

Quadrilateral learning experiments: Baseline

January 22, 2016

```
library(langcog)
```

```
##  
## Attaching package: 'langcog'  
  
## The following object is masked from 'package:base':  
##  
##     scale
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)  
library(tidyr)  
library(binom)  
library(lme4)
```

```
## Loading required package: Matrix  
  
##  
## Attaching package: 'Matrix'  
  
## The following object is masked from 'package:tidyr':  
##  
##     expand
```

```
library(bootstrap)  
library(magrittr)
```

```
##  
## Attaching package: 'magrittr'  
  
## The following object is masked from 'package:tidyr':  
##  
##     extract
```

```
library(stringr)
theme_set(theme_classic())
```

Load data.

```
d1 <- read.csv("../data/baseline_expt/quadmods-baseline.csv", header=TRUE,
               row.names=NULL, stringsAsFactors = FALSE)

d1 <- d1 %>%
  distinct() %>%
  mutate(block_num = as.numeric(block))
```

```
## Warning in eval(substitute(expr), envir, enclos): NAs introduced by
## coercion
```

Get the block each ss reached criterion

```
d1 %<>%
  filter(trial_type == "training") %>%
  group_by(subids) %>%
  summarise(block_reached_crit = max(block_num)) %>%
  left_join(., d1) %>%
  mutate(
    reached_crit = ifelse(block_reached_crit < 15, "yes", "no"),
    blocks_to_go = block_reached_crit - block_num + 1
  )
```

```
## Joining by: "subids"
```

Descriptives

How many participants?

```
d1 %>%
  select(subids, reached_crit) %>%
  unique() %>%
  group_by(reached_crit) %>%
  summarise(n_subs = n()) %>%
  knitr::kable()
```

reached_crit	n_subs
no	26
yes	13

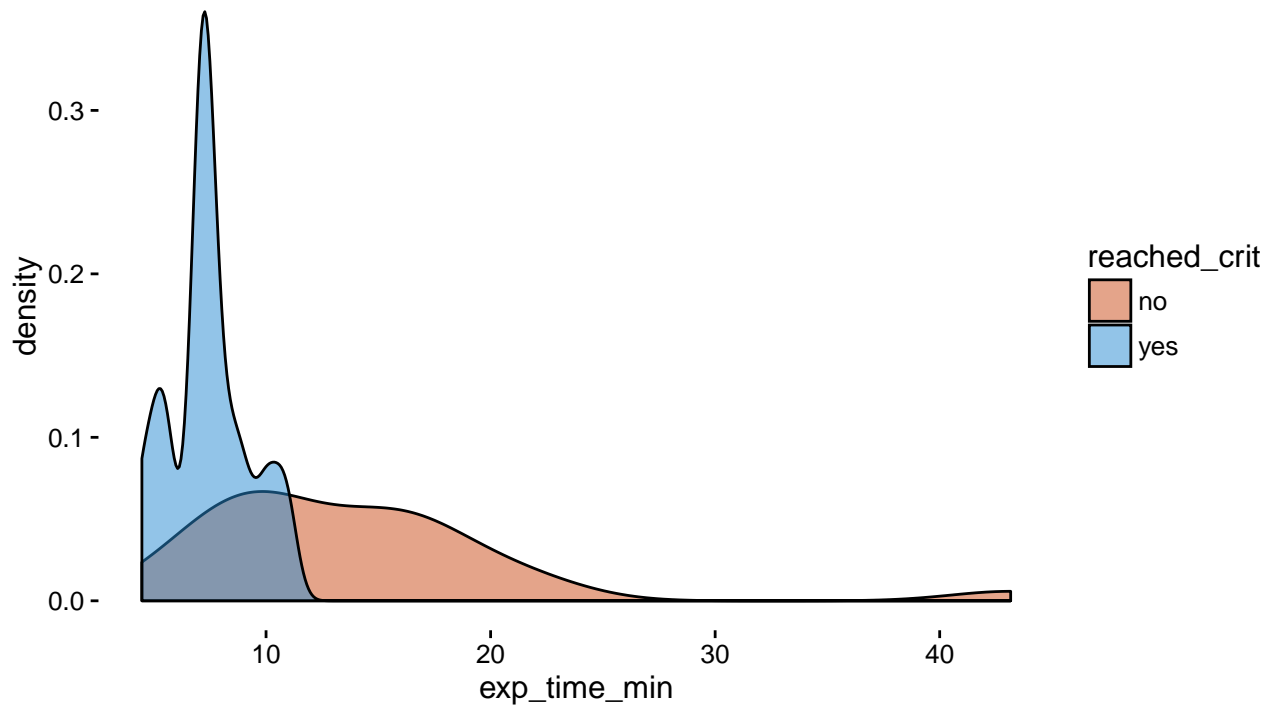
How long did experiment take?

```
d1 %>%
  filter(trial_type == "training") %>%
  group_by(reached_crit) %>%
  summarise(m_train_time_sec = (mean(trial_time) / 1000),
            m_exp_time_min = (mean(exp_time) / 1000) / 60) %>%
  knitr::kable()
```

reached_crit	m_train_time_sec	m_exp_time_min
no	1.548119	13.993036
yes	2.024719	8.234531

```
ss <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(subids, reached_crit) %>%
  summarise(train_time_sec = (mean(trial_time) / 1000),
            exp_time_min = (mean(exp_time) / 1000) / 60,
            max_training_block = max(as.numeric(block)))

ggplot(aes(x=exp_time_min, fill = reached_crit), data = ss) +
  geom_density(alpha = 0.5) +
  scale_fill_solarized()
```

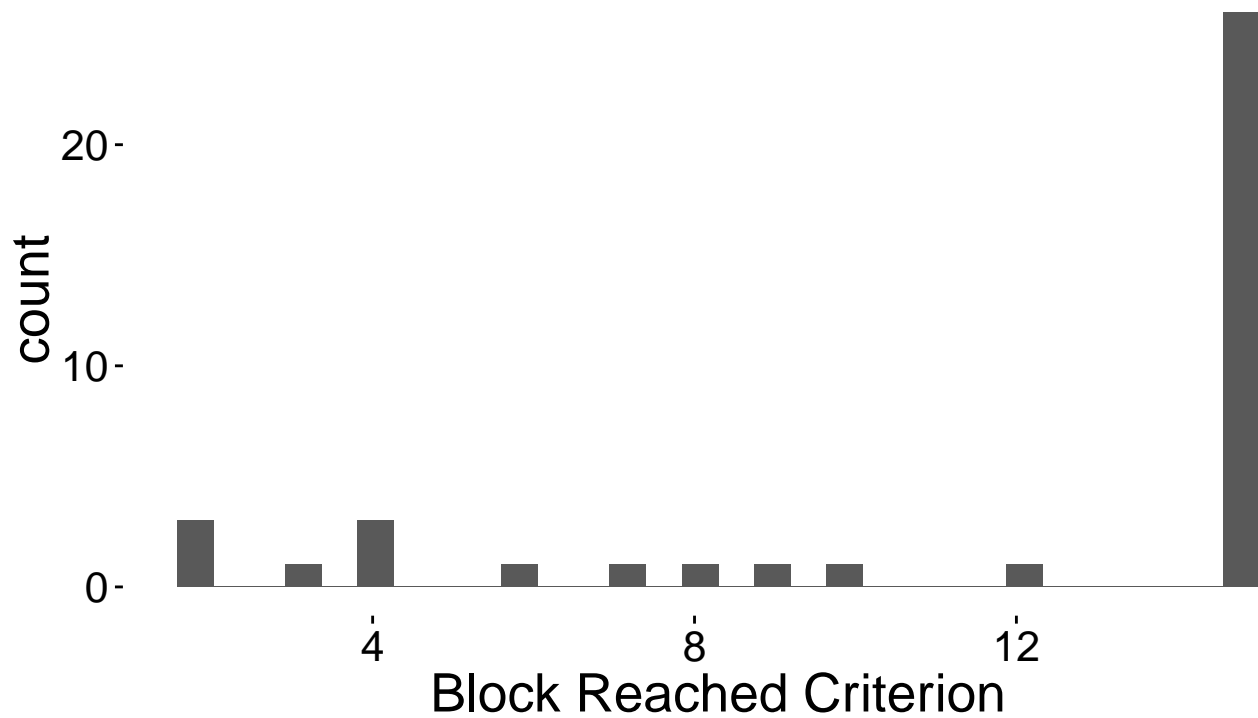


How many blocks did it take to reach criterion?

```
ss <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(subids) %>%
  summarise(block_reached_crit = max(as.numeric(block)))

ggplot(data = ss, aes(x = block_reached_crit)) +
  geom_histogram() +
  xlab("Block Reached Criterion") +
  theme(text = element_text(size = 20))
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Comments

```
d1 %>%
  distinct(subids) %>%
  select(subids:exp_improve) %>%
  knitr::kable()
```

subids	block_reached_crit	age	gender	comments
1	15	43	m	none
2	2	33	f	no
3	15	26	female	no
4	15	52	f	no

subids	block_reached_crit	age	gender	comments
5	15	24	male	great hit
6	2	48	m	no
7	15	34	F	NO THANK YOU
8	15	40	female	no
9	15	42	m	Thank you for letting me participate.
10	9	32	m	
11	15	36	m	No
12	15	30	f	
13	3	39	f	none, good survey/game
14	10	38	f	
15	15	64	f	
16	15	24	f	Thanks.
17	8	27	f	
18	15	24	male	no
19	15	27	f	Nope.
20	15	33	m	No.
21	2	39	m	no comments
22	15	58	female	that was on the difficult side
23	15	31	m	no
24	4	33	f	no
25	15	39	M	Nope, thanks!
26	4	20	m	
27	6	26	m	No, thanks.
28	12	54	f	None
29	15	49	f	no
30	15	27	m	na
31	15	40	m	None
32	7	35	f	
33	4	56	f	Not really, other than it was actually pretty useful and that I learned something
34	15	33	F	Fun & interesting study
35	15	42	f	none
36	15	32	f	na
37	15	25	f	None.
38	15	29	M	
39	15	23	m	N/A

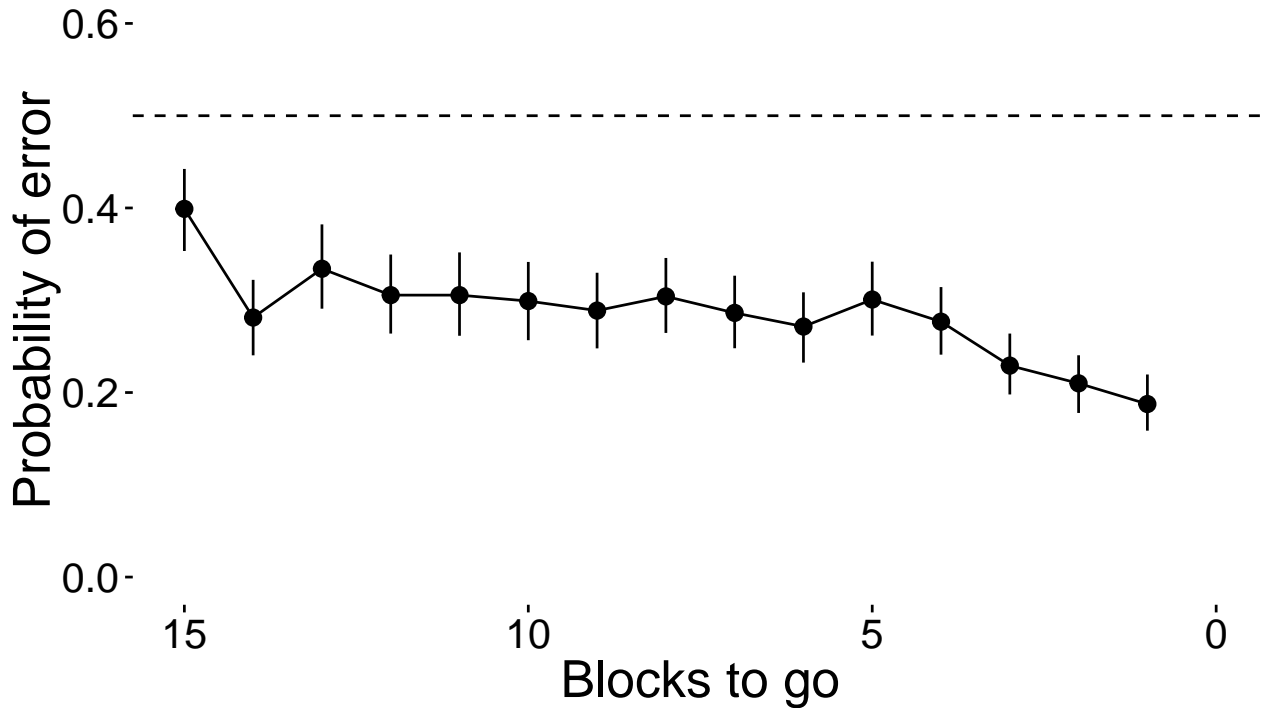
Visualization

Training performance across blocks for all Ss aggregate

```
ms <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(blocks_to_go) %>%
  multi_boot_standard("correct") %>%
  mutate(
    p_error = 1 - mean,
    ci_lower = 1 - ci_lower,
    ci_upper = 1 - ci_upper)
```

```
## Joining by: "blocks_to_go"
```

```
ggplot(data = ms, aes(x=as.numeric(blocks_to_go), y=p_error)) +
  geom_pointrange(aes(ymin=ci_lower, ymax=ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0, .6) +
  xlim(15,0) +
  scale_color_solarized() +
  ylab("Probability of error") +
  xlab("Blocks to go") +
  theme(text = element_text(size = 20))
```



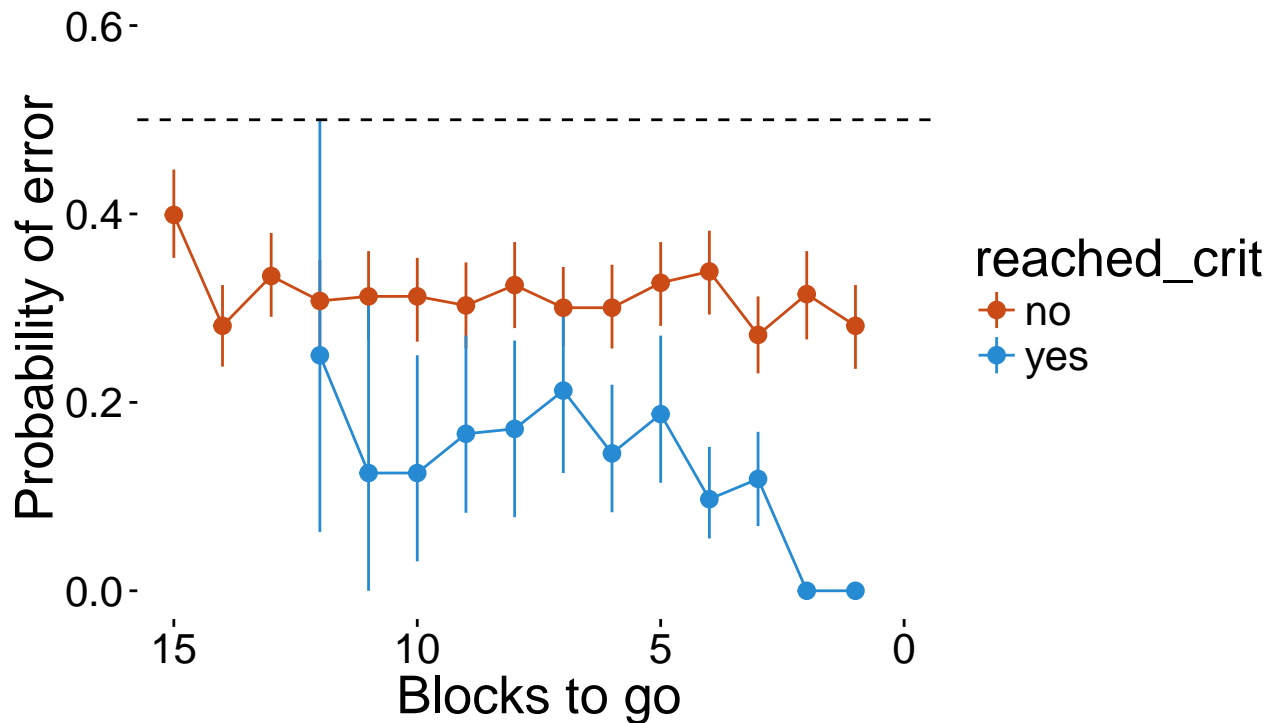
Training performance across blocks for all Ss by reached crit

```
ms <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(blocks_to_go, reached_crit) %>%
  multi_boot_standard("correct") %>%
  mutate(
    p_error = 1 - mean,
    ci_lower = 1 - ci_lower,
    ci_upper = 1 - ci_upper)
```

```
## Joining by: c("blocks_to_go", "reached_crit")
```

```
ggplot(data = ms, aes(x=as.numeric(blocks_to_go), y=p_error, color=reached_crit)) +
  geom_pointrange(aes(ymin=ci_lower, ymax=ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
```

```
ylim(0, .6) +
xlim(15,0) +
scale_color_solarized() +
ylab("Probability of error") +
xlab("Blocks to go") +
theme(text = element_text(size = 20))
```



Training performance across blocks for all Ss by shape

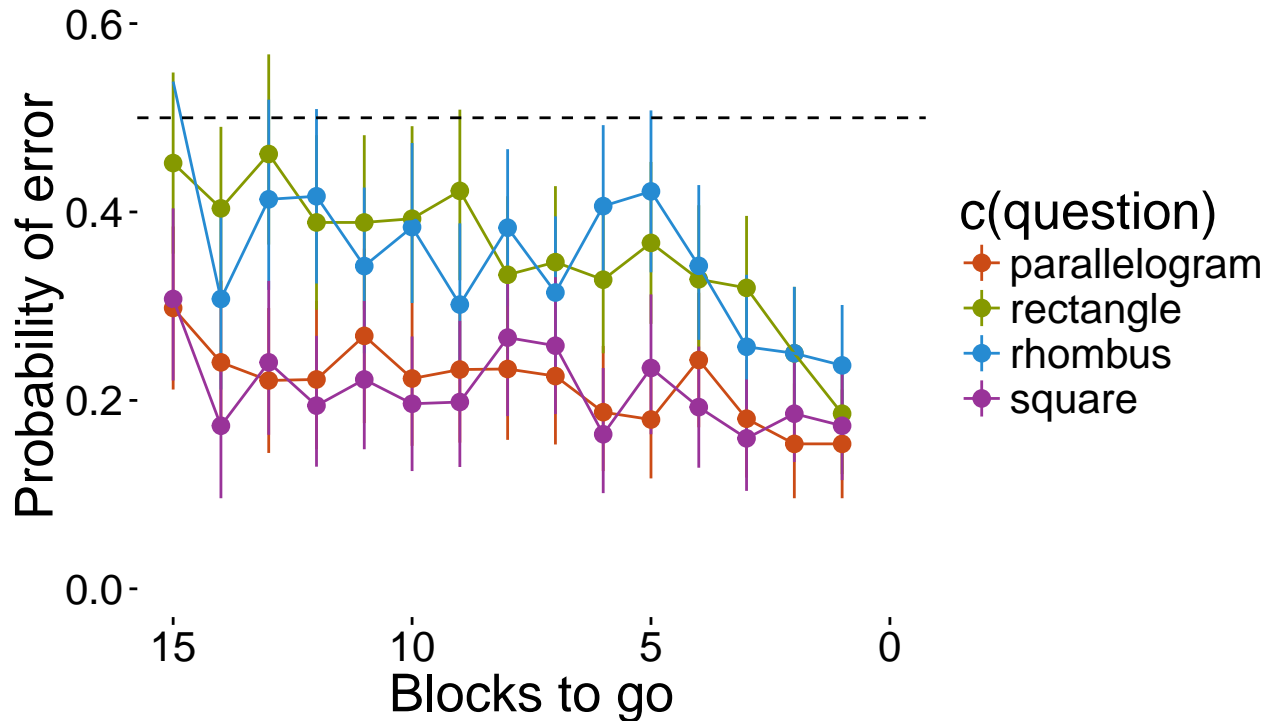
```
ms <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(blocks_to_go, question) %>%
  multi_boot_standard("correct") %>%
  mutate(
    p_error = 1 - mean,
    ci_lower = 1 - ci_lower,
    ci_upper = 1 - ci_upper)
```

```
## Joining by: c("blocks_to_go", "question")
```

```
ggplot(data = ms, aes(x=blocks_to_go, y=p_error, color=c(question))) +
  geom_pointrange(aes(ymin=ci_lower, ymax=ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0, .6) +
  xlim(15,0) +
  scale_color_solarized() +
```

```
ylab("Probability of error") +
xlab("Blocks to go") +
theme(text = element_text(size = 20))
```

```
## Warning: Removed 1 rows containing missing values (geom_pointrange).
```



Training performance across blocks for all Ss by shape for those who reached criterion

```
ms <- d1 %>%
  filter(trial_type == "training", reached_crit=="yes") %>%
  group_by(blocks_to_go, question) %>%
  multi_boot_standard("correct") %>%
  mutate(
    p_error = 1 - mean,
    ci_lower = 1 - ci_lower,
    ci_upper = 1 - ci_upper)
```

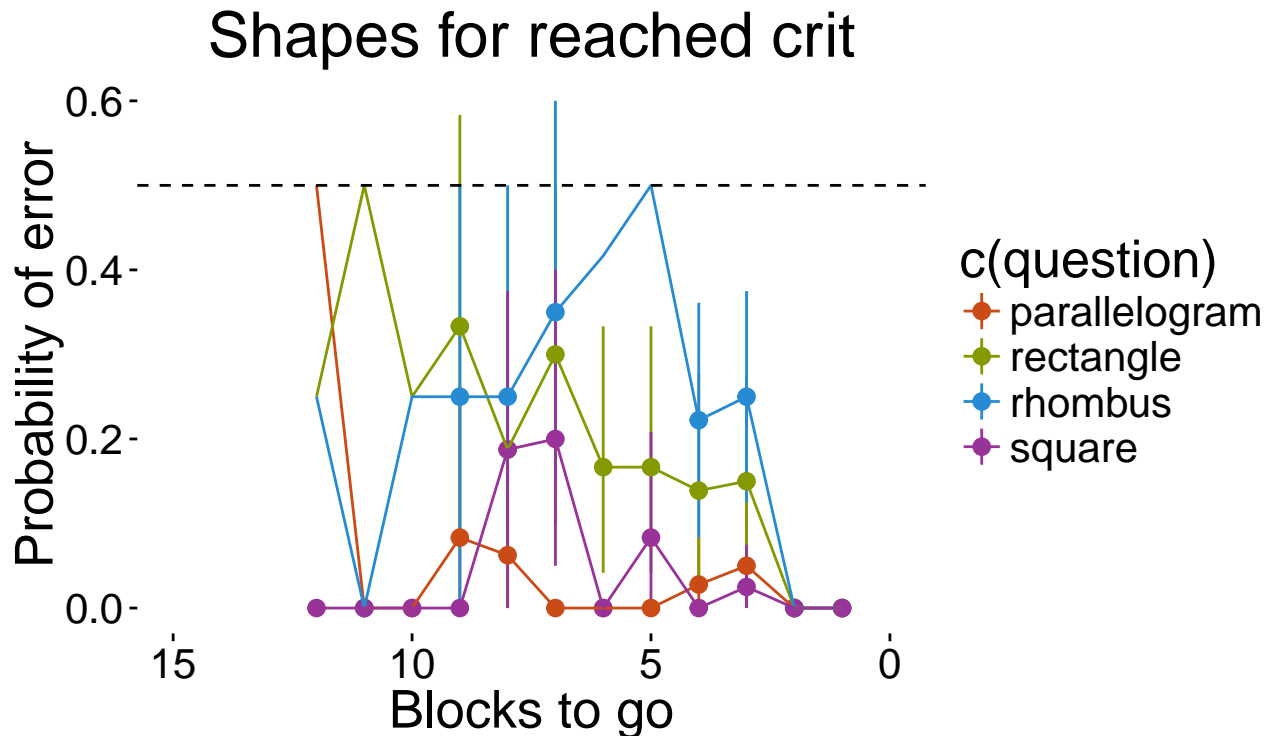
```
## Joining by: c("blocks_to_go", "question")
```

```
ggplot(data = ms, aes(x=blocks_to_go, y=p_error, color=c(question))) +
  geom_pointrange(aes(ymin=ci_lower, ymax=ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0, .6) +
  xlim(15,0) +
  scale_color_solarized() +
```



```
ylab("Probability of error") +
xlab("Blocks to go") +
ggtitle("Shapes for reached crit") +
theme(text = element_text(size = 20))
```

Warning: Removed 8 rows containing missing values (geom_pointrange).



Training performance across blocks for all Ss by shape for those who did not reach criterion

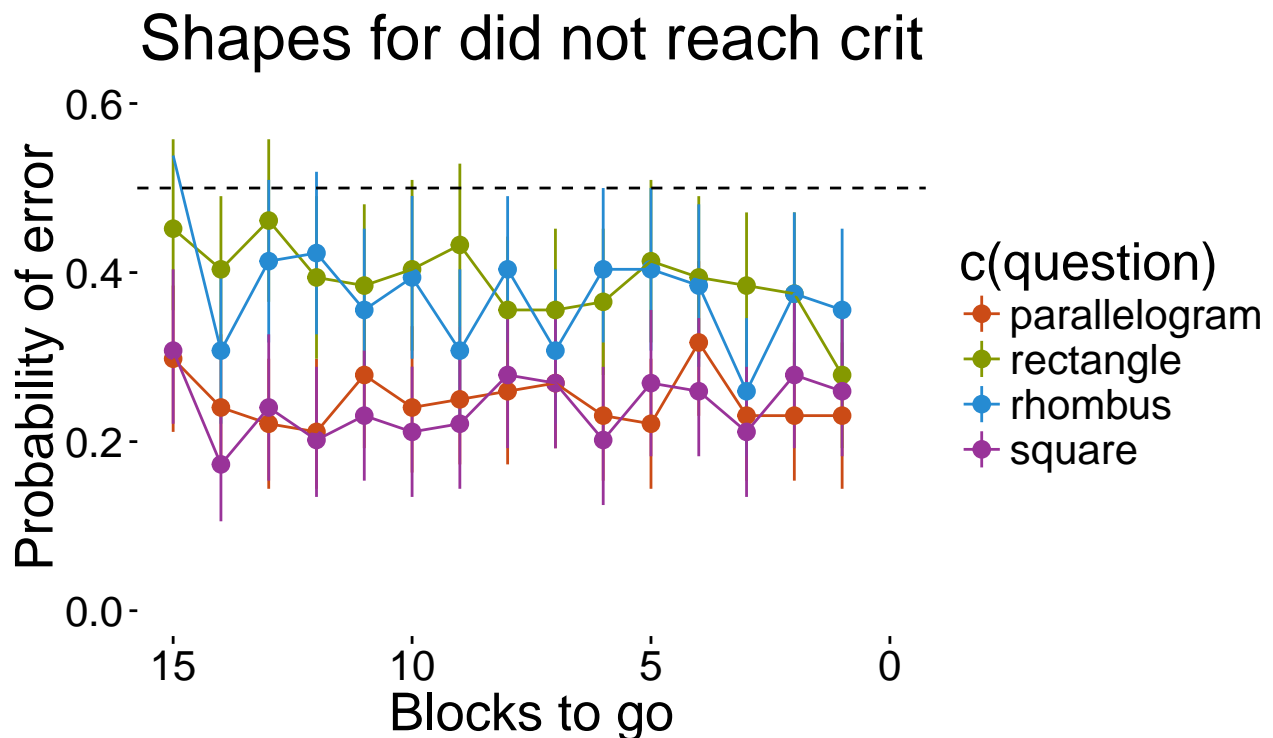
```
ms <- d1 %>%
  filter(trial_type == "training", reached_crit=="no") %>%
  group_by(blocks_to_go, question) %>%
  multi_boot_standard("correct") %>%
  mutate(
    p_error = 1 - mean,
    ci_lower = 1 - ci_lower,
    ci_upper = 1 - ci_upper)
```

Joining by: c("blocks_to_go", "question")

```
ggplot(data = ms, aes(x=blocks_to_go, y=p_error, color=c(question))) +
  geom_pointrange(aes(ymin=ci_lower, ymax=ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0, .6) +
  xlim(15,0) +
```

```
scale_color_solarized() +
ylab("Probability of error") +
xlab("Blocks to go") +
ggtitle("Shapes for did not reach crit") +
theme(text = element_text(size = 20))
```

Warning: Removed 1 rows containing missing values (geom_pointrange).

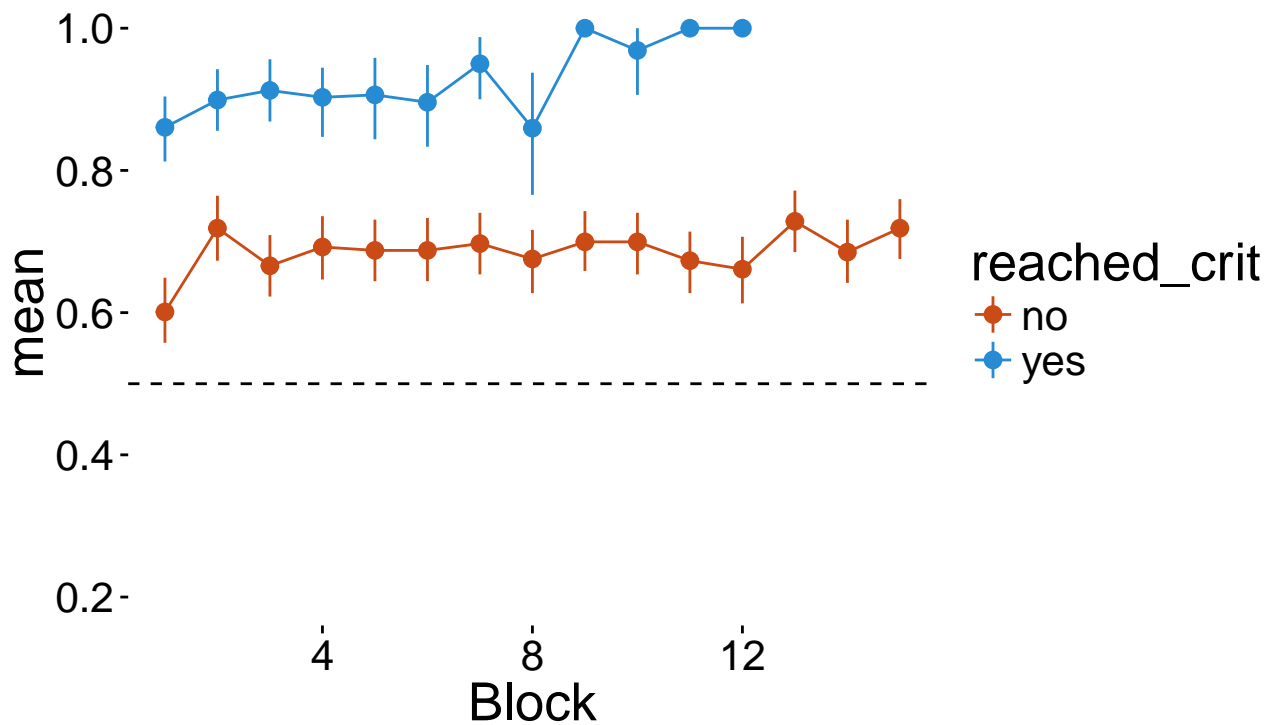


Training performance across blocks split by ss who reached training criterion

```
ms <- d1 %>%
  filter(trial_type == "training") %>%
  group_by(block, reached_crit) %>%
  multi_boot_standard(column = "correct") %>%
  mutate(prob_error = 1 - mean)
```

Joining by: c("block", "reached_crit")

```
ggplot(data = ms, aes(x = as.numeric(block), y = mean, color = reached_crit)) +
  geom_pointrange(aes(ymin = ci_lower, ymax = ci_upper)) +
  geom_line() +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0.2, 1) +
  scale_color_solarized() +
  xlab("Block") +
  theme(text = element_text(size = 20))
```



Average blocks to criterion

```
d1 %>%
  filter(trial_type == "training") %>%
  mutate(condition = "baseline") %>%
  group_by(subids, condition) %>%
  summarise(block_reached_crit = max(as.numeric(block))) %>%
  filter(block_reached_crit != 15) %>% # remove ss who didn't reach criterion
  group_by(condition) %>%
  multi_boot_standard(column = "block_reached_crit") %>%
  knitr::kable()
```

Joining by: "condition"

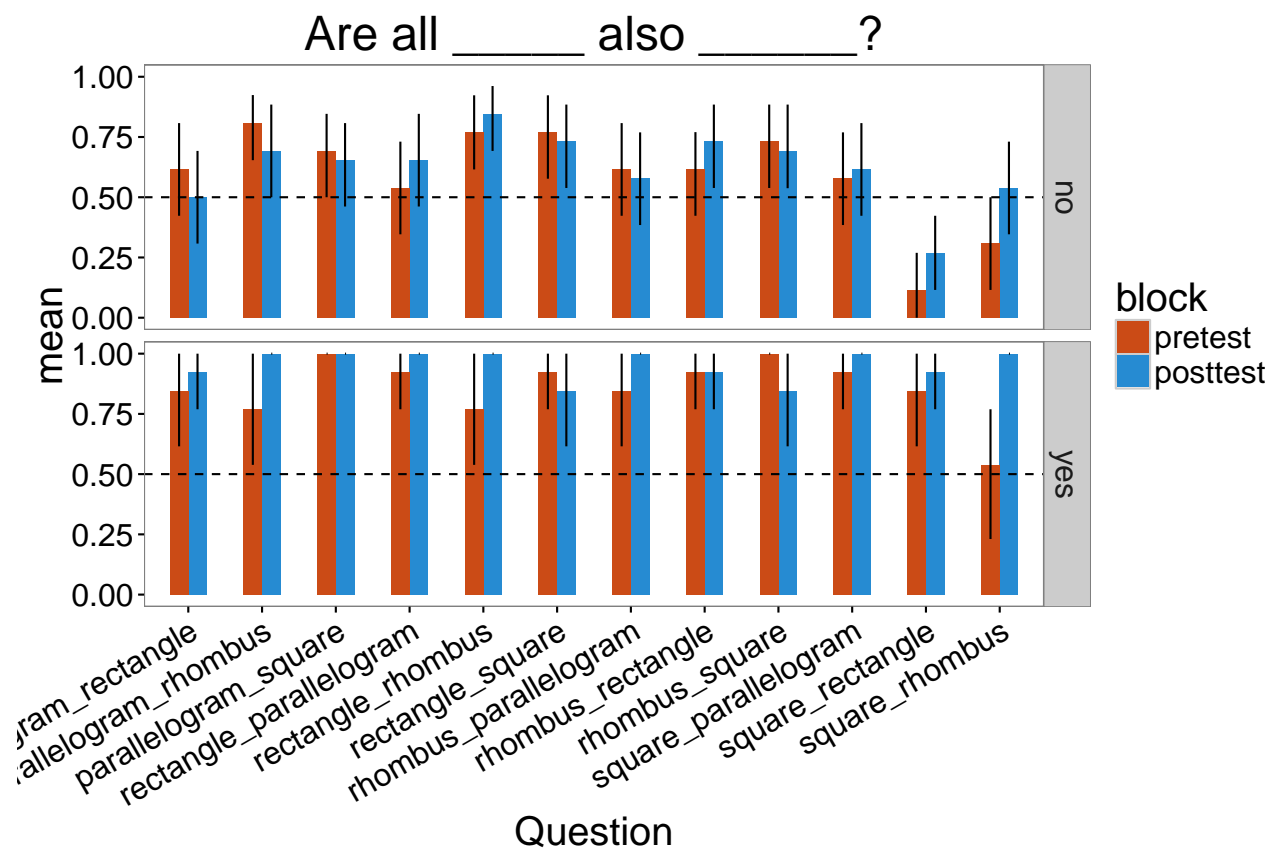
condition	mean	ci_lower	ci_upper
baseline	5.615385	4	7.384615

Relational test broken down by question and block

```
ms <- d1 %>%
  filter(trial_type %in% c("relational")) %>%
  group_by(question_and_shape, question, block = as.factor(block), reached_crit) %>%
  multi_boot_standard(column = "correct") %>%
  ungroup() %>%
  mutate(block = relevel(block, "pretest"))
```

```
## Joining by: c("question_and_shape", "question", "block", "reached_crit")
```

```
ggplot(aes(x = question_and_shape, y = mean, fill = block), data = ms) +
  geom_bar(stat="identity", position="dodge", width = 0.5) +
  geom_linerange(aes( ymin = ci_lower, ymax = ci_upper),
    position = position_dodge(width = 0.5)) +
  geom_hline(yintercept = .5, lty = 2) +
  ylim(0, 1) +
  xlab("Question") +
  scale_fill_solarized() +
  facet_grid(reached_crit~.) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1),
    text = element_text(size = 20)) +
  ggtitle("Are all _____ also _____?")
```



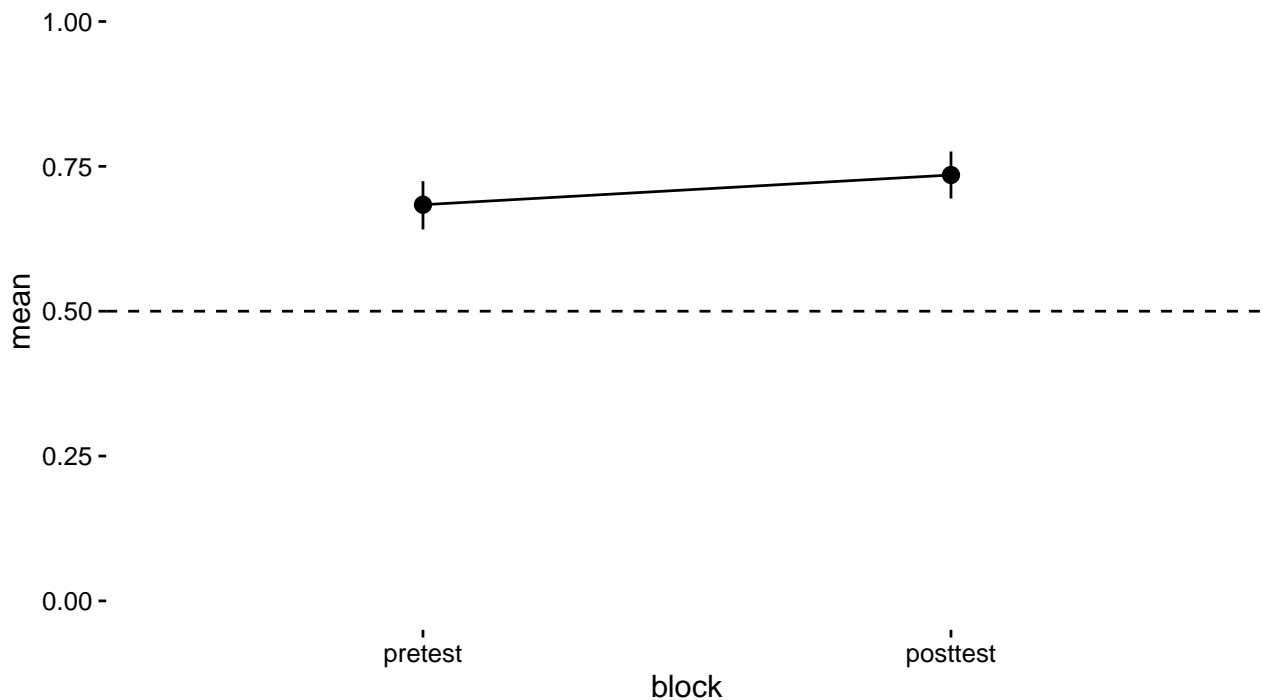
Overall accuracy analysis for relational test across blocks

```
ms <- d1 %>%
  filter(trial_type %in% c("relational")) %>%
  group_by(trial_type, block = as.factor(block)) %>%
  multi_boot_standard(column = "correct")
```

```
## Joining by: c("trial_type", "block")
```

```
ms %<>% mutate(chance_line = ifelse(trial_type == "relational", 0.5, 0.5),
              block = relevel(block, "pretest"))
```

```
ggplot(aes(x = block, y = mean), data = ms) +
  geom_pointrange(aes(ymin = ci_lower, ymax = ci_upper)) +
  geom_line(aes(group=1)) +
  geom_hline(aes(yintercept = chance_line), linetype = "dashed") +
  ylim(0,1.0) +
  scale_color_solarized()
```



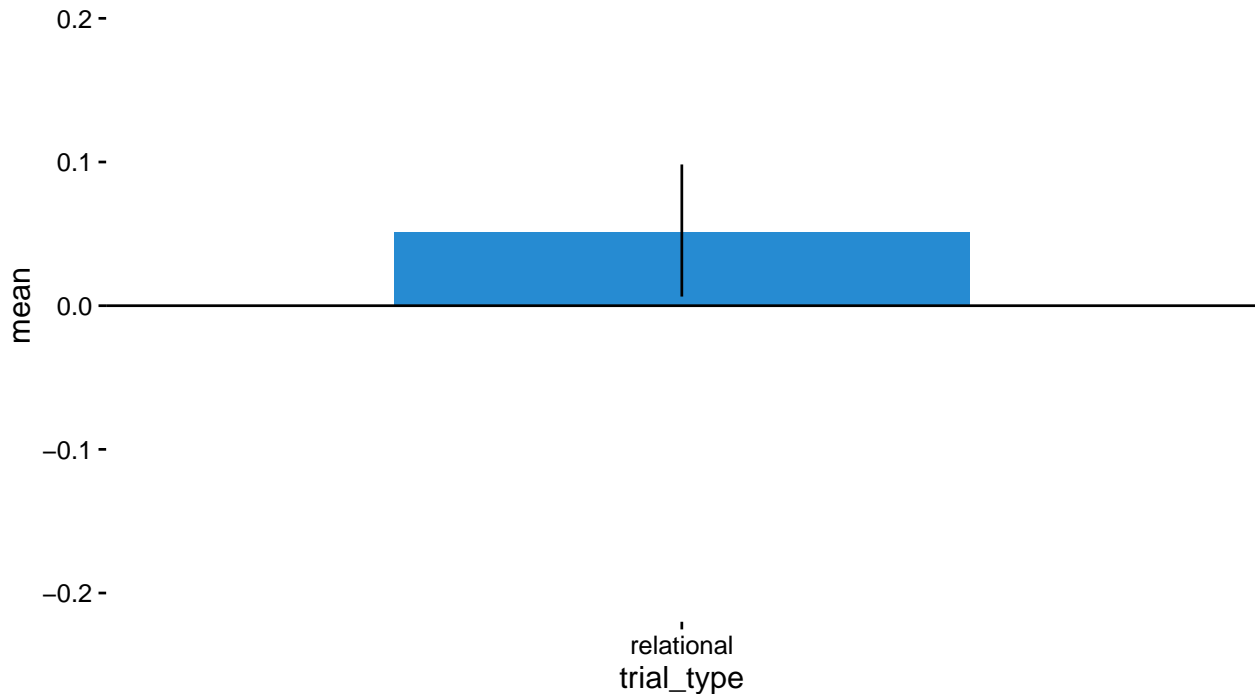
Within subjects change scores for all shapes

```
ss_acc <- d1 %>%
  filter(trial_type %in% c("relational")) %>%
  group_by(subids, block, trial_type, reached_crit) %>%
  summarise(m_acc = mean(correct)) %>%
  spread(key = block, value = m_acc) %>%
  mutate(m_diff_score = posttest - pretest)

ms_change <- ss_acc %>%
  group_by(trial_type) %>%
  multi_boot_standard(column = "m_diff_score")
```

```
## Joining by: "trial_type"
```

```
ggplot(aes(x = trial_type, y = mean, fill = trial_type),
      data = ms_change) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_linerange(aes(ymin = ci_lower, ymax = ci_upper)) +
  geom_hline(yintercept = 0) +
  scale_fill_solarized() +
  ylim(-0.2, 0.2) +
  guides(fill=F)
```



Analysis splitting by ss who reached criterion

Overall accuracy analysis

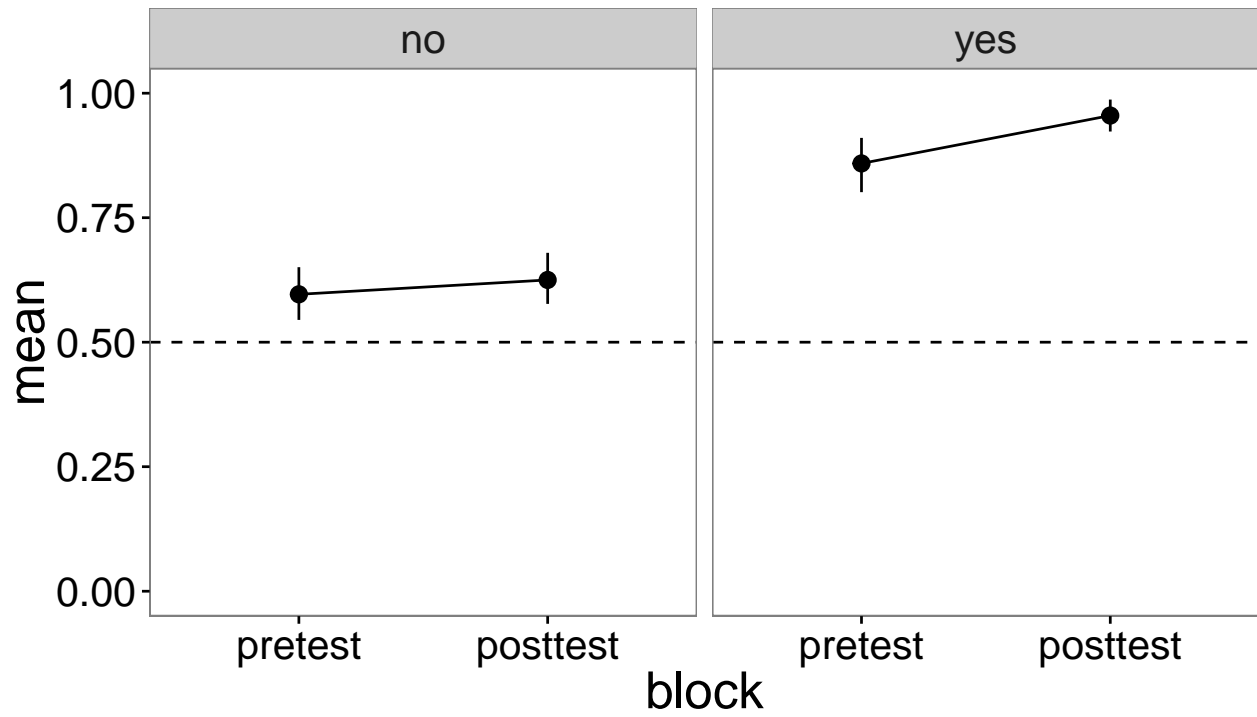
```
ms <- d1 %>%
  filter(trial_type %in% c("relational")) %>%
  group_by(trial_type, block = as.factor(block), reached_crit) %>%
  multi_boot_standard(column = "correct")
```

```
## Joining by: c("trial_type", "block", "reached_crit")
```

```
ms %<>% ungroup() %>% mutate(block = relevel(block, "pretest"))

ggplot(aes(x = block, y = mean), data = ms) +
  geom_pointrange(aes(ymin = ci_lower, ymax = ci_upper)) +
  geom_line(aes(group=1)) +
  geom_hline(aes(yintercept = 0.5), linetype = "dashed") +
  ylim(0,1.0) +
  facet_grid(.~reached_crit) +
```

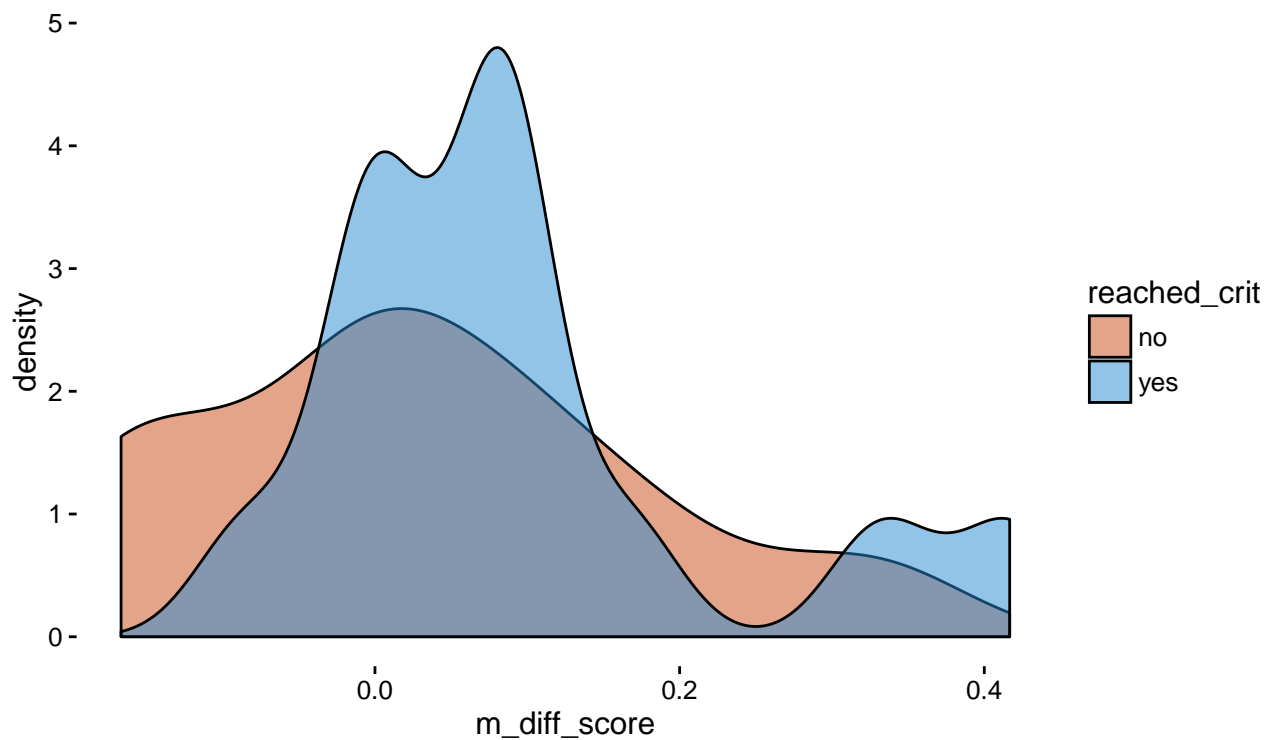
```
scale_color_solarized() +
theme_bw() +
theme(text = element_text(size = 20))
```



Within subjects change scores

Distribution of change scores split by if ss reached criterion.

```
ggplot(data = ss_acc, aes(x = m_diff_score, fill = reached_crit)) +
  geom_density(alpha = 0.5) +
  scale_fill_solarized()
```



```
ss_acc <- d1 %>%
  filter(trial_type %in% c("relational")) %>%
  group_by(subids, block, trial_type, reached_crit) %>%
  summarise(m_acc = mean(correct)) %>%
  spread(key = block, value = m_acc) %>%
  mutate(m_diff_score = posttest - pretest)

# filter out ss 3SD away from mean
ss_acc %<>%
  mutate(include = ifelse(abs(m_diff_score) > sd(ss_acc$m_diff_score) * 3 +
    mean(ss_acc$m_diff_score),
    "no", "yes"))

ms_change <- ss_acc %>%
  filter(include == "yes") %>%
  group_by(trial_type, reached_crit) %>%
  multi_boot_standard(column = "m_diff_score")
```

```
## Joining by: c("trial_type", "reached_crit")
```

```
ggplot(aes(x = trial_type, y = mean, fill = trial_type),
  data = ms_change) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_linerange(aes(ymin = ci_lower, ymax = ci_upper)) +
  facet_grid(.~reached_crit) +
  geom_hline(yintercept = 0) +
  scale_fill_solarized() +
  ylim(-0.2, 0.2) +
  guides(fill=F) +
  theme_bw()
```