Two-stage exams: Study 2

George Kinnear

20/06/2020

Contents

Data

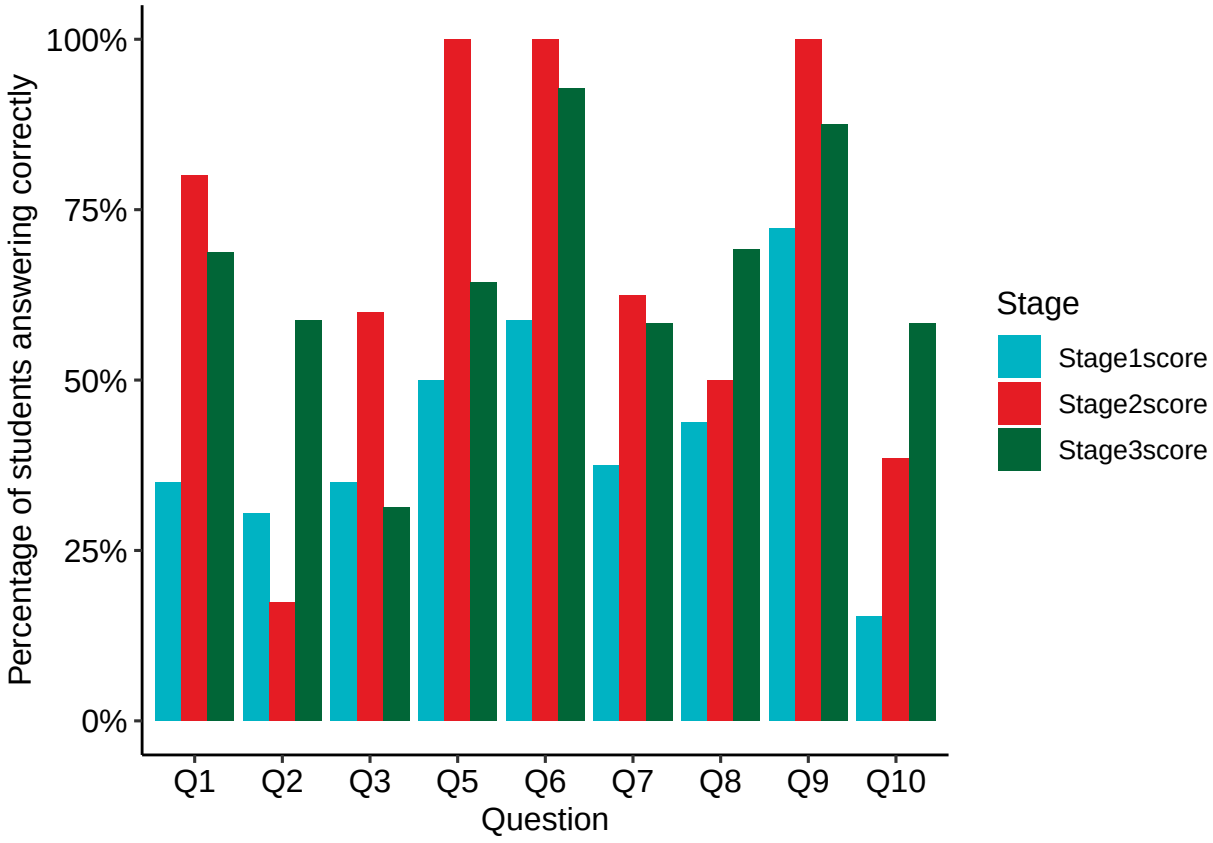
Import and combine the datasets.

```
data = read.csv('Study2_data.csv', header=T) %>%
  dplyr::select(-c(X)) %>%
  mutate(Q = fct_relevel(Q, "Q10", after=Inf))


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

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

View a summary of the data

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

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

plotData <- aggregate(ld$score,
                      by = list(Q = ld$Q, Stage = ld$stage),
```

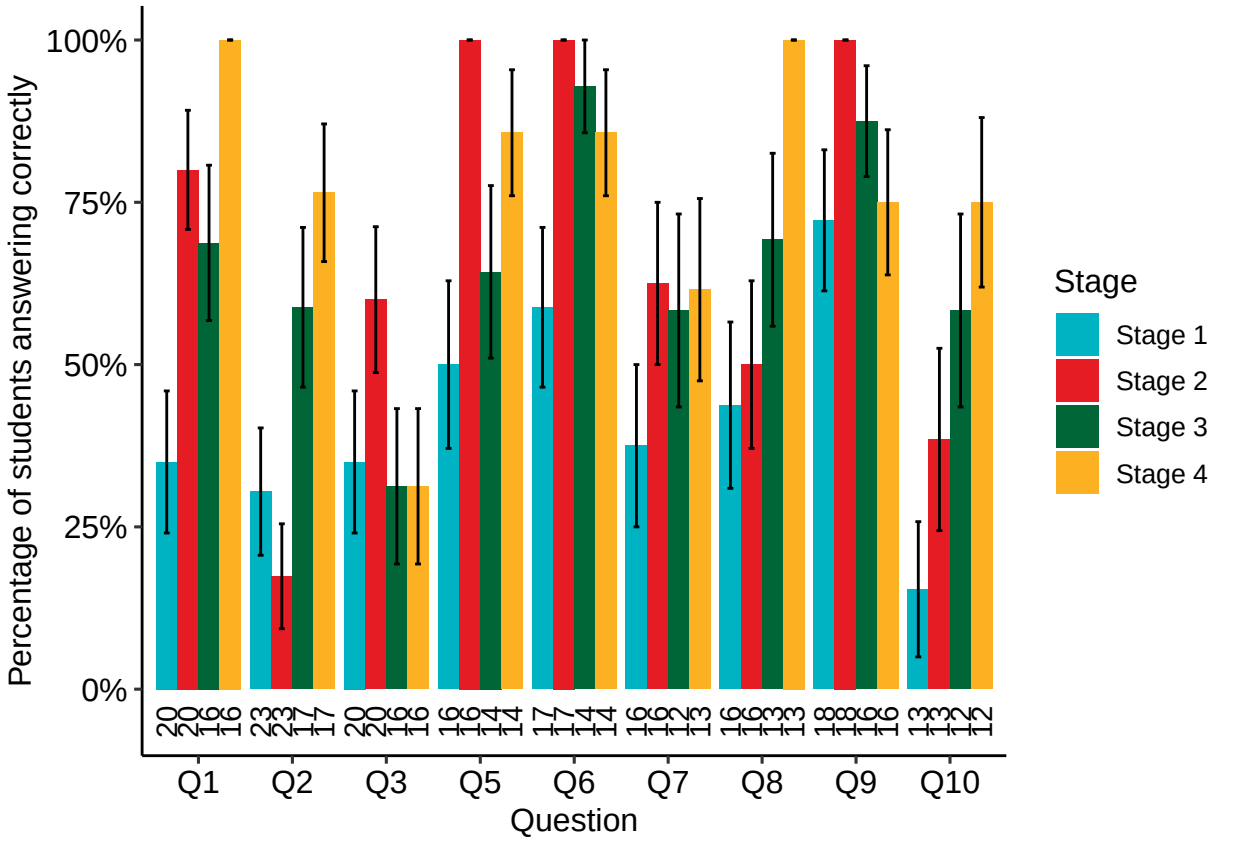
```
                      FUN = function(x) c(mean = mean(x), sd = sd(x),
                                          n = length(x)))
plotData <- do.call(data.frame, plotData)
plotData$se <- plotData$x.sd / sqrt(plotData$x.n)

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

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))) +
  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) +
  scale_y_continuous(labels = scales::percent)
p
```



Mean at each stage (as bars)

```
ld <- gather(data = S1234data,
             key = stage,
```

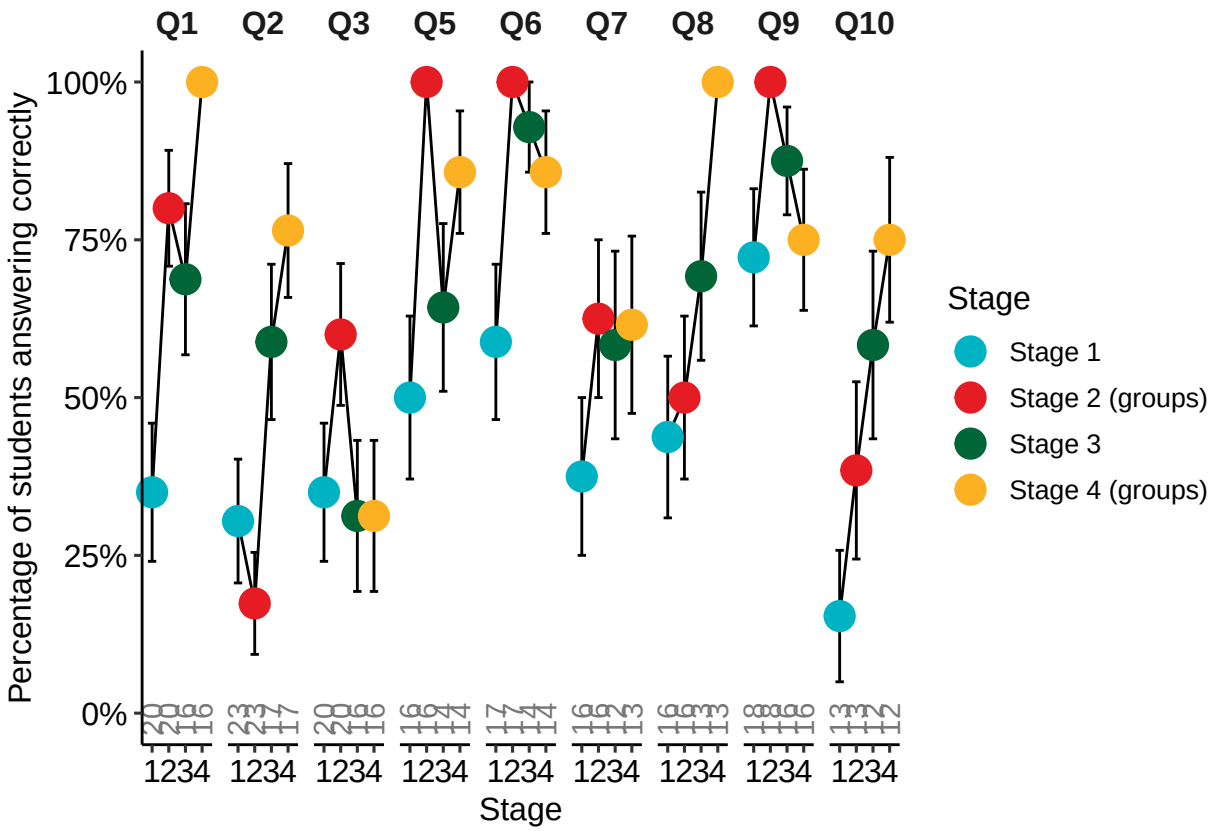
```
             value = score,
             Stage1score, Stage2score, Stage3score, Stage4score)
ld <- ld[complete.cases(ld),]

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(
    Stage = gsub(".*(\\d).*","\\1",Stage)  # alternatively: Stage = parse_number(Stage)
  ) %>%
  arrange(Q,Stage)

plotCounts = plotData %>% group_by(Q,Stage) %>% dplyr::select(Stage,n) %>%
  mutate(scale_lab = paste0("Stage ",Stage, " (n=",n,")")) %>%
  ungroup

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

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)
```




```
ggsave("Figs/Study2_S1234_means.pdf",width=20,height=10,units="cm",dpi=300)
```

Mean at each stage (as points)

```
ggplot(data=plotData,aes(x=Stage,y=mean,group=Q,label=n))+
  geom_line()+
  geom_errorbar(aes(ymax = plotData$mean + plotData$se,
                    ymin = plotData$mean - plotData$se),
                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=paste0("Stage ",c(1:4),rep(c("","  (groups)"),2))) +
  labs(x = "Stage",
       y = "Percentage of students answering correctly",
       fill = "Stage") +
  scale_y_continuous(labels = scales::percent)+
  facet_grid(cols=vars(Q))+
  coord_cartesian(ylim=c(0,1),clip="off")+
  geom_text(position = position_dodge(width = 0.9),
            aes(y=-0.01, label=paste0("",plotCounts$n)),
            angle=90,
            color="#777777") +
  theme(strip.background = element_rect(fill=NA,colour = NA),
        strip.text = element_text(size=12, face="bold"))
```




```
ggsave("Figs/Study2_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	Q1	Q2	Q3	Q5	Q6	Q7	Q8	Q9	Q10
1	35 (10.9)	30 (9.8)	35 (10.9)	50 (12.9)	59 (12.3)	38 (12.5)	44 (12.8)	72 (10.9)	15 (10.4)
2	80 (9.2)	17 (8.1)	60 (11.2)	100 (0.0)	100 (0.0)	62 (12.5)	50 (12.9)	100 (0.0)	38 (14.0)
3	69 (12.0)	59 (12.3)	31 (12.0)	64 (13.3)	93 (7.1)	58 (14.9)	69 (13.3)	88 (8.5)	58 (14.9)
4	100 (0.0)	76 (10.6)	31 (12.0)	86 (9.7)	86 (9.7)	62 (14.0)	100 (0.0)	75 (11.2)	75 (13.1)

Forming the Zipp tables

This constructs the data in Table 4 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,")"))
print("Entries of the Zipp table for Stage 3 only")
```

```
## [1] "Entries of the Zipp table for Stage 3 only"
```

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

ZippGroup	numcorrect	numingroup	pc	entry
1	17	36	0.4722222	47.2 (17/36)
2	24	38	0.6315789	63.2 (24/38)
3	4	6	0.6666667	66.7 (4/6)
4	40	50	0.8000000	80.0 (40/50)

Group dynamics: Stage 1 vs Stage 2

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

This is discussed in section 5.2.2 of the paper.

```
dataStudents = read.csv('Study2_data_raw_stage1.csv', header=T)
dataGroups = read.csv('Study2_data_raw_stage2.csv', header=T)

responses = dataStudents %>%
  select(AnonKey, contains("Q")) %>%
  gather(key,Response,-AnonKey) %>%
  tidyr::extract(key, c("Week", "Question"), regex="Wk(.*)Q(.*)")

qnkey = responses %>%
  filter(AnonKey=="KEY")

studentGroups = dataStudents %>%
  filter(AnonKey != "KEY") %>%
  select(AnonKey, contains("Gp")) %>%
  gather(key,Group,-AnonKey) %>%
  tidyr::extract(key,"Week",regex="Wk(.*)Gp")

groupResponses = dataGroups %>%
  gather(Question,Response,-Group) %>%
  tidyr::extract(Question,"Question",regex="Q(.*)")

S12data = responses %>%
  filter(AnonKey!="KEY") %>%
  left_join(studentGroups,by=c("AnonKey","Week")) %>%
  rename( Stage1response = Response ) %>%
  left_join(groupResponses,by=c("Group","Question")) %>%
  rename( Stage2response = Response) %>%
  filter(Stage1response!="")
```

```
# Check if the ultimate response is correct
S12data_scored = S12data %>%
  left_join( qnkey %>% select(-AnonKey) %>% rename(Key=Response), c("Week","Question")) %>%
  mutate(
    Stage1score = case_when(
      Stage1response=="" ~ NA_real_,
      Stage1response==Key ~ 1,
      Stage1response!=Key ~ 0,
      TRUE ~ NA_real_
    ),
    Stage2score = case_when(
      Stage2response=="" ~ NA_real_,
      Stage2response==Key ~ 1,
      Stage2response!=Key ~ 0,
      TRUE ~ NA_real_
    ),
    Stage2scorePartial = case_when(
      Stage2response=="" | is.na(Stage2response) ~ NA_real_,
      Stage2response==Key ~ 1,
      str_length(Stage2response)==2 & str_sub(Stage2response,2,2)==Key ~ 0.5,
      TRUE ~ 0
    )
  ) %>%
  select(-Key) %>%
  rename(Student=AnonKey)

S1modalresponse = S12data_scored %>%
  group_by(Group,Question,Stage1response) %>%
  tally() %>%
  arrange(Group,Question,-n) %>%
  mutate(
    nextmost = lead(n),
    topmost = max(n),
    ismodal = ifelse(n!=nextmost & n==topmost,1,0)
  ) %>%
  filter(ismodal==1) %>%
  select(Group,Question,Stage1response) %>%
  rename(S1ModalResponse = Stage1response)

groupPerf = S12data_scored %>%
  group_by(Week,Group,Question) %>%
  summarise(
    n=length(Student),
    tot=sum(Stage1score),
    mean=mean(Stage1score),
    Responses = str_c(Stage1response,collapse = "")
  ) %>%
  left_join(S12data_scored %>% select("Group","Question","Stage2response",
                                      "Stage2score","Stage2scorePartial") %>% distinct(),
            by=c("Group","Question")) %>%
  left_join(S1modalresponse) %>%
  ungroup() %>%
  left_join( qnkey %>% select(-AnonKey) %>% rename(Key=Response)) %>%
```

```
  mutate(
    Stage2responseFirst = str_sub(Stage2response,1,1),
    WentWithModal = ifelse(S1ModalResponse==Stage2responseFirst,1,0),
    modalCorrect = ifelse(S1ModalResponse==Key,1,0),
    usedModalSecond = ifelse(S1ModalResponse==str_sub(Stage2response,2,2),1,0)
  )

groupPerf %>%
  group_by(WentWithModal)%>%
  summarise(
    count = n(),
    correctness = mean(Stage2score)
  ) %>% knitr::kable()
```

WentWithModal	count	correctness
0	44	0.4318182
1	110	0.8363636
NA	96	0.7812500

On 110 out of 250 occasions, the group went with the modal response, and about 84% of the time they were correct to do so 96 times the group did not have a clear modal option.

```
groupPerf %>%
  group_by(WentWithModal,modalCorrect) %>%
  summarise(
    count = n(),
    correctness = mean(Stage2score)
  ) %>% knitr::kable()
```

WentWithModal	modalCorrect	count	correctness
0	0	35	0.5428571
0	1	9	0.0000000
1	0	18	0.0000000
1	1	92	1.0000000
NA	NA	96	0.7812500

When the group did not use the modal response, it was correct 9/44 times.

Did they go on to try this as their next answer?

```
groupPerf %>%
  group_by(WentWithModal,modalCorrect,usedModalSecond) %>%
  summarise(
    count = n(),
    correctness = mean(Stage2score),
    partial = mean(Stage2scorePartial)
  ) %>% knitr::kable()
```

WentWithModal	modalCorrect	usedModalSecond	count	correctness	partial
0	0	0	27	0.7037037	0.7407407
0	0	1	8	0.0000000	0.0000000
0	1	0	2	0.0000000	0.0000000

WentWithModal	modalCorrect	usedModalSecond	count	correctness	partial
0	1	1	7	0.0000000	0.5000000
1	0	0	18	0.0000000	0.3333333
1	1	0	92	1.0000000	1.0000000
NA	NA	NA	96	0.7812500	0.8385417

Yes, in 7/9 cases they did

These next tables look at a different way of slicing the data:

```
groupPerf %>%
  group_by(modalCorrect,WentWithModal) %>%
  summarise(
    count = n(),
    correctness = mean(Stage2score),
    partial = mean(Stage2scorePartial)
  ) %>% knitr::kable()
```

modalCorrect	WentWithModal	count	correctness	partial
0	0	35	0.5428571	0.5714286
0	1	18	0.0000000	0.3333333
1	0	9	0.0000000	0.3888889
1	1	92	1.0000000	1.0000000
NA	NA	96	0.7812500	0.8385417

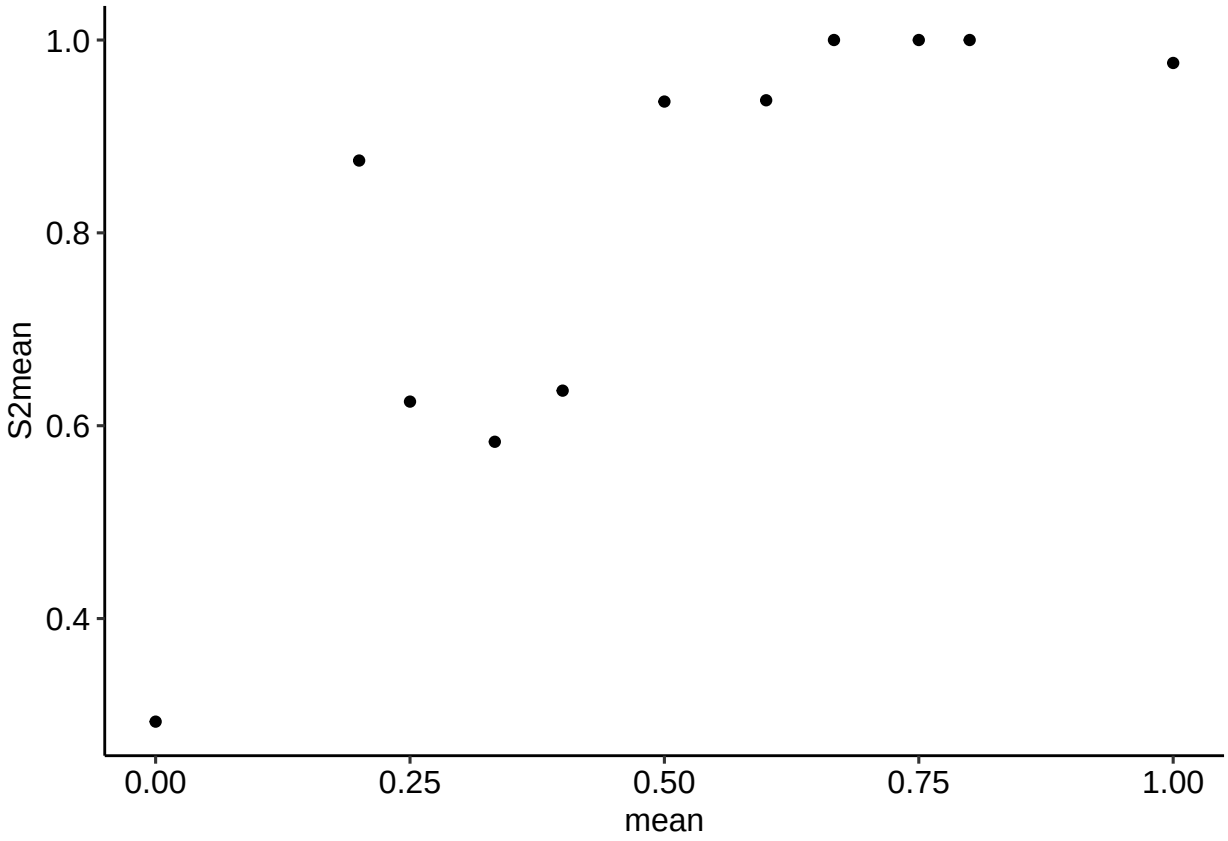
```
groupPerf %>%
  group_by(WentWithModal) %>%
  summarise(
    count = n(),
    correctness = mean(Stage2score),
    partial = mean(Stage2scorePartial)
  ) %>% knitr::kable()
```

WentWithModal	count	correctness	partial
0	44	0.4318182	0.5340909
1	110	0.8363636	0.8909091
NA	96	0.7812500	0.8385417

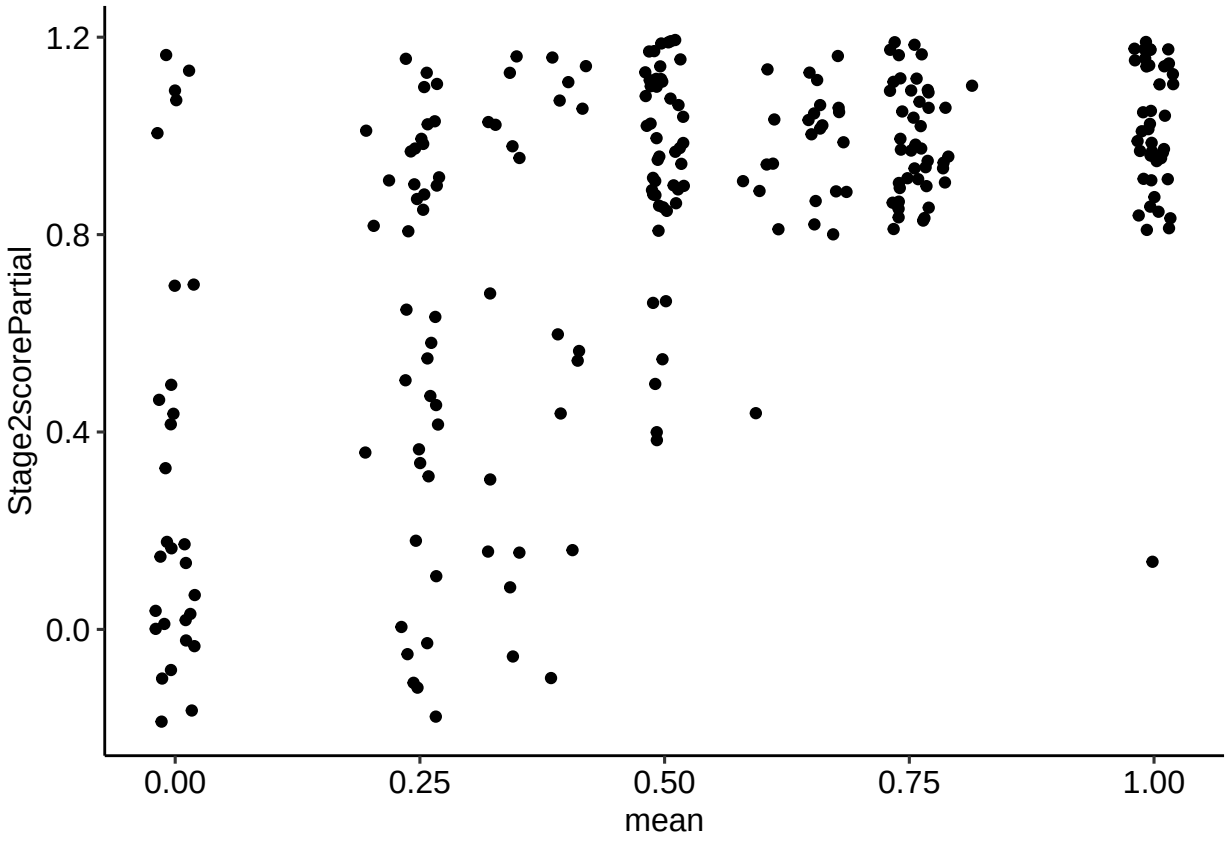
Stage 1 vs Stage 2

Exploring the association between individuals at stage 1 and the group correctness at stage 2, plotting this in different ways.

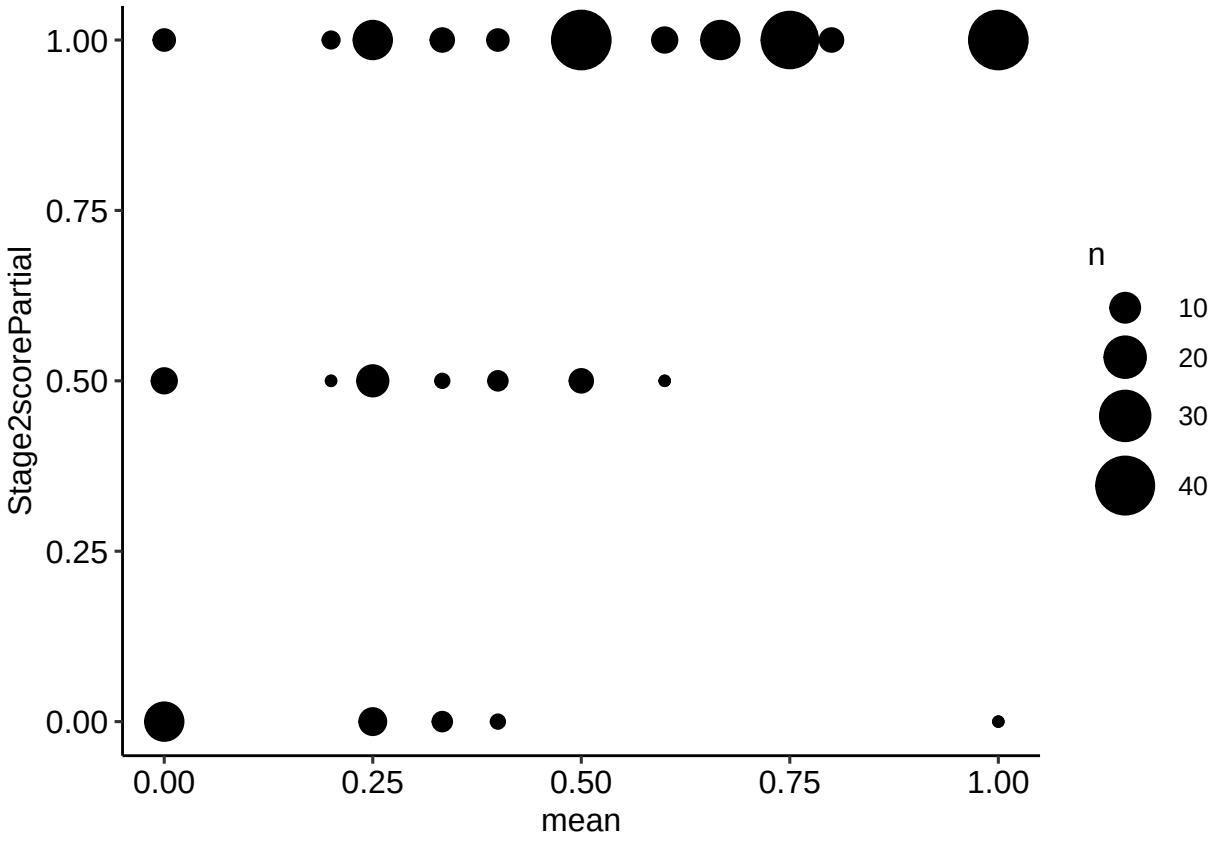
```
S2meanByS1 = groupPerf %>%
  group_by(mean) %>%
  summarise(
    S2mean = mean(Stage2scorePartial)
  )

ggplot(S2meanByS1, aes(x=mean, y=S2mean)) + geom_point()
```



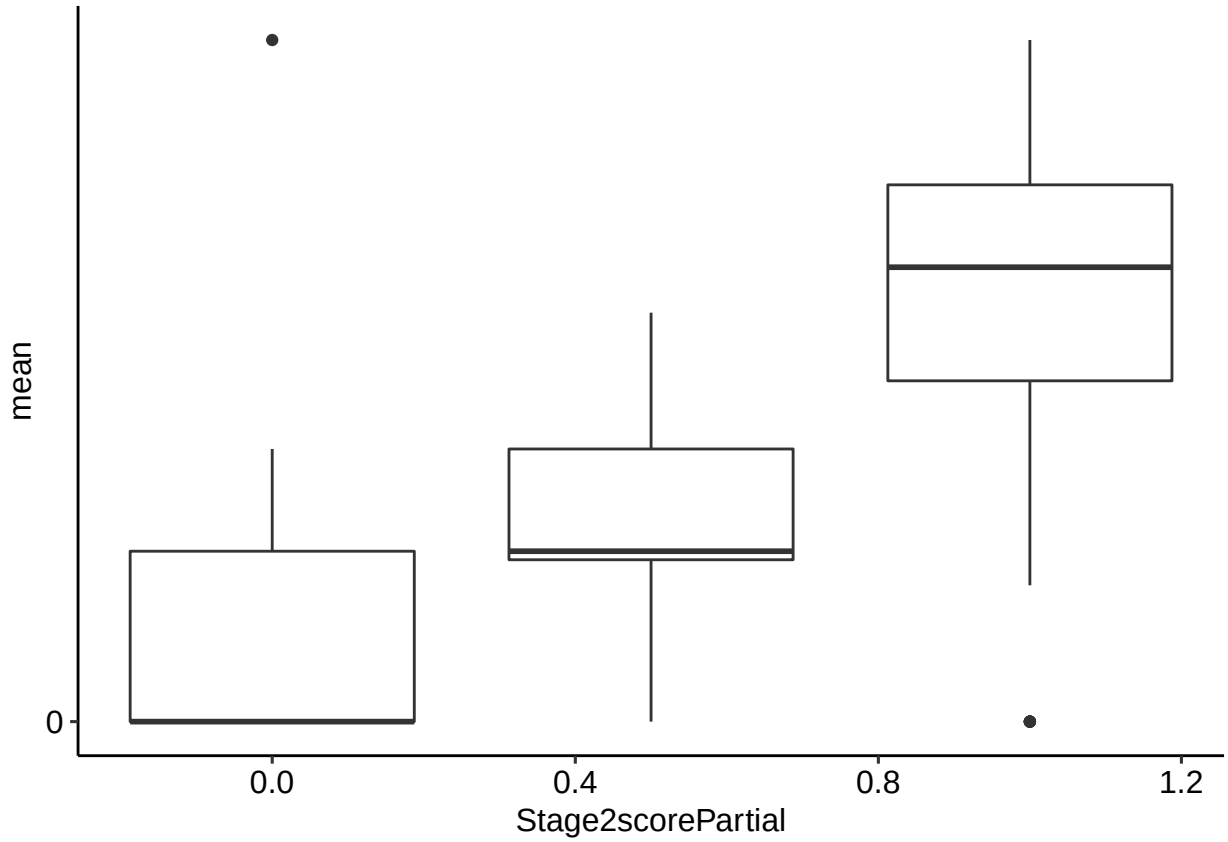

```
ggplot(groupPerf, aes(x=mean, y=Stage2scorePartial)) + geom_point(position="jitter")
```




```
ggplot(groupPerf, aes(x=mean, y=Stage2scorePartial)) + geom_point() +
  geom_count() + scale_size_area(max_size = 10)
```



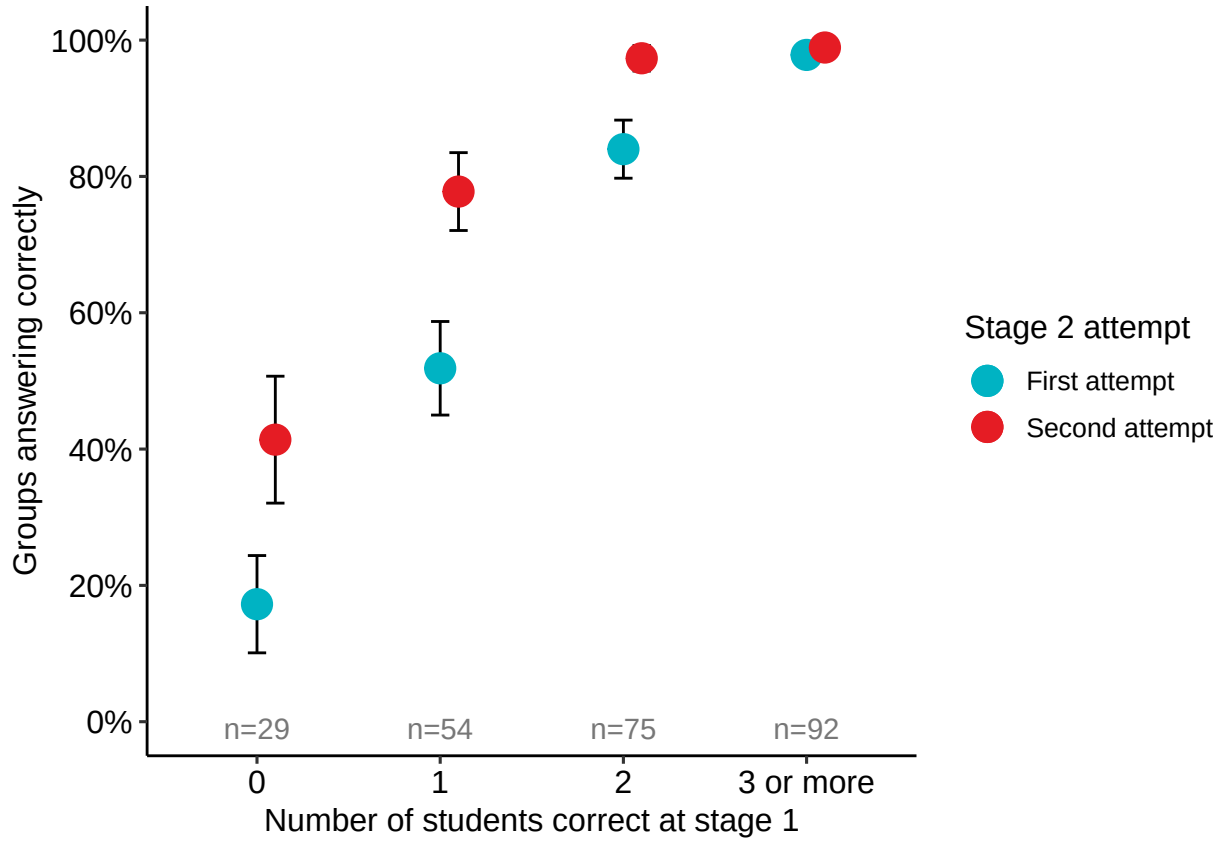

```
ggplot(groupPerf, aes(x = Stage2scorePartial, y = mean, group=Stage2scorePartial)) +
  geom_boxplot()+
  scale_y_continuous(breaks=seq(0,100,10))
```




```
groupPerfS12 = groupPerf %>%
  mutate(
    tot_group = cut(tot,breaks=c(-Inf,0.5,1.5,2.5,Inf),labels=c("0","1","2","3 or more"))
  ) %>%
  group_by(tot_group) %>%
  mutate(
    Stage2scorePartial = ceiling(Stage2scorePartial) # round this up so that groups scoring 0.5 count a
  ) %>%
  summarise(
    S2avg = mean(Stage2score),
    S2se = sd(Stage2score)/sqrt(n()),
    S2n = n(),
    S2Pavg = mean(Stage2scorePartial),
    S2Pse = sd(Stage2scorePartial)/sqrt(n())
  )
```

```
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/Study2_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.

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

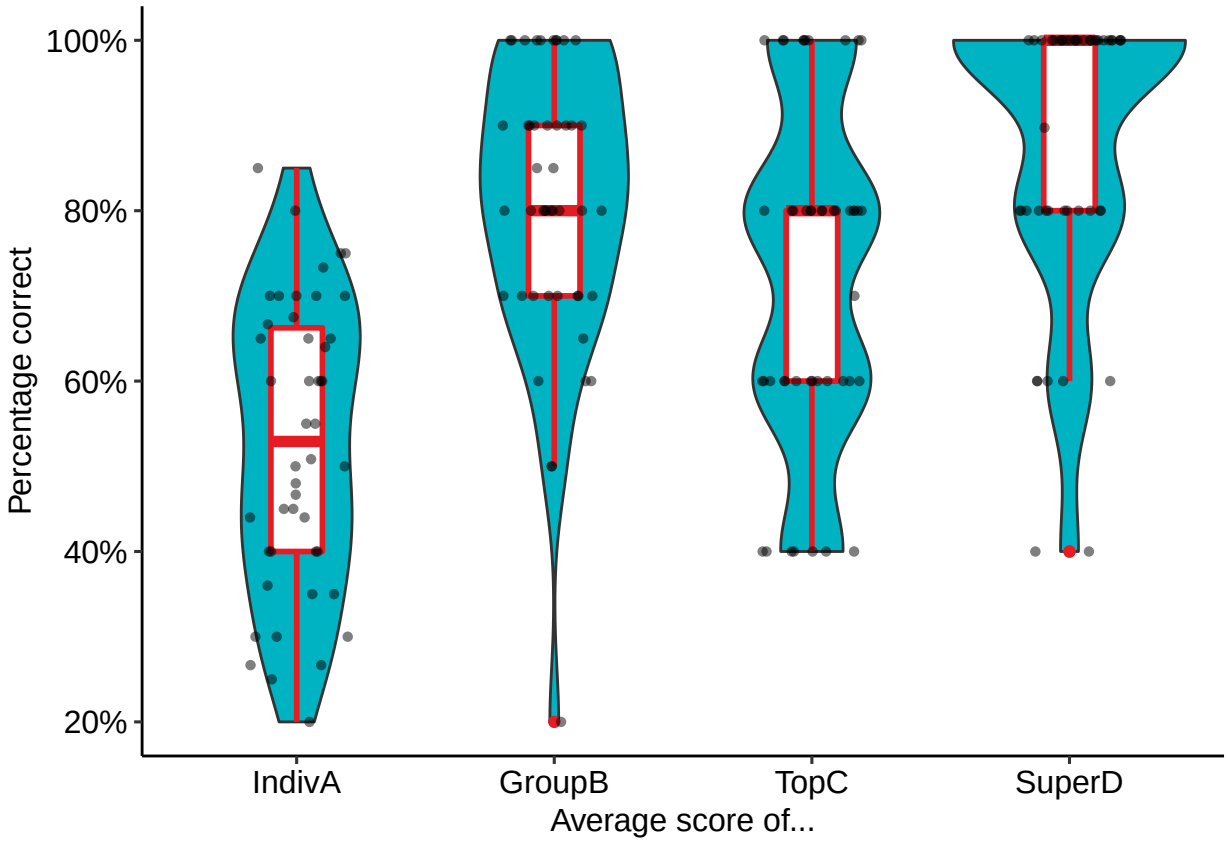
```
S1superandtop = S12data_scored %>%
  mutate(
    Week = as.numeric(Week)
```

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

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

LevyByGroup = groupPerf %>%
  mutate(
    Week = as.numeric(Week)
  ) %>%
  left_join(S1superandtop) %>%
  left_join(LevyA %>% select(Week,S1sd)) %>%
  arrange(Week) %>%
  group_by(Week,Group) %>%
  summarise(
    n = max(n),
    sd = max(S1sd),
    IndivA = mean(mean),
    GroupB = mean(Stage2scorePartial),
    TopC = max(topstudent),
    SuperD = max(superstudent),
    GainBA = (GroupB-IndivA)/sd,
    TopSurplus = (TopC-IndivA)/sd,
    SuperSurplus = (SuperD-IndivA)/sd,
    CollabEfficiency = GainBA / SuperSurplus
  ) %>%
  ungroup()
LevyByGroup %>% knitr::kable(digits = 2)
```

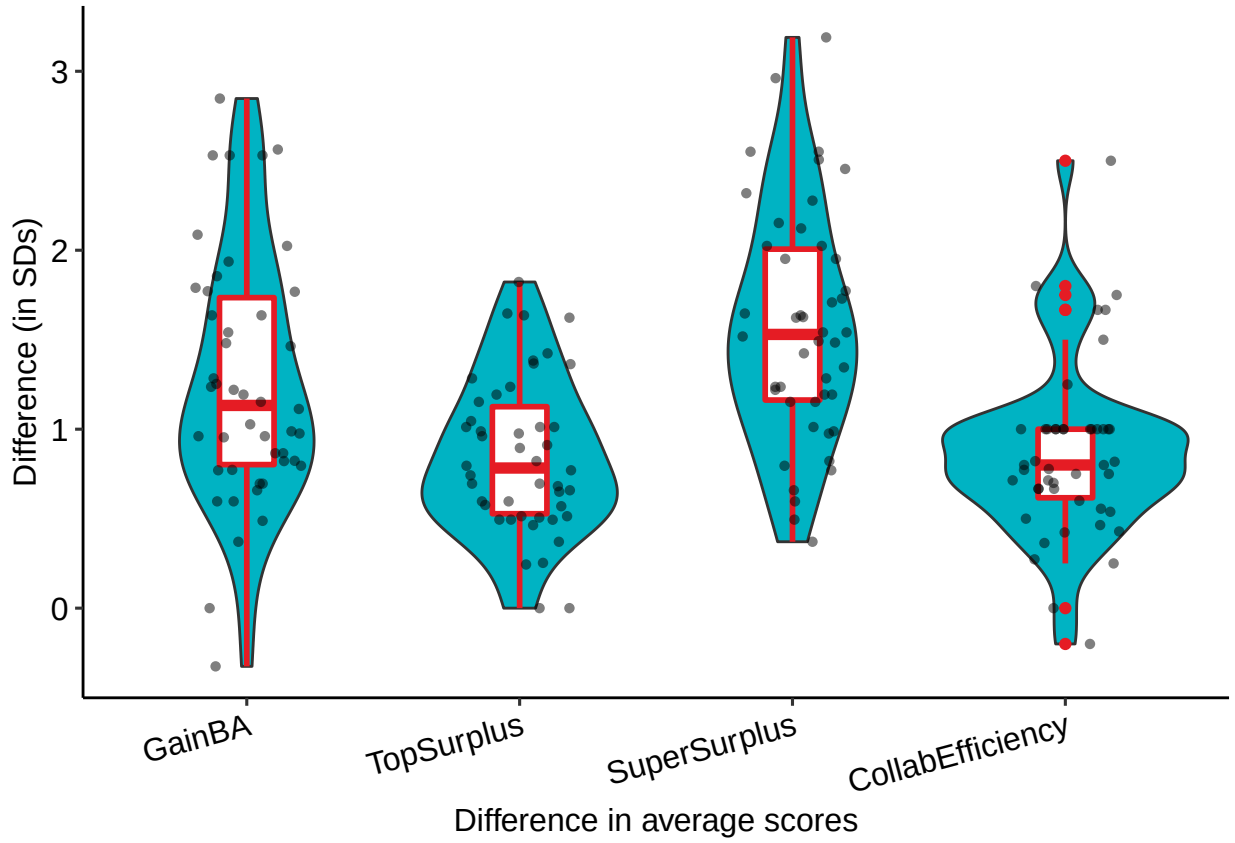
Week	Group	n	sd	IndivA	GroupB	TopC	SuperD	GainBA	TopSurplus	SuperSurplus	CollabEfficienc
1	W1T1	5	0.19	0.64	0.80	0.8	0.8	0.82	0.82	0.82	1.0
1	W1T2	4	0.19	0.70	1.00	0.8	1.0	1.54	0.51	1.54	1.0
1	W1T3	4	0.19	0.75	1.00	1.0	1.0	1.28	1.28	1.28	1.0
1	W1T4	4	0.19	0.70	0.90	0.8	1.0	1.03	0.51	1.54	0.6
1	W1T5	4	0.19	0.85	1.00	1.0	1.0	0.77	0.77	0.77	1.0
2	W2T1	5	0.18	0.36	0.70	0.6	0.8	1.94	1.37	2.51	0.7
2	W2T2	4	0.18	0.35	0.80	0.6	0.6	2.56	1.42	1.42	1.8
2	W2T3	5	0.18	0.44	0.70	0.6	1.0	1.48	0.91	3.19	0.4
2	W2T4	5	0.18	0.48	0.70	0.8	1.0	1.25	1.82	2.96	0.4
2	W2T5	4	0.18	0.30	0.80	0.4	0.6	2.85	0.57	1.71	1.6
3	W3T1	4	0.20	0.40	0.90	0.6	0.8	2.53	1.01	2.02	1.2
3	W3T2	4	0.20	0.40	0.80	0.6	0.8	2.02	1.01	2.02	1.0
3	W3T3	4	0.20	0.55	0.90	0.6	1.0	1.77	0.25	2.28	0.7
3	W3T4	4	0.20	0.20	0.70	0.4	0.4	2.53	1.01	1.01	2.5
3	W3T5	4	0.20	0.30	0.80	0.4	0.6	2.53	0.51	1.52	1.6
5	W5T1	4	0.22	0.45	0.90	0.8	1.0	2.09	1.62	2.55	0.8
5	W5T2	4	0.22	0.65	0.80	0.8	1.0	0.70	0.70	1.62	0.4
5	W5T3	4	0.22	0.30	0.70	0.4	0.8	1.85	0.46	2.32	0.8
5	W5T4	4	0.22	0.45	0.60	0.6	1.0	0.70	0.70	2.55	0.2
6	W6T1	3	0.34	0.40	1.00	0.6	0.8	1.79	0.60	1.19	1.5
6	W6T2	4	0.34	0.50	0.90	0.8	1.0	1.19	0.90	1.49	0.8
6	W6T3	3	0.34	0.73	1.00	1.0	1.0	0.80	0.80	0.80	1.0
6	W6T4	3	0.34	0.60	0.80	1.0	1.0	0.60	1.19	1.19	0.5
6	W6T5	4	0.34	0.80	1.00	1.0	1.0	0.60	0.60	0.60	1.0
7	W7T1	3	0.20	0.27	0.20	0.4	0.6	-0.33	0.65	1.63	-0.2
7	W7T2	3	0.20	0.60	0.70	0.6	1.0	0.49	0.00	1.95	0.2
7	W7T3	2	0.20	0.60	0.80	0.6	0.8	0.98	0.00	0.98	1.0
7	W7T4	4	0.20	0.40	0.70	0.6	0.8	1.46	0.98	1.95	0.7
7	W7T5	4	0.20	0.75	1.00	0.8	1.0	1.22	0.24	1.22	1.0
8	W8T1	3	0.27	0.47	0.70	0.6	0.8	0.87	0.49	1.24	0.7
8	W8T2	3	0.27	0.27	0.50	0.4	0.4	0.87	0.49	0.49	1.7
8	W8T3	4	0.27	0.60	0.90	0.8	1.0	1.11	0.74	1.48	0.7
8	W8T4	3	0.27	0.67	1.00	1.0	1.0	1.24	1.24	1.24	1.0
8	W8T5	4	0.27	0.70	0.80	0.8	0.8	0.37	0.37	0.37	1.0
9	W9T1	4	0.30	0.60	0.60	0.8	0.8	0.00	0.66	0.66	0.0
9	W9T2	4	0.30	0.65	0.90	0.8	1.0	0.82	0.49	1.15	0.7
9	W9T3	4	0.30	0.50	0.80	1.0	1.0	0.99	1.65	1.65	0.6
9	W9T4	4	0.30	0.65	1.00	1.0	1.0	1.15	1.15	1.15	1.0
9	W9T5	4	0.30	0.70	0.90	1.0	1.0	0.66	0.99	0.99	0.6
10	W10T1	5	0.26	0.44	0.90	0.8	1.0	1.77	1.38	2.15	0.8
10	W10T2	4	0.26	0.35	0.60	0.6	0.8	0.96	0.96	1.73	0.5
10	W10T3	4	0.26	0.25	0.50	0.4	0.6	0.96	0.58	1.35	0.7
11	W11T1	4	0.18	0.51	0.65	0.7	0.9	0.77	1.05	2.12	0.3
11	W11T2	4	0.18	0.68	0.85	0.8	1.0	0.95	0.68	1.77	0.5
11	W11T3	4	0.18	0.55	0.85	0.8	1.0	1.64	1.36	2.45	0.6
11	W11T4	5	0.18	0.70	1.00	1.0	1.0	1.64	1.64	1.64	1.0

```
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/Study2_LevyABCD.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study2_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),alpha=0.5) +
  labs(x = "Difference in average scores",
       y = "Difference (in SDs)")+
  theme(axis.text.x = element_text(angle = 15, hjust = 1))
```




```
ggsave("Figs/Study2_LevyDiffs.pdf",width=20,height=10,units="cm",dpi=300)
ggsave("Figs/Study2_LevyDiffs_small.pdf",width=10,height=7,units="cm",dpi=300)
```

```
LevyByGroup %>%
  summarise(
    CollabEfficiency_m = mean(CollabEfficiency),
    CollabEfficiency_sd = sd(CollabEfficiency),
    n = n()
  )
```

```
## # A tibble: 1 x 3
##   CollabEfficiency_m CollabEfficiency_sd     n
##                <dbl>               <dbl> <int>
## 1              0.864               0.482    46
```

```
LevyByGroup %>%
  filter(CollabEfficiency>1) %>%
  count() %>%
  knitr::kable()
```

n
26

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

```
  knitr::kable()
```

n
50

```
LevyByGroup %>%
  filter(CollabEfficiency<1) %>%
  count() %>%
  knitr::kable()
```

n
104

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.

```
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)
```

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.

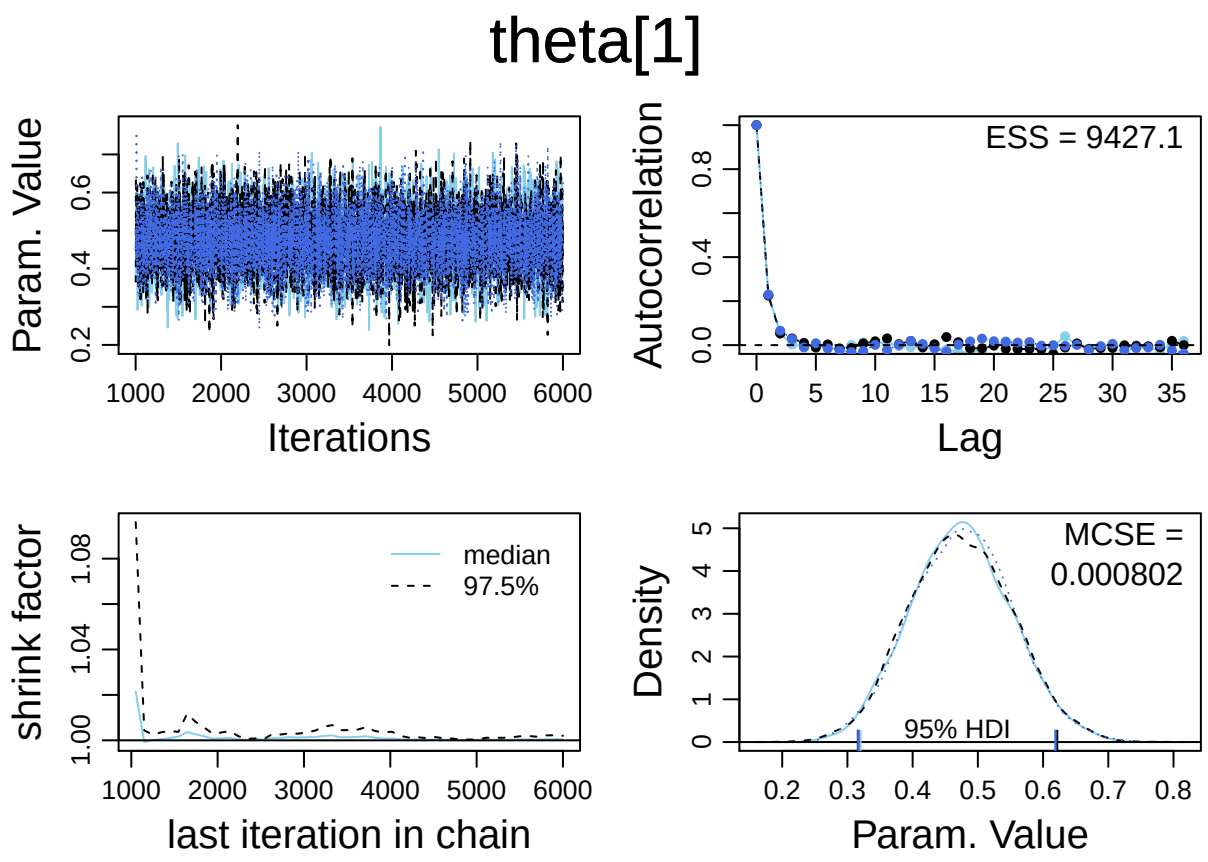
```
samp_mat <- as.matrix(samplesS3)
print(quantile(samp_mat[, "theta[2]"] - samp_mat[, "theta[1]"], c(0.025, 0.5, 0.975)))
```

```
##        2.5%         50%       97.5%
## -0.07012088  0.14540568  0.34791809
```

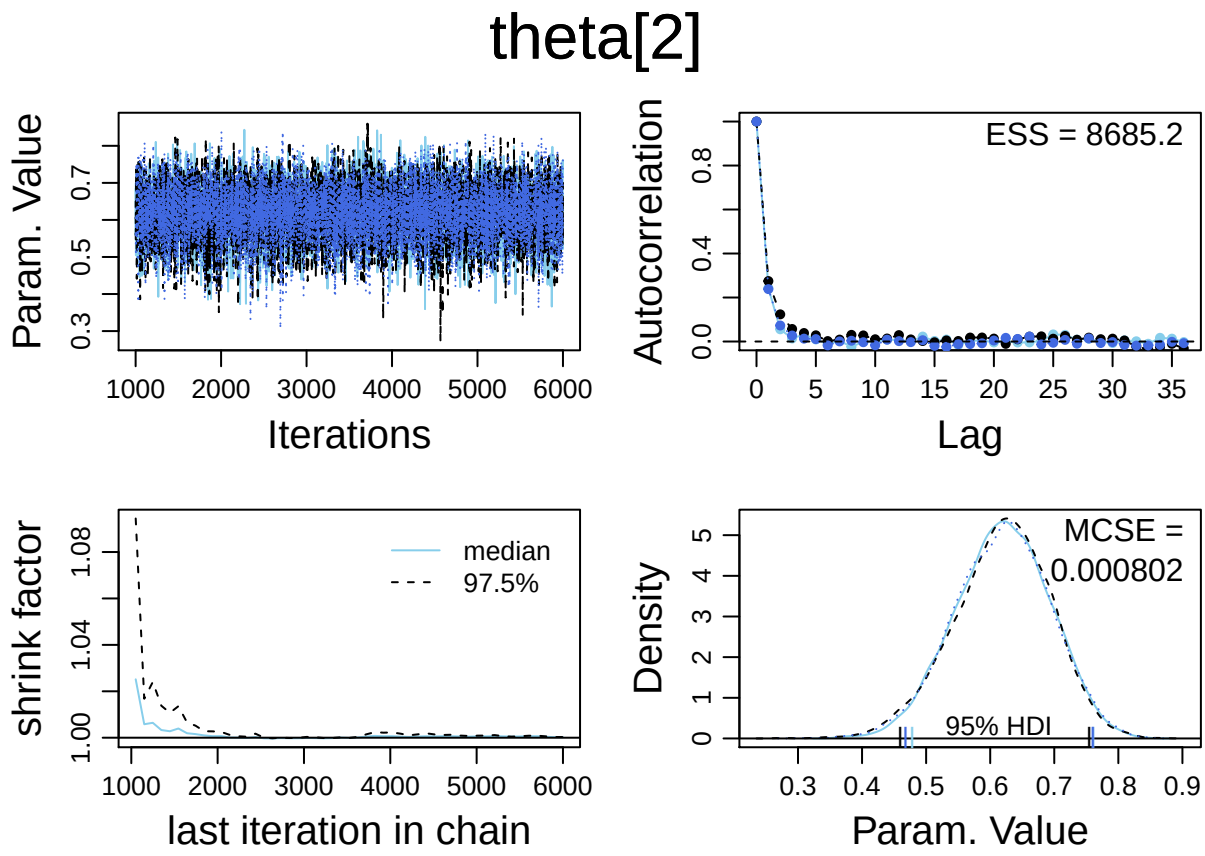
```
print(quantile(samp_mat[, "theta[4]"] - samp_mat[, "theta[3]"], c(0.025, 0.5, 0.975)))
```

```
##       2.5%        50%      97.5%
## -0.1097606  0.1729397  0.5020112
```

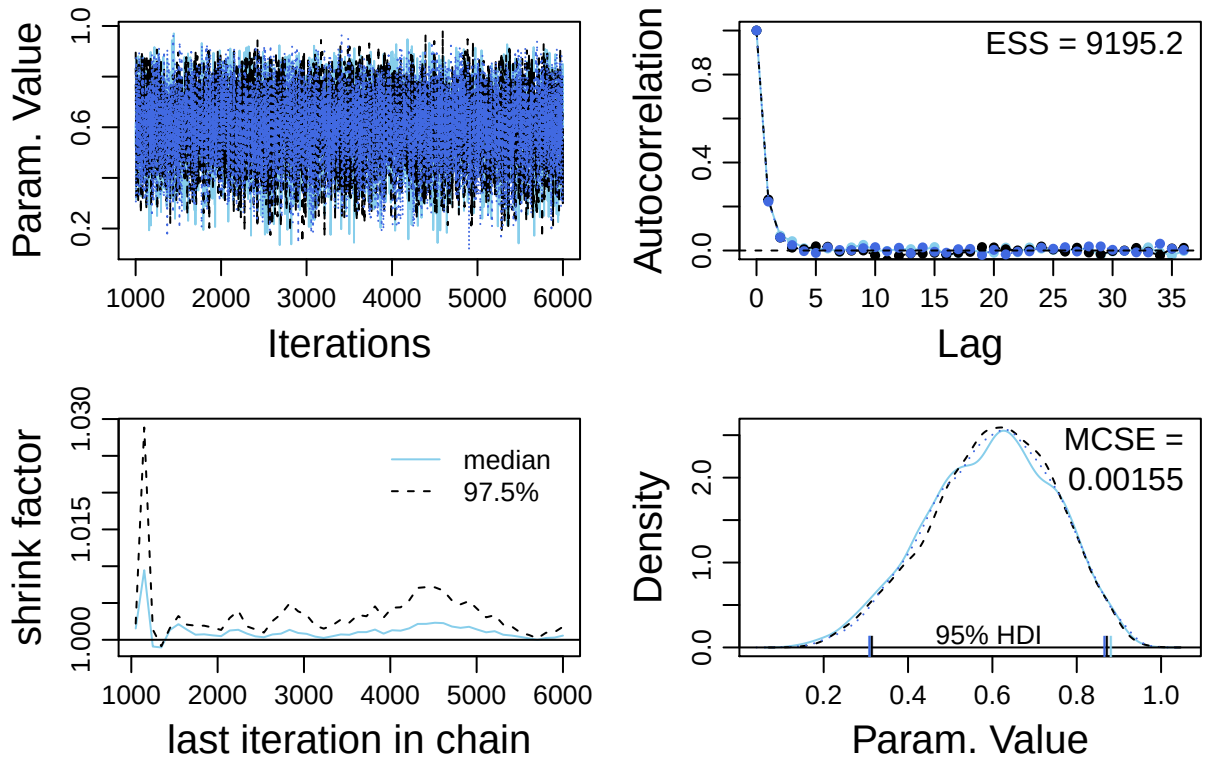
```
diagMCMC(samplesS3, parName = "theta[1]", saveName="Figs/Study2_S3props", saveType = "pdf")
```



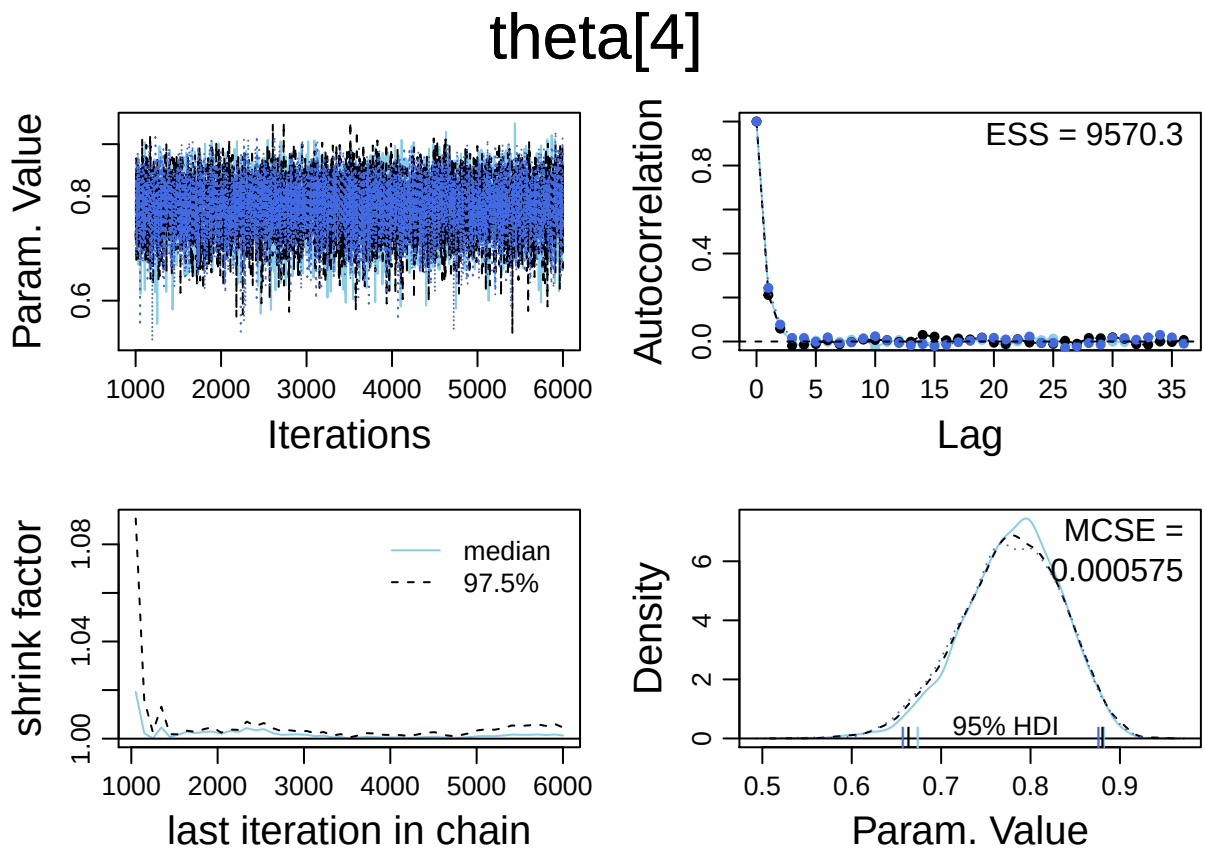

```
diagMCMC(samplesS3, parName = "theta[2]", saveName="Figs/Study2_S3props", saveType = "pdf")
```



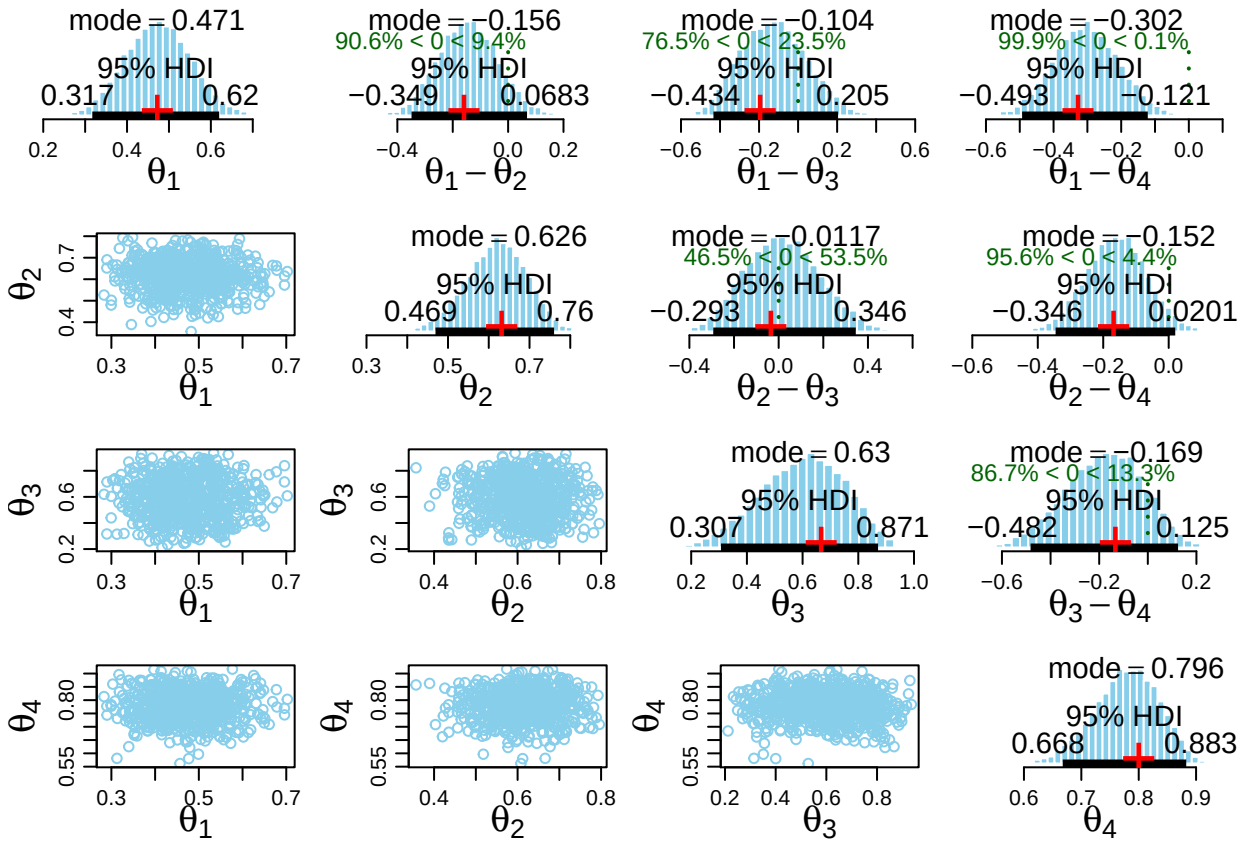

```
diagMCMC(samplesS3, parName = "theta[3]", saveName="Figs/Study2_S3props", saveType = "pdf")
```



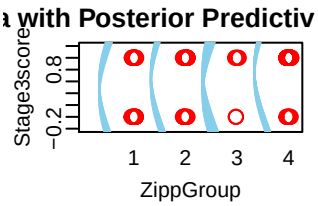

```
diagMCMC(samplesS3, parName = "theta[4]", saveName="Figs/Study2_S3props", saveType = "pdf")
```




```
plotMCMC( samplesS3, data=myData, yName="Stage3score", sName="ZippGroup", compVal=NULL, compValDiff=0.0
          saveName="Figs/Study2_S3props", saveType = "pdf")
```



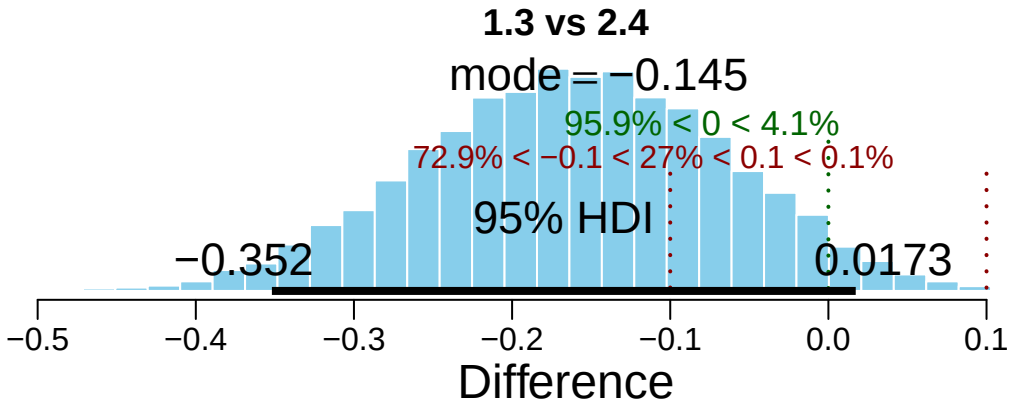

```
contrasts = list(
  list( c(1,3), c(2,4), compVal=0.0, ROPE=c(-0.1,0.1)),
  list( c(1,2), c(3,4), compVal=0.0, ROPE=c(-0.1,0.1))
)
plotMCMCwithContrasts( samplesS3, datFrm=data.frame(myData), yName="Stage3score", xName="ZippGroup", co
          saveName="Figs/Study2_S3props", saveType = "pdf")
```

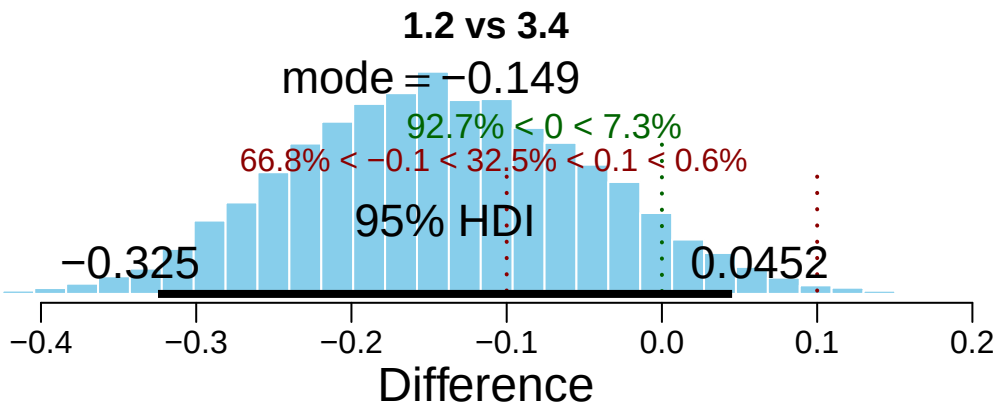



```
## [1] 1
## [1] 1
## [1] 790
## [1] 1580
## [1] 2369
## [1] 3159
## [1] 3948
## [1] 4738
## [1] 5527
## [1] 6316
## [1] 7106
## [1] 7895
## [1] 8685
## [1] 9474
## [1] 10263
## [1] 11053
## [1] 11842
## [1] 12632
## [1] 13421
## [1] 14211
## [1] 15000
## [1] 2
## [1] 1
## [1] 790
## [1] 1580
## [1]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




1.2 vs 3.4
mode = −0.149
92.7% < 0 < 7.3%
66.8% < −0.1 < 32.5% < 0.1 < 0.6%
95% HDI
−0.325
0.0452
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
Difference