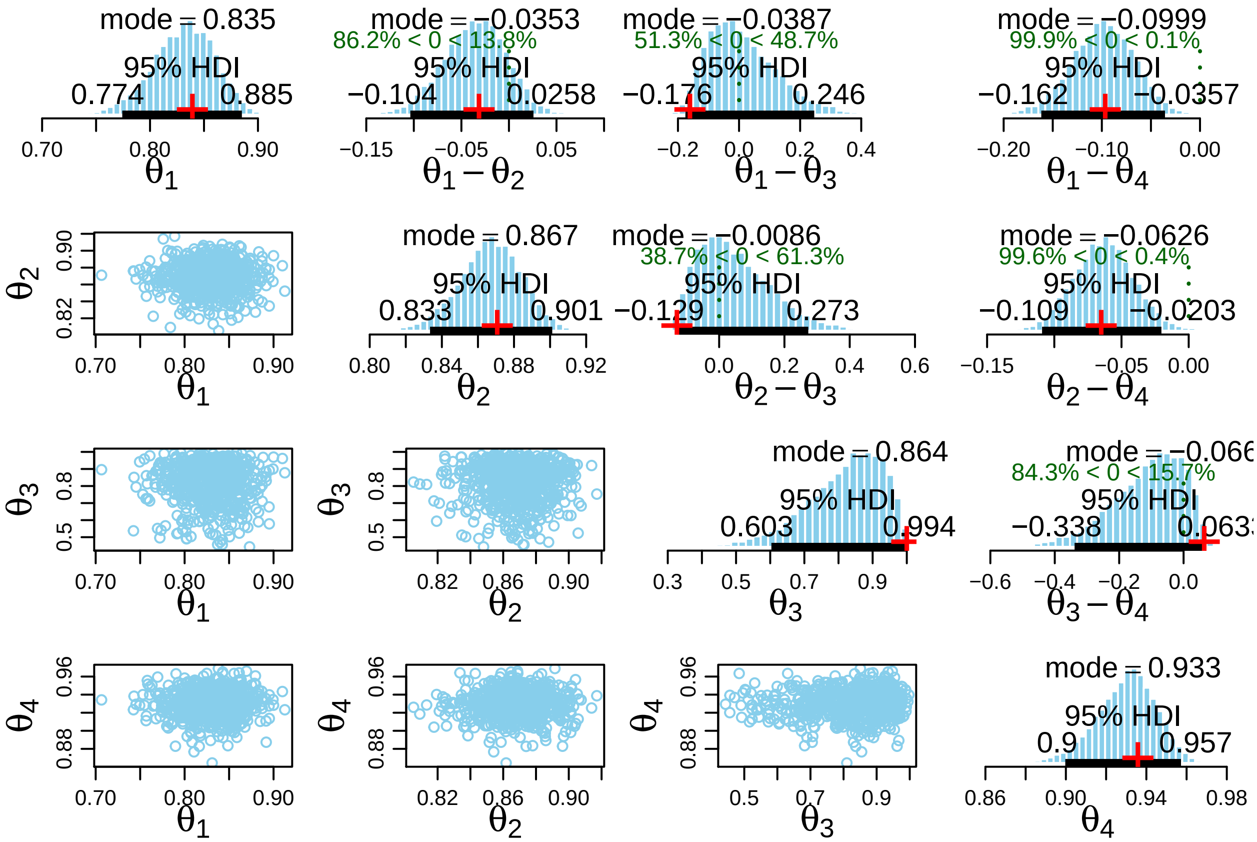

```
diagMCMC(samplesS3, parName = "theta[3]", saveName="Figs/Study3_S3props", saveType = "pdf")
```



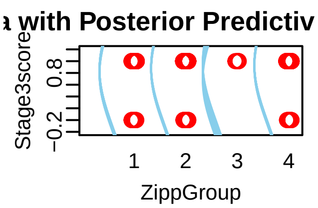

```r
diagMCMC(samplesS3, parName = "theta[4]", saveName="Figs/Study3_S3props", saveType = "pdf")
```




```
plotMCMC( samplesS3, data=myData, yName="Stage3score", sName="ZippGroup", compVal=NULL, compValDiff=0.0
          saveName="Figs/Study3_S3props", saveType = "pdf")
```



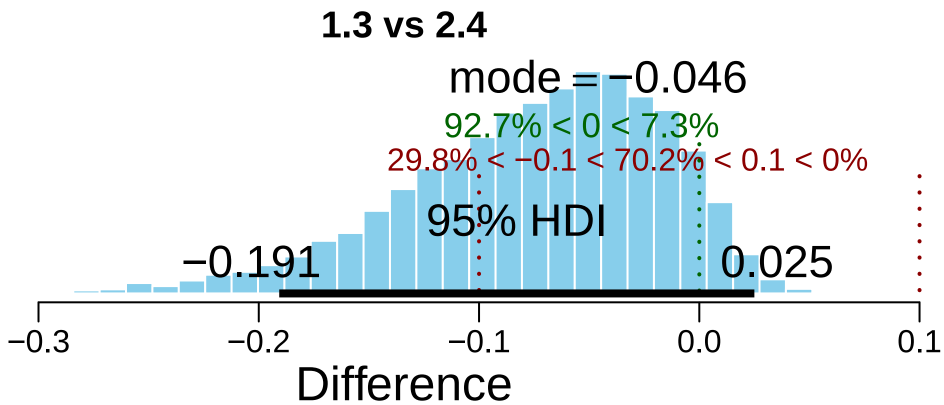

```
contrasts = list(
  list( c(1,3), c(2,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(3,4), compVal=0.0, ROPE=c(-0.1,0.1))
)
plotMCMCwithContrasts( samplesS3, datFrm=data.frame(myData), yName="Stage3score", xName="ZippGroup", co
          saveName="Figs/Study3_S3props", saveType = "pdf")
```

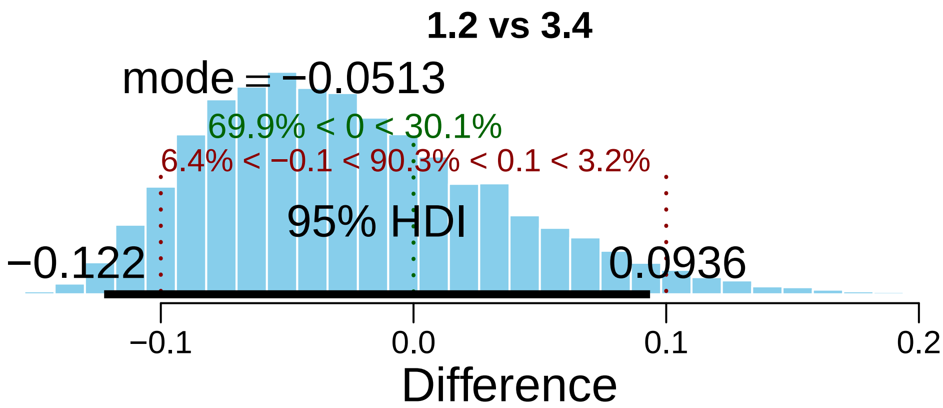



```
## [1] 1
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 2
## [1] 1
## [1] 790
## [1] 1580
## [1]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





Experimental analysis

This conducts the Bayesian analysis of the main experiment.

```
myData = data %>%
  dplyr::select(ZippGroup, Stage3score) %>%
  drop_na() %>%
  mutate(ZippGroup = fct_relevel(ZippGroup, "C0", after=Inf)) %>%
  mutate(ZippGroup = fct_relevel(ZippGroup, "C1", after=Inf)) %>%
  mutate(
    ZippGroup = case_when(
      ZippGroup=="C0" ~ 5,
      ZippGroup=="C1" ~ 6,
      TRUE ~ as.numeric(str_sub(ZippGroup,2,2))
    )
  )

myData %>% group_by(ZippGroup) %>% summarise( mean(Stage3score)) %>% knitr::kable()
```

ZippGroup	mean(Stage3score)
1	0.8392857
2	0.8707124
3	1.0000000
4	0.9358108
5	0.8705882

ZippGroup	mean(Stage3score)
6	0.9400000

```r
zipptab = myData %>%
  group_by(ZippGroup) %>%
  summarise(
    numcorrect = sum(Stage3score),
    numingroup = n()
  )
zipptab %>% knitr::kable()
```

ZippGroup	numcorrect	numingroup
1	141	168
2	330	379
3	7	7
4	277	296
5	222	255
6	141	150

```r
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
modelEXPT <- jags.model(textConnection(model_string), data = list(x = zipptab$numcorrect, n = zipptab$nu
                       n.chains = 3, n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 12
##    Total graph size: 25
##
## Initializing model
```

```r
samplesEXPT <- coda.samples(modelEXPT, 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.
```

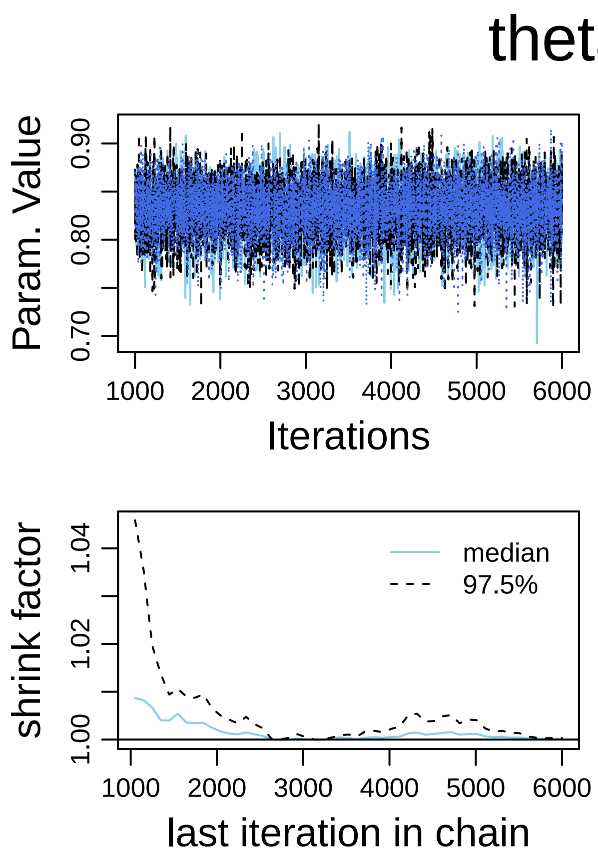
```r
samp_mat <- as.matrix(samplesEXPT)
print(quantile(samp_mat[, "theta[2]"] - samp_mat[, "theta[1]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.02836855  0.03491845  0.10216618
```

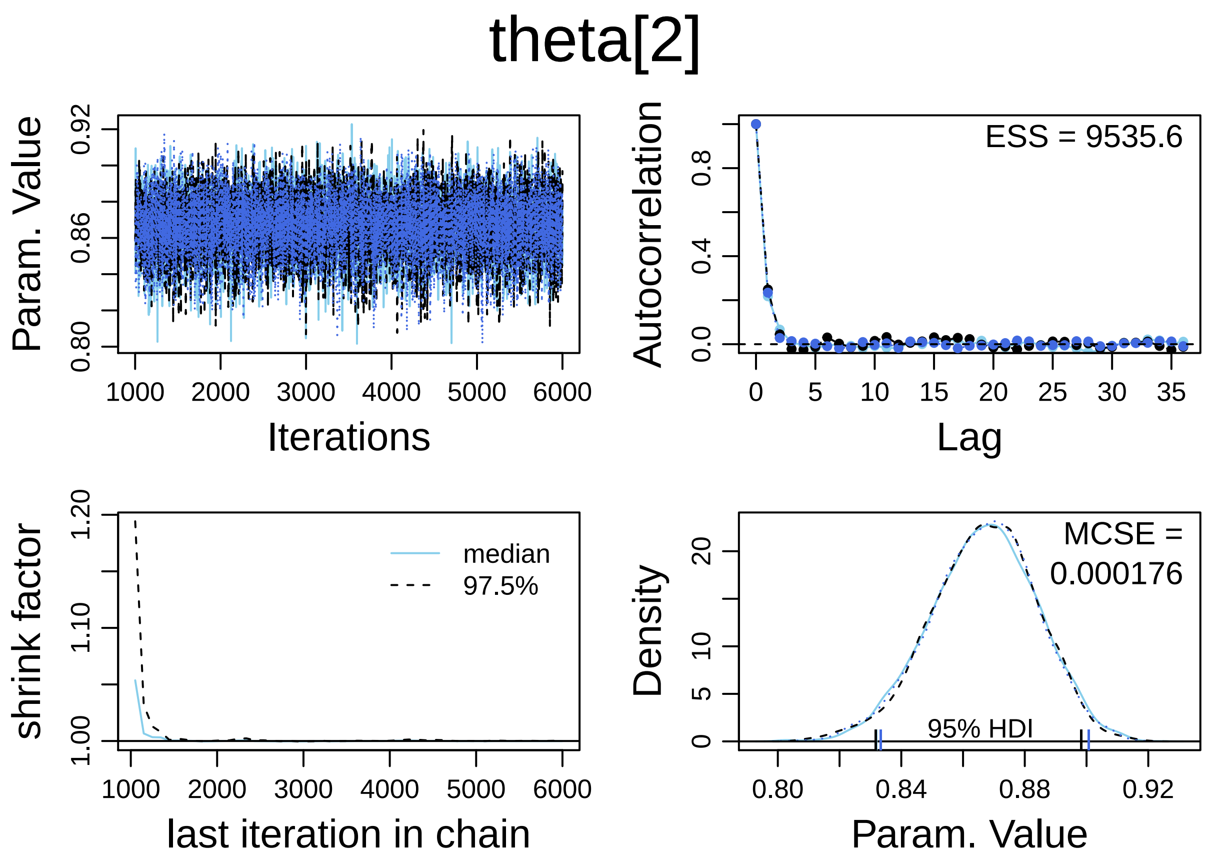
```r
print(quantile(samp_mat[, "theta[4]"] - samp_mat[, "theta[3]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.04819208  0.09248183  0.37211846
```

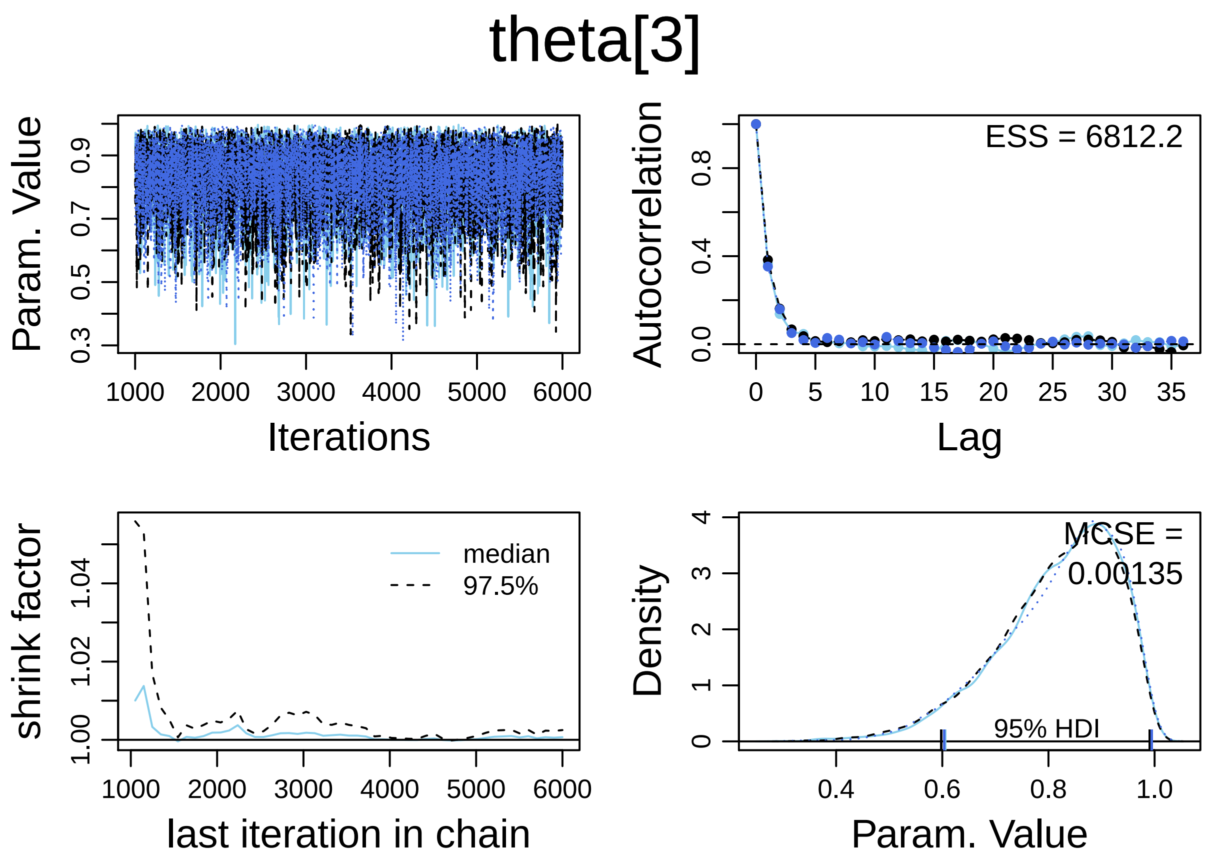
```r
diagMCMC(samplesEXPT, parName = "theta[1]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



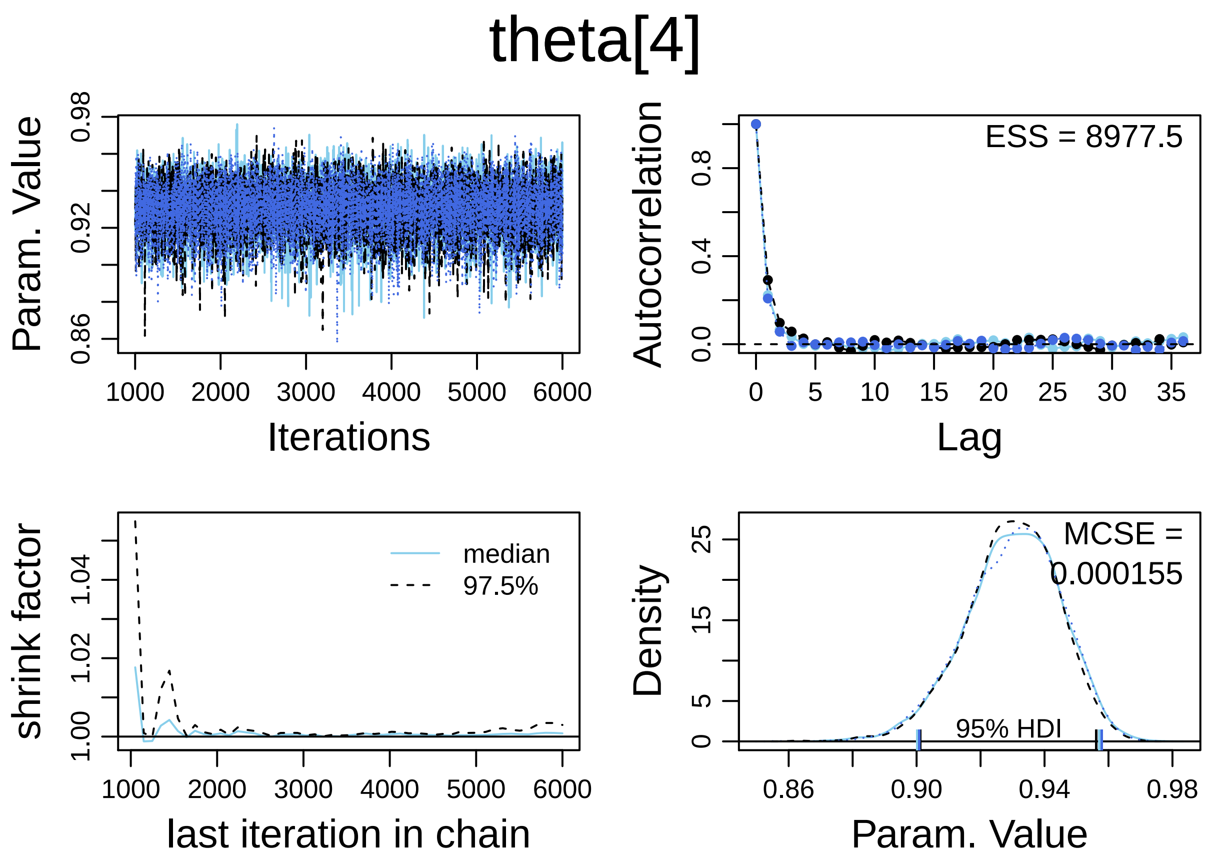

```r
diagMCMC(samplesEXPT, parName = "theta[2]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



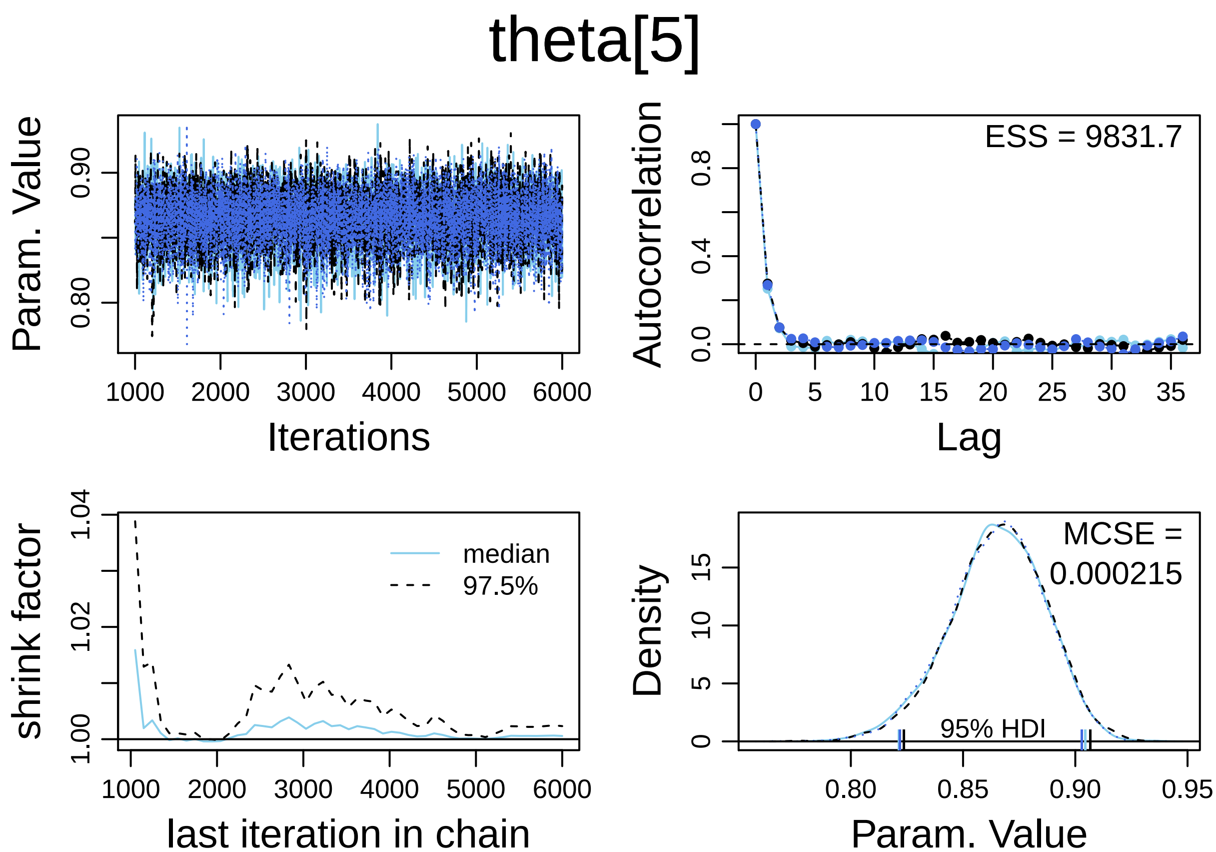

```r
diagMCMC(samplesEXPT, parName = "theta[3]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



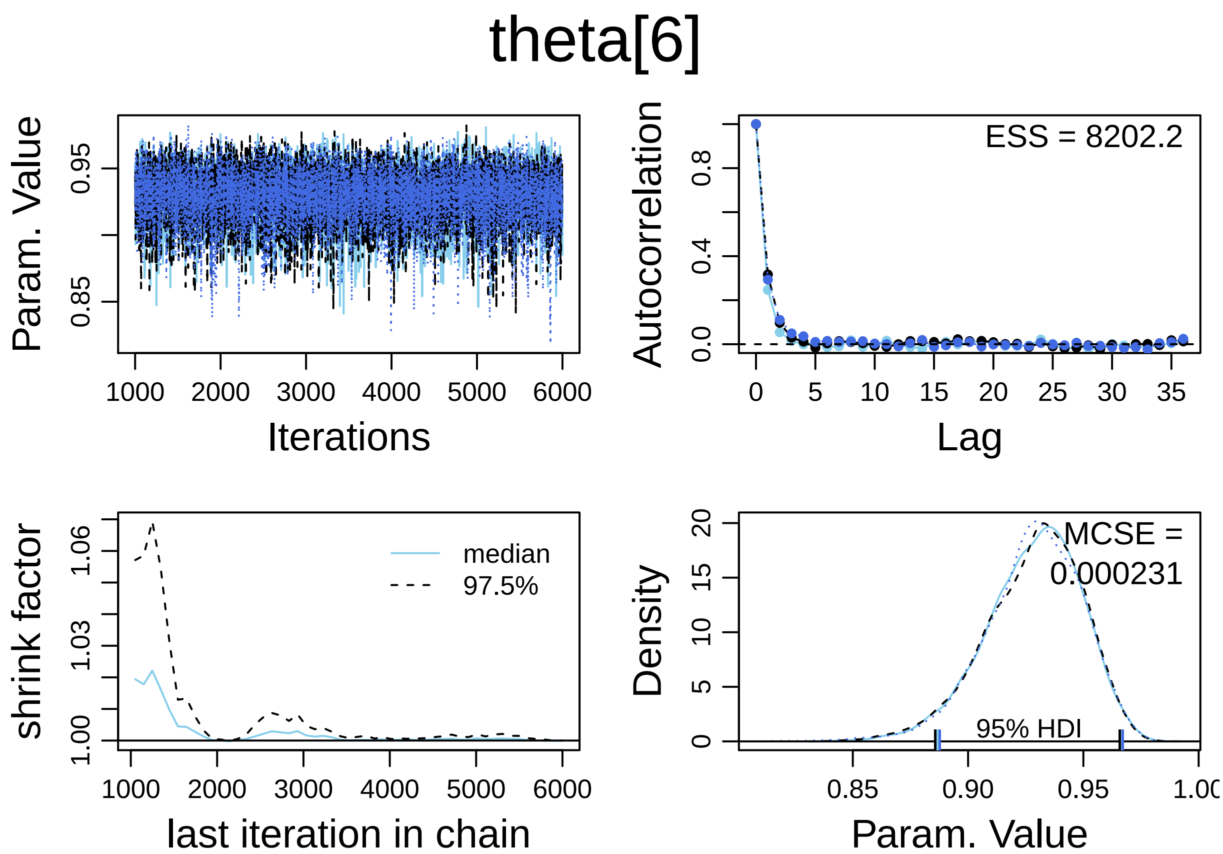

```
diagMCMC(samplesEXPT, parName = "theta[4]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



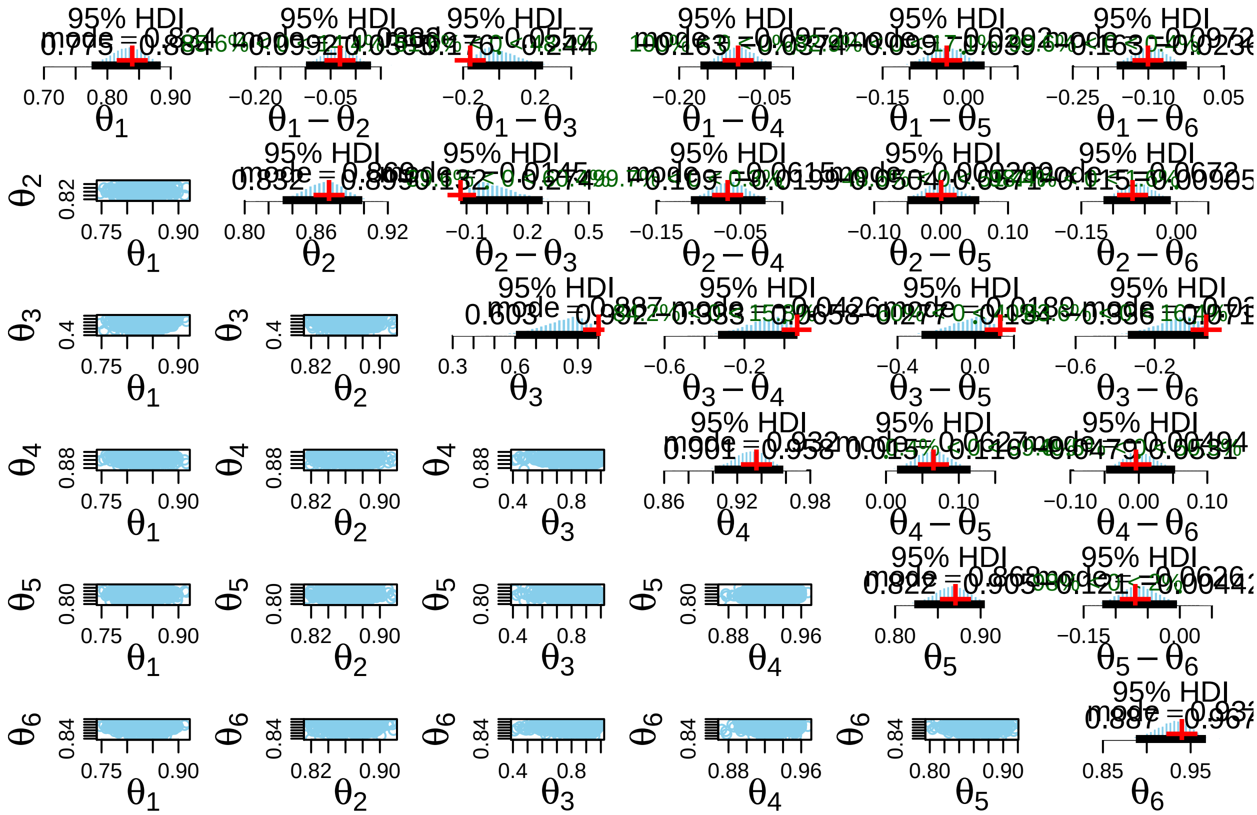

```
diagMCMC(samplesEXPT, parName = "theta[5]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



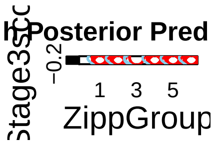

```r
diagMCMC(samplesEXPT, parName = "theta[6]", saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```




```
#plotPost( samplesS3[,"theta[1]"] , main="theta[1]" , xlab=bquote(theta[1]) )
#plotPost( samplesS3[,"theta[2]"] , main="theta[2]" , xlab=bquote(theta[2]) )
#plotPost( samplesS3[,"theta[3]"] , main="theta[3]" , xlab=bquote(theta[3]) )
#plotPost( samplesS3[,"theta[4]"] , main="theta[4]" , xlab=bquote(theta[4]) )

plotMCMC( samplesEXPT, data=myData, yName="Stage3score", sName="ZippGroup", compVal=NULL, compValDiff=0
          saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



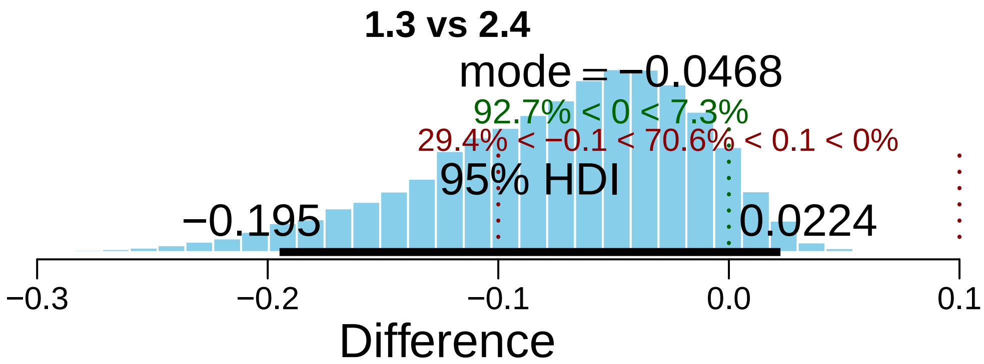
```
contrasts = list(
  list( c(1,3), c(2,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(3,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2,3,4), c(5,6), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(5), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(3,4), c(6), compVal=0.0, ROPE=c(-0.1,0.1))
)

plotMCMCwithContrasts( samplesEXPT, datFrm=data.frame(myData), yName="Stage3score", xName="ZippGroup", 
                       saveName="Figs/Study3_EXPTprops", saveType = "pdf")
```



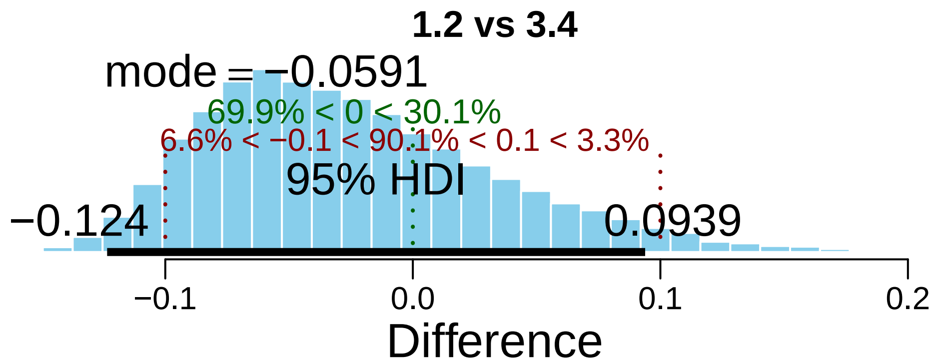

```
## [1] 1
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 2
## [1] 1
## [1] 790
## [1] 1580
## [1]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

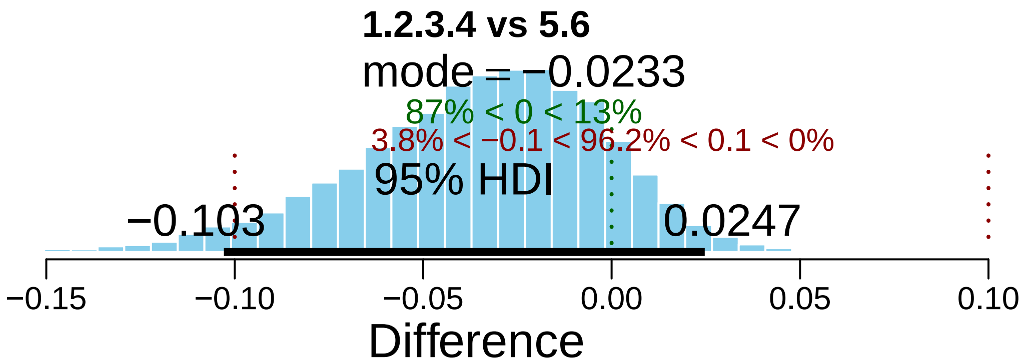
```
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 3
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [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] 5
## [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] 6
## [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
```

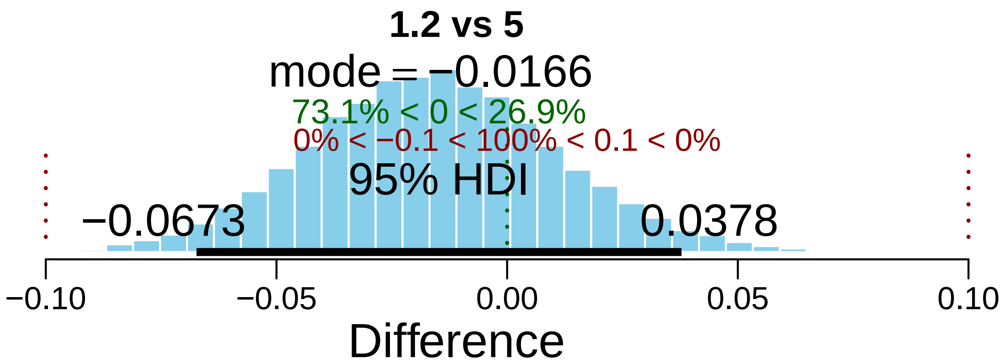


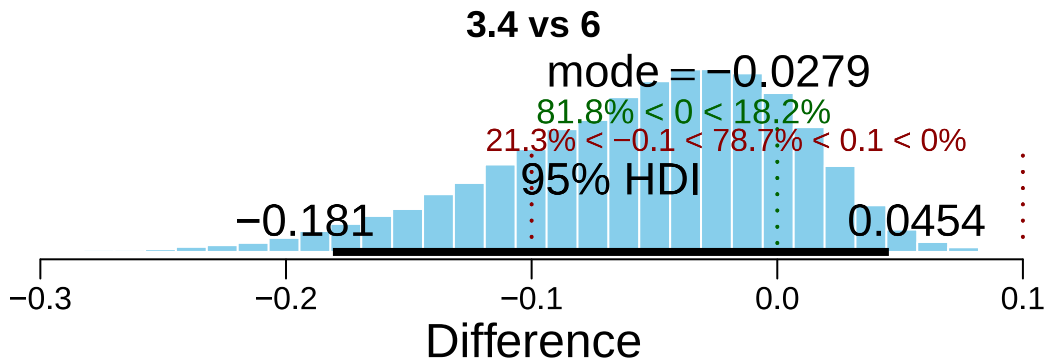
1.3 vs 2.4
mode = −0.0468
92.7% < 0 < 7.3%
29.4% < −0.1 < 70.6% < 0.1 < 0%
95% HDI
−0.195
0.0224
−0.3
−0.2
−0.1
0.0
0.1
Difference


1.2 vs 3.4
mode = −0.0591
69.9% < 0 < 30.1%
6.6% < −0.1 < 90.1% < 0.1 < 3.3%
95% HDI
−0.124
0.0939
−0.1
0.0
0.1
0.2
Difference



1.2.3.4 vs 5.6
mode = −0.0233
87% < 0 < 13%
3.8% < −0.1 < 96.2% < 0.1 < 0%
95% HDI
−0.103
0.0247
−0.15
−0.10
−0.05
0.00
0.05
0.10
Difference


1.2 vs 5
mode = −0.0166
73.1% < 0 < 26.9%
0% < −0.1 < 100% < 0.1 < 0%
95% HDI
−0.0673
0.0378
−0.10
−0.05
0.00
0.05
0.10
Difference


3.4 vs 6
mode = −0.0279
81.8% < 0 < 18.2%
21.3% < −0.1 < 78.7% < 0.1 < 0%
95% HDI
−0.181
0.0454
−0.3
−0.2
−0.1
0.0
0.1
Difference