

Task Switching Replication

Laura Marusich and Jon Bakdash

2022-04-06

Read in and Check Raw Data

```
#Import current data
task_switching_raw <- read.csv(paste0(workingdir, "/task-switching-replication-recoded-2.csv"))
head(task_switching_raw)
```

	participant	session	condition	trialType	posture	blockNum	trialNum	switchTrialType
## 1	1	1	1	experiment	standing	1	1	buffer
## 2	1	1	1	experiment	standing	1	2	noswitch
## 3	1	1	1	experiment	standing	1	3	switch
## 4	1	1	1	experiment	standing	1	4	noswitch
## 5	1	1	1	experiment	standing	1	5	noswitch
## 6	1	1	1	experiment	standing	1	6	noswitch

	congruentTrialType	cueType	shapeType	shapeColor	response	correctResponse	correct	reactionTime
## 1	incongruent	solid	square	blue	right	left	no	0.9088130
## 2	incongruent	solid	square	blue	left	left	yes	0.5947349
## 3	incongruent	dashed	square	blue	right	right	yes	0.7084870
## 4	incongruent	dashed	square	blue	right	right	yes	0.5995200
## 5	congruent	dashed	square	yellow	right	right	yes	0.4399409
## 6	congruent	dashed	square	yellow	right	right	yes	0.3847258

	date	utcTime
## 1	2021-11-10 10:22:00	1636561737
## 2	2021-11-10 10:22:00	1636561744
## 3	2021-11-10 10:22:00	1636561746
## 4	2021-11-10 10:22:00	1636561748
## 5	2021-11-10 10:22:00	1636561750
## 6	2021-11-10 10:22:00	1636561752

```
#does every person have 392 trials?
ntrials_sub <- task_switching_raw %>%
  group_by(participant) %>%
  summarize(ntrials = n()) %>%
  pull(ntrials)

all(ntrials_sub == 392)

## [1] TRUE

#does every block start with a buffer and have 49 trials?
task_switching_raw <- task_switching_raw %>%
  mutate(condblock = paste0(posture, blockNum))

blocktrials <- task_switching_raw %>%
```

```

group_by(participant, condblock) %>%
  summarize(ntrials = n(), firsttrial = first(switchTrialType))

## `summarise()` has grouped output by 'participant'. You can override using the `.groups` argument.
all(blocktrials$ntrials == 49)

## [1] TRUE

all(blocktrials$firsttrial == "buffer")

## [1] TRUE

```

Clean Data

```

#Drop buffer trials
task_switching_raw2 <- task_switching_raw %>%
  filter(switchTrialType != "buffer")

#Recode Correct to 1 and Incorrect to 0
task_switching_raw2$correct_bin <- recode(task_switching_raw2$correct,
                                          "no" = 0,
                                          "yes" = 1)

#Calc overall acc by participant
ts_overall_acc <- task_switching_raw2 %>%
  group_by(participant) %>%
  summarize(Accuracy = mean(correct_bin))

#find participants with less than 80% accuracy
#2, 8, 15, 44, 49, 51
low_acc_subs <- ts_overall_acc %>% filter(Accuracy < 0.80) %>%
  pull(participant)

task_switching_raw3 <- task_switching_raw2 %>%
  filter(!(participant %in% low_acc_subs))

#Calc mean Acc by participant and conditions (posture, con, switch)
#Narrow format
ts_acc_mean <- task_switching_raw3 %>%
  group_by(participant,
            posture,
            congruentTrialType,
            switchTrialType) %>%
  summarize(Accuracy = mean(correct_bin))

## `summarise()` has grouped output by 'participant', 'posture', 'congruentTrialType'. You can override

#Convert data to wide format (for statview/SPSS/etc)
ts_acc_mean_wide <- ts_acc_mean %>%
  pivot_wider(names_from = c(posture,
                              congruentTrialType,
                              switchTrialType),
              values_from = Accuracy)

```

```

write.csv(ts_acc_mean_wide, file = "new_recoded_for_statview.csv", row.names = F)

ts_acc_mean <- data.frame(ts_acc_mean)
ts_acc_mean$posture <- as.factor(ts_acc_mean$posture)
ts_acc_mean$participant <- as.factor(ts_acc_mean$participant)
ts_acc_mean$congruentTrialType <- as.factor(ts_acc_mean$congruentTrialType)
ts_acc_mean$switchTrialType <- as.factor(ts_acc_mean$switchTrialType)
str(ts_acc_mean)

## 'data.frame': 408 obs. of 5 variables:
## $ participant : Factor w/ 51 levels "1","3","4","5",...: 1 1 1 1 1 1 1 1 2 2 ...
## $ posture : Factor w/ 2 levels "sitting","standing": 1 1 1 1 2 2 2 2 1 1 ...
## $ congruentTrialType: Factor w/ 2 levels "congruent","incongruent": 1 1 2 2 1 1 2 2 1 1 ...
## $ switchTrialType : Factor w/ 2 levels "noswitch","switch": 1 2 1 2 1 2 1 2 1 2 ...
## $ Accuracy : num 0.96 0.978 0.957 0.88 0.981 ...

#Total N = 51 (6 dropped for acc < 80%)
length(unique(ts_acc_mean$participant))

## [1] 51

```

Summarize Demographics

```

demo_raw <- read.csv(paste0(workingdir, "/Task Switching_February 24, 2022_13.05.csv"), skip = 1) %>%
  slice(-1) %>%
  select(-c(Response.Type, IP.Address, Recipient.Last.Name:Distribution.Channel))
colnames(demo_raw)[10:15] <- c("Gender.Pick", "Gender.Text", "Age", "Race.Pick", "Race.Text", "Eng.First")

dim(demo_raw)

## [1] 59 15

#59 records
#first two are test data
# need to match up the 6 dropped participants from behavioral data
demo_df <- demo_raw %>%
  filter(!(X %in% c("test", low_acc_subs)))
dim(demo_df)

## [1] 51 15

demo_df <- demo_df %>%
  mutate(Gender.New = ifelse(Gender.Pick %in% c("Man", "Woman"), Gender.Pick, Gender.Text),
         Eng.First = toupper(Eng.First))

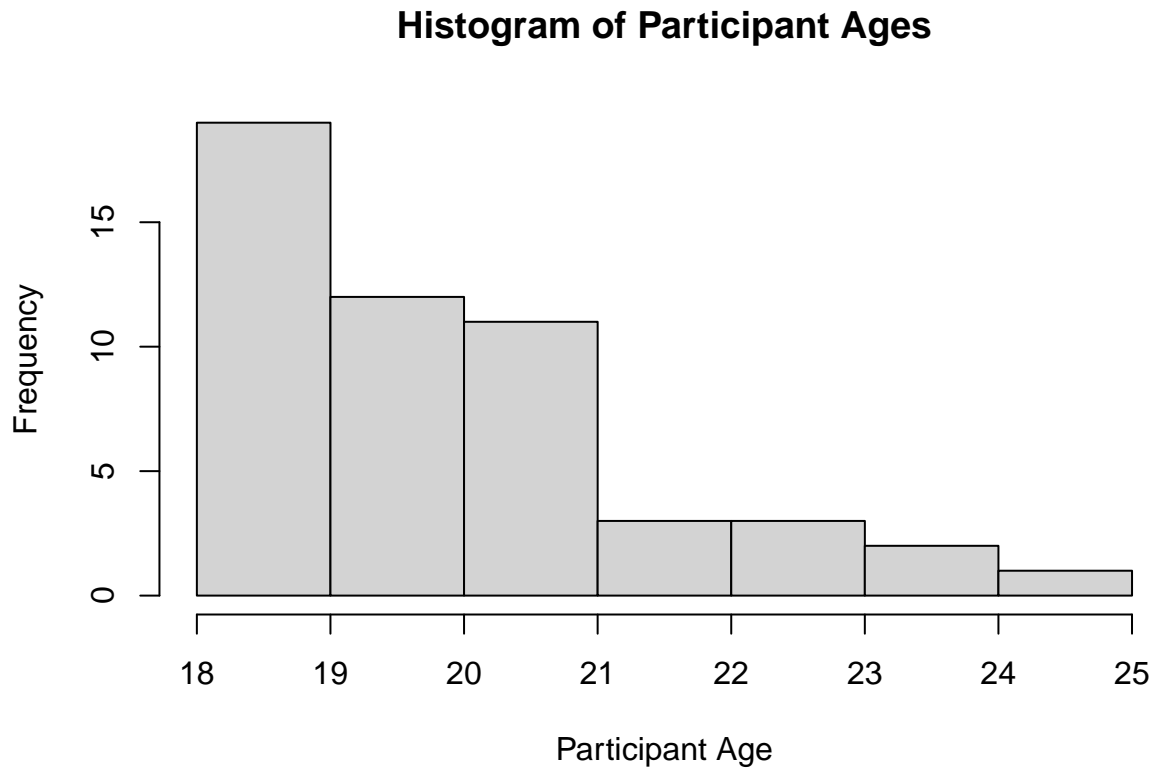
#gender breakdown
gender_table <- demo_df %>%
  group_by(Gender.New) %>%
  summarize(n = n())
gender_table

## # A tibble: 3 x 2
##   Gender.New      n
##   <chr>      <int>
## 1 Man        23

```

```
## 2 non binaary      1
## 3 Woman           27

#age breakdown
hist(as.numeric(demo_df$Age),
     main = "Histogram of Participant Ages",
     xlab = "Participant Age")
```



```
age_table <- demo_df %>%
  group_by(Age) %>%
  summarize(n = n())
age_table
```

```
## # A tibble: 8 x 2
##   Age      n
##   <chr> <int>
## 1 18      9
## 2 19     10
## 3 20     12
## 4 21     11
## 5 22      3
## 6 23      3
## 7 24      2
## 8 25      1
```

```
#age mean and sd
mean_age <- mean(as.numeric(demo_df$Age))
sd_age <- sd(as.numeric(demo_df$Age))
```

```
kable(matrix(c(mean_age, sd_age), nrow = 1), col.names = c("Mean of Age", "SD of Age"))
```

Mean of Age	SD of Age
20.21569	1.73567

```
#race breakdown
```

```
race_table <- demo_df %>%
  group_by(Race.Pick) %>%
  summarize(n = n()) %>%
  arrange(desc(n))
race_table
```

```
## # A tibble: 6 x 2
##   Race.Pick          n
##   <chr>          <int>
## 1 White /European American    22
## 2 Black / African American    11
## 3 Hispanic/Latino/Latina/Latinx 11
## 4 Asian /South Pacific Islander   3
## 5 Central Asian /Indian /Pakistani 3
## 6 Native American / American Indian 1
```

```
#language breakdown
```

```
lang_table <- demo_df %>%
  group_by(Eng.First) %>%
  summarize(n=n())
lang_table
```

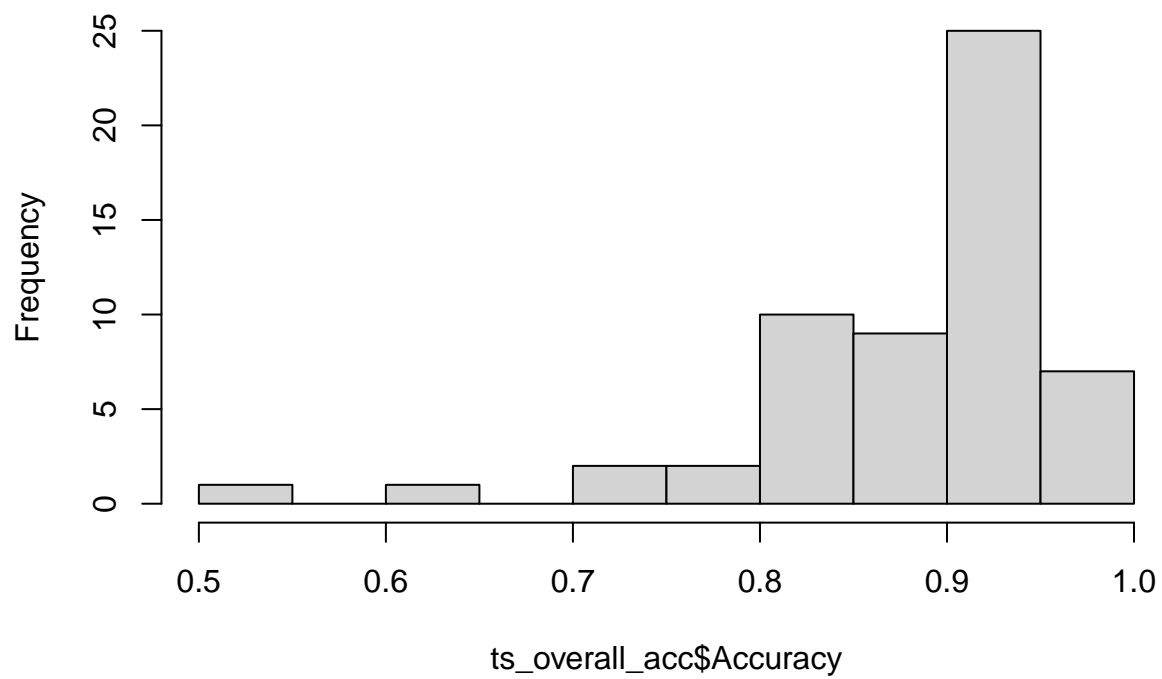
```
## # A tibble: 2 x 2
##   Eng.First    n
##   <chr>    <int>
## 1 NO        8
## 2 YES       43
```

Plots and Analyses

Accuracy

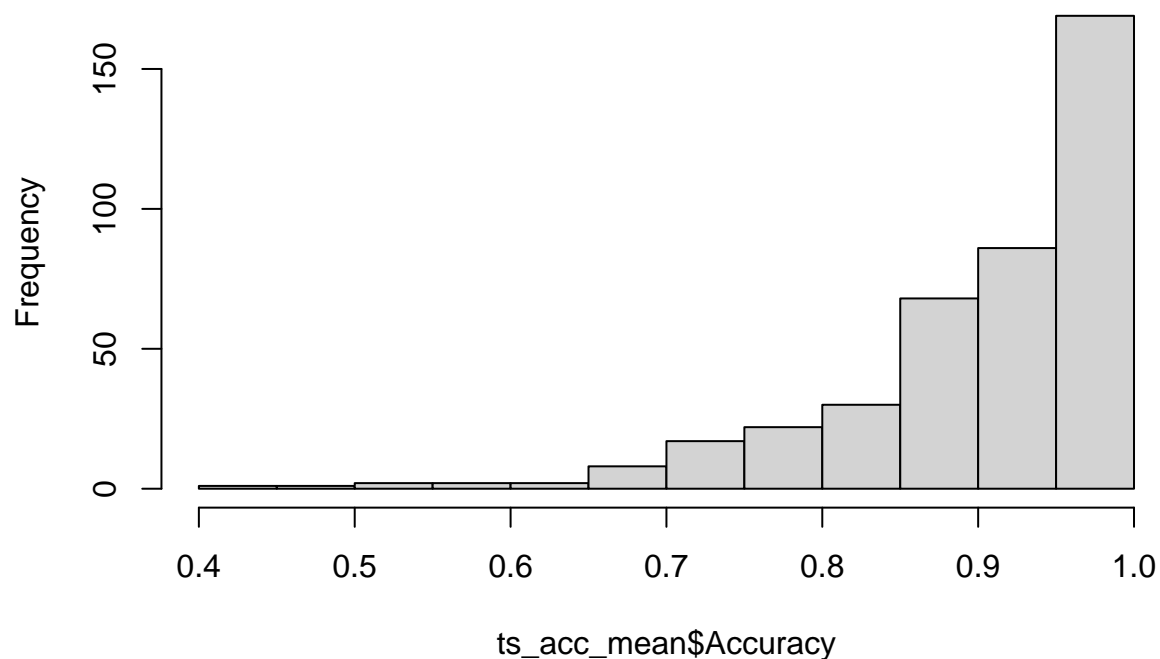
```
#Accuracy by participant, before drops
hist(ts_overall_acc$Accuracy)
```

Histogram of ts_overall_acc\$Accuracy



```
#Accuracy for all cells  
hist(ts_acc_mean$Accuracy)
```

Histogram of ts_acc_mean\$Accuracy



```
exp1_anova <- ezANOVA(ts_acc_mean,
  dv = Accuracy,
  wid = participant,
  within = .(posture, congruentTrialType, switchTrialType),
  type = 3,
  detailed = TRUE,
  return_aov = T
)

exp1_anova2 <- aov_ez(data = ts_acc_mean,
  dv = "Accuracy",
  id = "participant",
  within = c("posture", "congruentTrialType", "switchTrialType"),
  type = 3,
  anova_table = list(es = "pes")
)

exp1_stats <- ezStats(ts_acc_mean,
  dv = Accuracy,
  wid = participant,
  within = .(posture, congruentTrialType, switchTrialType),
  type = 3
)

write.csv(exp1_stats[, -7], file = "Task_Switching_Descriptives_ACC.csv", row.names = F)

#data.frame(exp1_anova$ANOVA)
```

```

output_acc <- aovEffectSize(exp1_anova, effectSize = "pes")
output_acc <- data.frame(output_acc$ANOVA) %>%
  mutate(MSE = c(NA, exp1_anova2$anova_table$MSE), .before = F)
write.csv(output_acc, "Task_switching_ANOVA_acc.csv")
output_acc

```

```

##              Effect DFn DFd          SSn          SSd          MSE
## 1      (Intercept)    1   50 3.331616e+02 0.84392719          NA
## 2           posture    1   50 9.386943e-03 0.44204149 0.008840830
## 3 congruentTrialType    1   50 8.577579e-01 0.43034136 0.008606827
## 4      switchTrialType    1   50 2.633377e-01 0.14305407 0.002861081
## 5 posture:congruentTrialType    1   50 1.038526e-04 0.20920092 0.004184018
## 6 posture:switchTrialType    1   50 1.767731e-03 0.12000107 0.002400021
## 7 congruentTrialType:switchTrialType    1   50 8.851220e-02 0.07573624 0.001514725
## 8 posture:congruentTrialType:switchTrialType    1   50 2.360230e-03 0.09401427 0.001880285
##              F              p p..05      pes
## 1 1.973876e+04 1.301952e-66      * 0.9974733137
## 2 1.061772e+00 3.077703e-01      0.0207938669
## 3 9.966017e+01 1.702444e-13      * 0.6659097700
## 4 9.204131e+01 6.367282e-13      * 0.6479897331
## 5 2.482125e-02 8.754478e-01      0.0004961787
## 6 7.365482e-01 3.948644e-01      0.0145171124
## 7 5.843451e+01 5.925302e-10      * 0.5388921804
## 8 1.255251e+00 2.679047e-01      0.0244901935

```

```

#Calculate confidence interval: PES for posture x switch/condition interaction
#using ANOVA results (partial eta-squared)
interaction_effect_CI <-
  get.ci.partial.eta.squared(exp1_anova$ANOVA$F[6],
                             exp1_anova$ANOVA$DFn[6],
                             exp1_anova$ANOVA$DFd[6],
                             conf.level = 0.90) #90% CI is the convention for PES
interaction_effect_CI

```

```

## $LL
## [1] 0
##
## $UL
## [1] 0.1073579

```

```

# This almost works for getting CIs around all effects....
# test.model <-
#   aov_car(Accuracy ~
#           posture*congruentTrialType*switchTrialType +
#           Error(participant/posture*congruentTrialType *switchTrialType),
#           ts_acc_mean, anova_table = (es = "pes"),
#           return = "aov",
#           include_aov = T,
#           type = 3)
# # test.model
# anova_stats(test.model)

```

```

#The CIs around effect sizes are calculated using Type I sums of squares?
#' ## Type of Sums of Squares
#' The sums of squares (or *F* statistics) used for the computation of the

```



```

# ' effect sizes is based on those returned by `anova(model)` (whatever those may
# ' be - for `aov` and `aovlist` these are *type-1* sums of squares; for
# ' `lmerMod` (and `lmerModLmerTest`) these are *type-3* sums of squares)
# test.pes<-
# effectsize::eta_squared(test.model,
#                           alternative = "two.sided",
#                           ci = 0.90,
#                           partial = T,
#                           verbose = T)
#
#https://mran.microsoft.com/snapshot/2018-06-30/web/packages/sjstats/vignettes/anova-statistics.html

#Below function won't work with an afex or aov object for an rm ANOVA
#car::Anova(..., type = 3)

congruent.labs <- c("Congruent", "Incongruent")
names(congruent.labs) <- c("1", "2")

#make plot like Smith et al's
acc_plot <-
  superbPlot(ts_acc_mean_wide,
             WSFactors = c("Condition(2)", "Congruent(2)", "Posture(2)"),
             variables = colnames(ts_acc_mean_wide)[2:9],
             errorbar = "SE", #Tempted to change to CI, should stay SE to be consistent with SMith
             plotStyle = "line",
             factorOrder = c("Condition", "Posture", "Congruent"),
             adjustments = list(purpose = "difference"))+
  theme_classic() +
  ylim(0.77, 1) + #Trying to make ylim same as the Smith w/o cutting off error bars
  facet_wrap(vars(Congruent), labeller = labeller(Congruent = congruent.labs)) +
  scale_x_discrete(labels=c("1" = "No Switch", "2" = "Switch"))+
  scale_color_manual(values=c("#E69F00", "#0072B2"),
                     labels = c("Sitting", "Standing")) +
  labs(y = "Accuracy")

```

superb::FYI: Here is how the within-subject variables are understood:

```

## Condition Congruent Posture variable
##      1      1      1 sitting_congruent_noswitch
##      2      1      1 sitting_congruent_switch
##      1      2      1 sitting_incongruent_noswitch
##      2      2      1 sitting_incongruent_switch
##      1      1      2 standing_congruent_noswitch
##      2      1      2 standing_congruent_switch
##      1      2      2 standing_incongruent_noswitch
##      2      2      2 standing_incongruent_switch

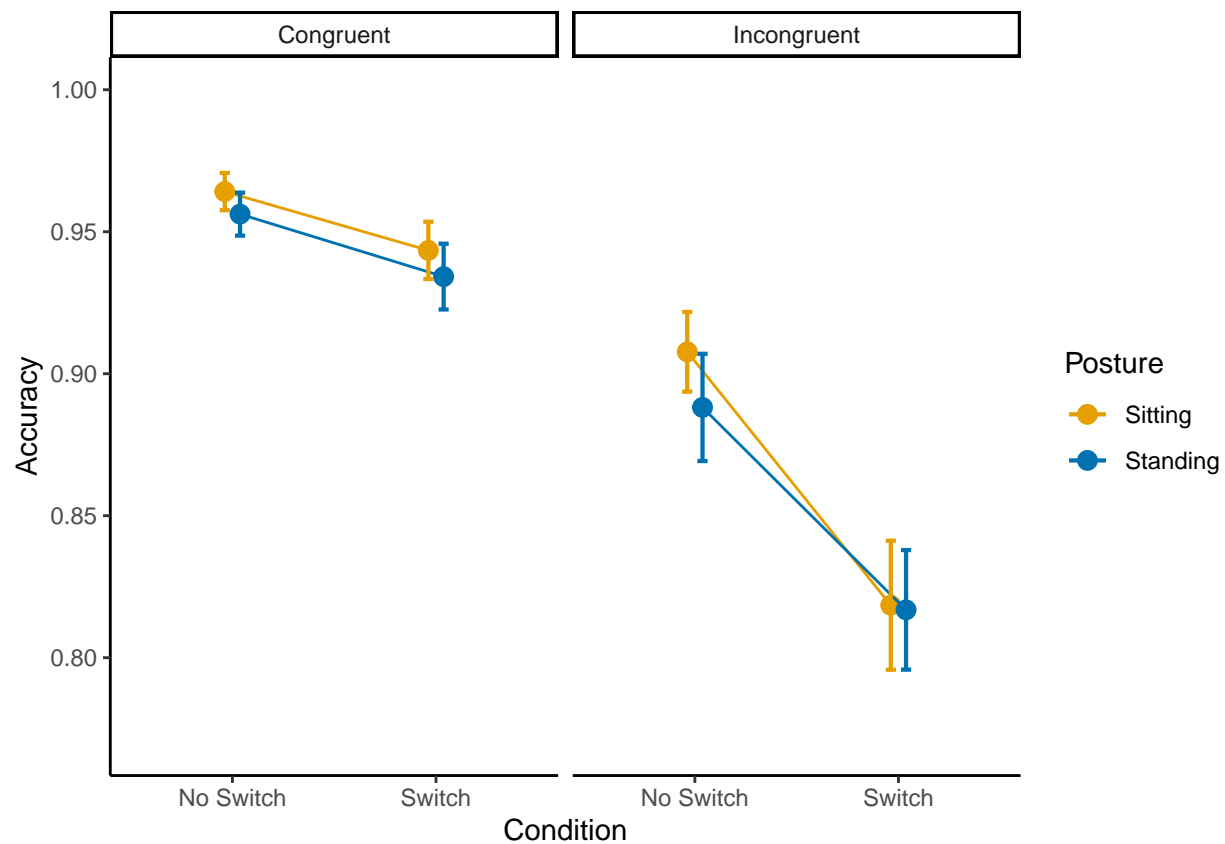
```

```

ggsave(acc_plot,
       file = "plots/acc_plot.pdf",
       units = "in",
       width = 6.62,
       height = 5.50,
       dpi = 600)

```

```
acc_plot
```



RT for correct trials only

```
#look at reaction time for correct trials?
ts_correct_only <- task_switching_raw3 %>%
  filter(correct_bin == 1)

ts_rt_mean <- ts_correct_only %>%
  group_by(participant,
            posture,
            congruentTrialType,
            switchTrialType) %>%
  summarize(mean_rt = mean(reactionTime))
```

`summarise()` has grouped output by 'participant', 'posture', 'congruentTrialType'. You can override

```
#Convert data to wide format
ts_rt_mean_wide <- ts_rt_mean %>%
  pivot_wider(names_from = c(posture,
                              congruentTrialType,
                              switchTrialType),
              values_from = mean_rt)
```

```
RT_plot <-
  superbPlot(ts_rt_mean_wide,
             WSFactors = c("Condition(2)", "Congruent(2)", "Posture(2)"),
```

```

    variables = colnames(ts_acc_mean_wide)[2:9],
    errorbar = "SE",
    plotStyle = "line",
    factorOrder = c("Condition", "Posture", "Congruent"),
    adjustments = list(purpose = "difference"))+
theme_classic()+
facet_wrap(vars(Congruent), labeller = labeller(Congruent = congruent.labs)) +
scale_x_discrete(labels=c("1" = "No Switch", "2" = "Switch"))+
scale_color_manual(values=c("#E69F00", "#0072B2"), labels = c("Sitting", "Standing")) +
labs(y = "Reaction Time (ms)")

```

superb::FYI: Here is how the within-subject variables are understood:

```

## Condition Congruent Posture variable
##      1      1      1 sitting_congruent_noswitch
##      2      1      1 sitting_congruent_switch
##      1      2      1 sitting_incongruent_noswitch
##      2      2      1 sitting_incongruent_switch
##      1      1      2 standing_congruent_noswitch
##      2      1      2 standing_congruent_switch
##      1      2      2 standing_incongruent_noswitch
##      2      2      2 standing_incongruent_switch

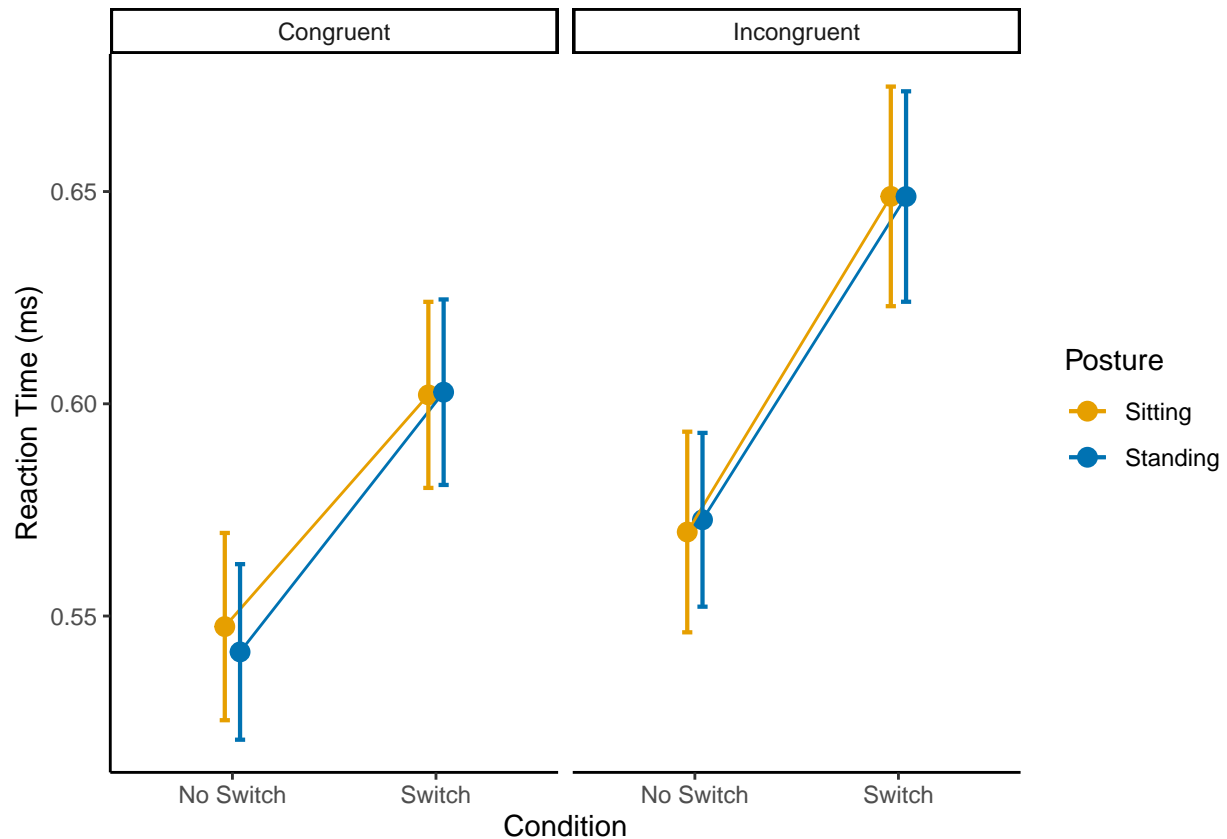
```

```

ggsave(RT_plot,
  file = "plots/RT_plot.pdf",
  units = "in",
  width = 6.62,
  height = 5.50,
  dpi = 600)

```

RT_plot



```

exp1_anova_rt <- ezANOVA(ts_rt_mean,
                        dv = mean_rt,
                        wid = participant,
                        within = .(posture, congruentTrialType, switchTrialType),
                        type = 3,
                        detailed = TRUE,
                        return_aov = T
)

## Warning: Converting "participant" to factor for ANOVA.
## Warning: Converting "posture" to factor for ANOVA.
## Warning: Converting "congruentTrialType" to factor for ANOVA.
## Warning: Converting "switchTrialType" to factor for ANOVA.

exp1_anova_rt2 <- aov_ez(data = ts_rt_mean,
                        dv = 'mean_rt',
                        id = 'participant',
                        within = c('posture', 'congruentTrialType', 'switchTrialType'),
                        type = 3
)

exp1_stats_rt <- ezStats(ts_rt_mean,
                        dv = mean_rt,
                        wid = participant,
                        within = .(posture, congruentTrialType, switchTrialType),

```

```

                                type = 3
)

## Warning: Converting "participant" to factor for ANOVA.
## Warning: Converting "posture" to factor for ANOVA.
## Warning: Converting "congruentTrialType" to factor for ANOVA.
## Warning: Converting "switchTrialType" to factor for ANOVA.
write.csv(exp1_stats_rt[, -7], file = "Task_Switching_Descriptives_RT.csv", row.names = F)

#data.frame(exp1_anova$ANOVA)
output_rt <- aovEffectSize(exp1_anova_rt, effectSize = "pes")
output_rt <- data.frame(output_rt$ANOVA) %>%
  mutate(MSE = c(NA, exp1_anova_rt2$anova_table$MSE), .before = F)
write.csv(output_rt, "Task_switching_ANOVA_RT.csv")
output_rt

```

	Effect	DFn	DFd	SSn	SSd	MSE
## 1	(Intercept)	1	50	1.428705e+02	4.29367155	NA
## 2	posture	1	50	3.979595e-05	0.43347417	0.0086694834
## 3	congruentTrialType	1	50	1.363938e-01	0.14478068	0.0028956136
## 4	switchTrialType	1	50	4.682022e-01	0.16485963	0.0032971927
## 5	posture:congruentTrialType	1	50	4.266607e-04	0.06545830	0.0013091659
## 6	posture:switchTrialType	1	50	8.700576e-05	0.05211061	0.0010422122
## 7	congruentTrialType:switchTrialType	1	50	9.914733e-03	0.04607197	0.0009214393
## 8	posture:congruentTrialType:switchTrialType	1	50	5.769395e-04	0.06995820	0.0013991640

	F	p	p .05	pes
## 1	1.663733e+03	4.809845e-40	*	9.708239e-01
## 2	4.590349e-03	9.462532e-01		9.179855e-05
## 3	4.710359e+01	9.823661e-09	*	4.850860e-01
## 4	1.420003e+02	3.195952e-16	*	7.395837e-01
## 5	3.259027e-01	5.706388e-01		6.475845e-03
## 6	8.348180e-02	7.738279e-01		1.666853e-03
## 7	1.076005e+01	1.893419e-03	*	1.770909e-01
## 8	4.123459e-01	5.237146e-01		8.179462e-03

Generate RT analysis/plots for trimmed data

```

#perhaps double check trimmed output with trimr package?

trimmingOutput = pjRecursiveTrim2(dataSet = ts_correct_only,
                                   dv = "reactionTime",
                                   splitvars = c("participant",
                                                  "posture",
                                                  "switchTrialType",
                                                  "congruentTrialType"))

trimmedData=trimmingOutput[[1]]
totalN = trimmingOutput[[2]]
rejected = trimmingOutput[[3]]
percentTrimmed = trimmingOutput[[4]] #this is very close to the percentage trimmed for stroop
Ncells = trimmingOutput[[5]] # 51 participants * 8 conditions

#now use trimmedData for same analyses as above

```

```

trimmed_rt_mean <- trimmedData %>%
  group_by(participant,
            posture,
            congruentTrialType,
            switchTrialType) %>%
  summarize(mean_rt = mean(reactionTime))

## `summarise()` has grouped output by 'participant', 'posture', 'congruentTrialType'. You can override
#Convert data to wide format
trimmed_rt_mean_wide <- trimmed_rt_mean %>%
  pivot_wider(names_from = c(posture,
                             congruentTrialType,
                             switchTrialType),
              values_from = mean_rt)

trimmed_RT_plot <-
  superbPlot(trimmed_rt_mean_wide,
             WSFactors = c("Condition(2)", "Congruent(2)", "Posture(2)"),
             variables = colnames(ts_acc_mean_wide)[2:9],
             errorbar = "SE",
             plotStyle = "line",
             factorOrder = c("Condition", "Posture", "Congruent"),
             adjustments = list(purpose = "difference"))+
  theme_classic()+
  facet_wrap(vars(Congruent), labeller = labeller(Congruent = congruent.labs)) +
  scale_x_discrete(labels=c("1" = "No Switch", "2" = "Switch"))+
  scale_color_manual(values=c("#E69F00", "#0072B2"), labels = c("Sitting", "Standing")) +
  labs(y = "Reaction Time (ms)")

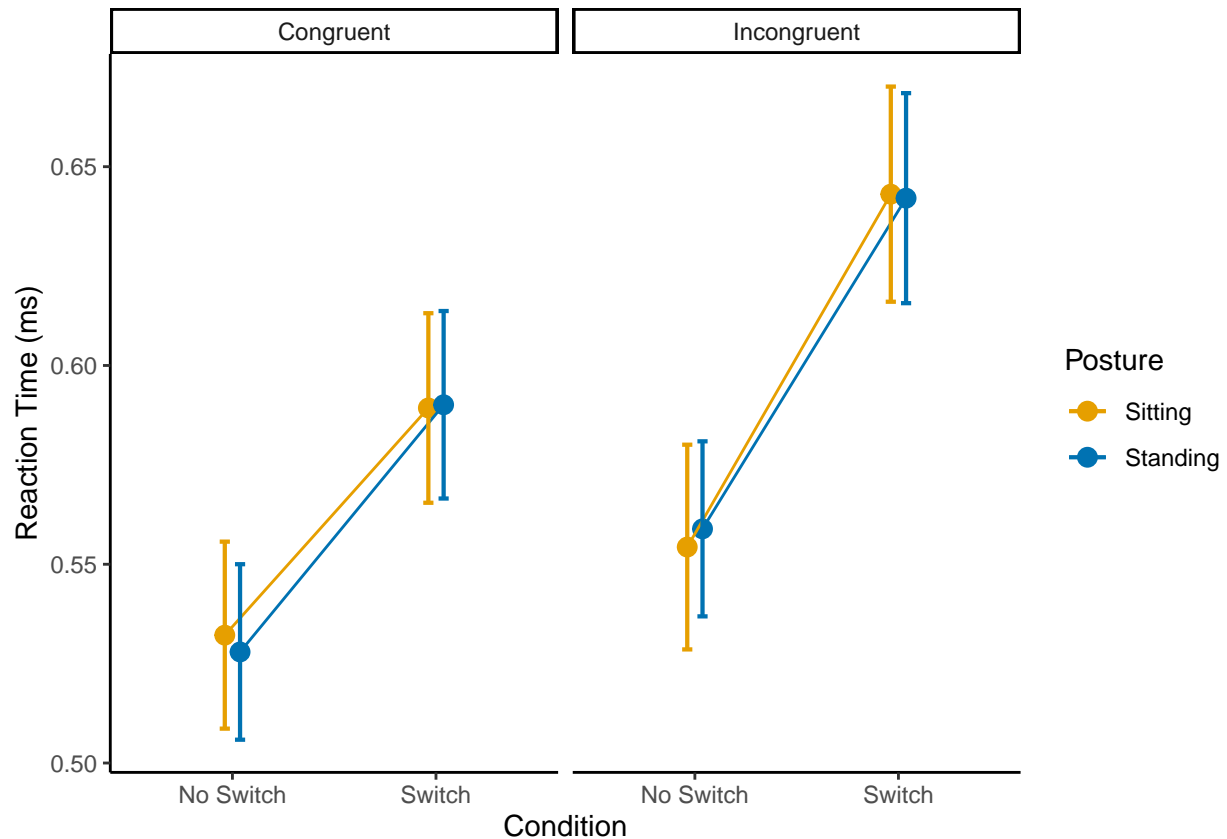
## superb::FYI: Here is how the within-subject variables are understood:

##   Condition Congruent Posture          variable
##         1         1         1 sitting_congruent_noswitch
##         2         1         1 sitting_congruent_switch
##         1         2         1 sitting_incongruent_noswitch
##         2         2         1 sitting_incongruent_switch
##         1         1         2 standing_congruent_noswitch
##         2         1         2 standing_congruent_switch
##         1         2         2 standing_incongruent_noswitch
##         2         2         2 standing_incongruent_switch

ggsave(trimmed_RT_plot,
        file = "plots/trimmed_RT_plot.pdf",
        units = "in",
        width = 6.62,
        height = 5.50,
        dpi = 600)

trimmed_RT_plot

```



```

exp1_anova_trimmed_rt <- ezANOVA(trimmed_rt_mean,
  dv = mean_rt,
  wid = participant,
  within = .(posture, congruentTrialType, switchTrialType),
  type = 3,
  detailed = TRUE,
  return_aov = T
)

## Warning: Converting "participant" to factor for ANOVA.
## Warning: Converting "posture" to factor for ANOVA.
## Warning: Converting "congruentTrialType" to factor for ANOVA.
## Warning: Converting "switchTrialType" to factor for ANOVA.

exp1_anova_trimmed_rt2 <- aov_ez(data = trimmed_rt_mean,
  dv = 'mean_rt',
  id = 'participant',
  within = c('posture', 'congruentTrialType', 'switchTrialType'),
  type = 3
)

exp1_stats_trimmed_rt <- ezStats(trimmed_rt_mean,
  dv = mean_rt,
  wid = participant,
  within = .(posture, congruentTrialType, switchTrialType),

```

```

    type = 3
)

## Warning: Converting "participant" to factor for ANOVA.
## Warning: Converting "posture" to factor for ANOVA.
## Warning: Converting "congruentTrialType" to factor for ANOVA.
## Warning: Converting "switchTrialType" to factor for ANOVA.
write.csv(exp1_stats_trimmed_rt[, -7],
          file = "Task_Switching_Descriptives_trimmed_RT.csv",
          row.names = F)

#data.frame(exp1_anova$ANOVA)
output_trimmed_rt <- aovEffectSize(exp1_anova_trimmed_rt, effectSize = "pes")
output_trimmed_rt <- data.frame(output_trimmed_rt$ANOVA) %>%
  mutate(MSE = c(NA, exp1_anova_trimmed_rt2$anova_table$MSE), .before = F)
write.csv(output_trimmed_rt, "Task_switching_ANOVA_trimmed_RT.csv")
output_trimmed_rt

```

```

##              Effect DFn DFd          SSn          SSd          MSE
## 1              (Intercept)    1  50 1.371287e+02 4.87941545          NA
## 2              posture      1  50 1.522652e-07 0.49708475 0.009941695
## 3      congruentTrialType    1  50 1.609093e-01 0.16613011 0.003322602
## 4      switchTrialType      1  50 5.408113e-01 0.20595811 0.004119162
## 5      posture:congruentTrialType    1  50 3.118425e-04 0.08040776 0.001608155
## 6      posture:switchTrialType    1  50 1.892588e-06 0.07609020 0.001521804
## 7      congruentTrialType:switchTrialType    1  50 1.765608e-02 0.06288865 0.001257773
## 8 posture:congruentTrialType:switchTrialType    1  50 7.189919e-04 0.07890550 0.001578110
##              F              p p .05              pes
## 1 1.405175e+03 2.877058e-38 * 9.656399e-01
## 2 1.531582e-05 9.968930e-01 3.063164e-07
## 3 4.842871e+01 6.954280e-09 * 4.920181e-01
## 4 1.312916e+02 1.355306e-15 * 7.242012e-01
## 5 1.939132e-01 6.615774e-01 3.863281e-03
## 6 1.243648e-03 9.720085e-01 2.487233e-05
## 7 1.403757e+01 4.648099e-04 * 2.192084e-01
## 8 4.556032e-01 5.027955e-01 9.029784e-03

```

Reproduce ANOVA for Smith et al. task-switching data

```

#load acc data
Smith_Exp2_acc <- read_excel("StandingData.xlsx",
                             sheet = "Exp2Acc")

head(Smith_Exp2_acc)

## # A tibble: 6 x 9
##   subj sit_congruent_noswitch sit_congruent_sw sit_incongruent_~ sit_incongruent~ stand_congruent~
##   <chr>          <dbl>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 1              1              1              0.957          0.951          0.98
## 2 2              0.976          0.978          0.981          0.796          0.98
## 3 3              0.977          0.979          0.98           0.935          0.980
## 4 4              0.893          0.884          0.816          0.549          0.980

```



```
## 5 5          1          0.98          0.925          0.933          0.978
## 6 6          1          0.95          0.957          0.868          1
## # ... with 3 more variables: stand_congruent_switch <dbl>, stand_incongruent_noswitch <dbl>,
## #   stand_incongruent_switch <dbl>
```

```
#Drop last row with comment
```

```
Smith_Exp2_acc <- Smith_Exp2_acc[1:30,]
```

```
#Restructure from wide to narrow, using tidyr
```

```
Smith_exp2_acc_narrow <- Smith_Exp2_acc %>%
  gather(v, accuracy, colnames(Smith_Exp2_acc)[2:9]) %>%
  separate(v, c("posture", "con", "switch")) %>%
  arrange(subj)
```

```
Smith_exp2_acc_anova <- ezANOVA(Smith_exp2_acc_narrow,
                                dv = accuracy,
                                wid = subj,
                                within = .(posture, con, switch),
                                type = 3,
                                detailed = TRUE,
                                return_aov=F
)
```

```
## Warning: Converting "subj" to factor for ANOVA.
```

```
## Warning: Converting "posture" to factor for ANOVA.
```

```
## Warning: Converting "con" to factor for ANOVA.
```

```
## Warning: Converting "switch" to factor for ANOVA.
```

```
Smith_exp2_acc_anova
```

```
## $ANOVA
```

##	Effect	DFn	DFd	SSn	SSd	F	p	p<.05	ges
## 1	(Intercept)	1	29	2.048909e+02	0.24225540	2.452716e+04	5.184222e-44	*	0.9965856538
## 2	posture	1	29	1.215410e-02	0.12309940	2.863286e+00	1.013416e-01		0.0170197030
## 3	con	1	29	2.719991e-01	0.11703974	6.739569e+01	4.729349e-09	*	0.2792700999
## 4	switch	1	29	1.403032e-01	0.06464470	6.294084e+01	9.482835e-09	*	0.1665777615
## 5	posture:con	1	29	2.734424e-03	0.04708210	1.684256e+00	2.045882e-01		0.0038802688
## 6	posture:switch	1	29	7.099180e-03	0.03715534	5.540959e+00	2.556408e-02	*	0.0100120392
## 7	con:switch	1	29	3.248637e-02	0.04037101	2.333617e+01	4.059829e-05	*	0.0442321480
## 8	posture:con:switch	1	29	5.244392e-04	0.03031748	5.016491e-01	4.844309e-01		0.0007465437

```
output_Smith_exp2_acc <- aovEffectSize(Smith_exp2_acc_anova, effectSize = "pes")
```

```
output_Smith_exp2_acc <- data.frame(output_Smith_exp2_acc$ANOVA)
```

```
write.csv(output_Smith_exp2_acc, "Smith_Exp2_ANOVA_acc.csv")
```

```
output_Smith_exp2_acc
```

##	Effect	DFn	DFd	SSn	SSd	F	p	p..05	pes
## 1	(Intercept)	1	29	2.048909e+02	0.24225540	2.452716e+04	5.184222e-44	*	0.99881903
## 2	posture	1	29	1.215410e-02	0.12309940	2.863286e+00	1.013416e-01		0.08986162
## 3	con	1	29	2.719991e-01	0.11703974	6.739569e+01	4.729349e-09	*	0.69915667
## 4	switch	1	29	1.403032e-01	0.06464470	6.294084e+01	9.482835e-09	*	0.68457979
## 5	posture:con	1	29	2.734424e-03	0.04708210	1.684256e+00	2.045882e-01		0.05488989
## 6	posture:switch	1	29	7.099180e-03	0.03715534	5.540959e+00	2.556408e-02	*	0.16041707
## 7	con:switch	1	29	3.248637e-02	0.04037101	2.333617e+01	4.059829e-05	*	0.44588988

```
## 8 posture:con:switch 1 29 5.244392e-04 0.03031748 5.016491e-01 4.844309e-01 0.01700410
```

```
#plot acc effect sizes for our results and Smith together
```

```
#Check that ANOVA tables (rows) match
```

```
output_acc$Effect
```

```
## [1] "(Intercept)"          "posture"
## [3] "congruentTrialType"    "switchTrialType"
## [5] "posture:congruentTrialType" "posture:switchTrialType"
## [7] "congruentTrialType:switchTrialType" "posture:congruentTrialType:switchTrialType"
```

```
output_Smith_exp2_acc$Effect
```

```
## [1] "(Intercept)"          "posture"          "con"          "switch"
## [5] "posture:con"          "posture:switch"    "con:switch"    "posture:con:switch"
```

```
ts.acc.pes <- data.frame(matrix(nrow = 8, ncol = 3))
```

```
ts.acc.pes[1] <- output_acc$Effect
```

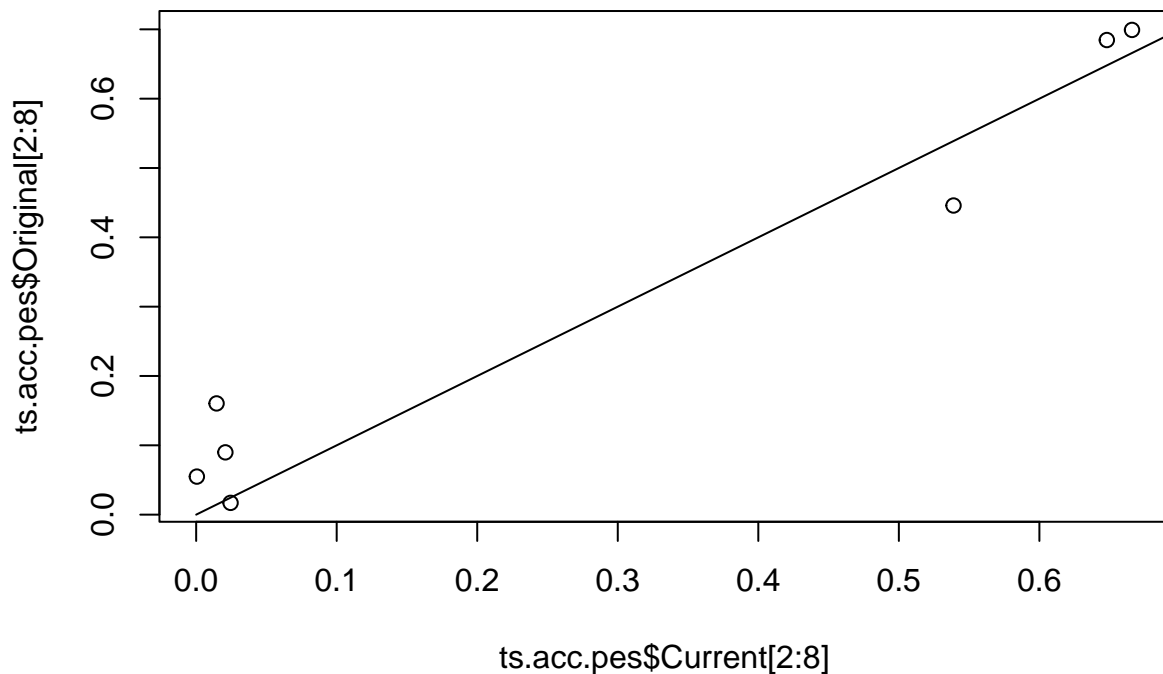
```
ts.acc.pes[2] <- output_acc$pes
```

```
ts.acc.pes[3] <- output_Smith_exp2_acc$pes
```

```
colnames(ts.acc.pes) <- c("Effect", "Current", "Original")
```

```
plot(ts.acc.pes$Current[2:8], ts.acc.pes$Original[2:8])
```

```
lines(x = c(0,1), y = c(0,1))
```



```
#Exact proportion of current to original effect: ~9%
```

```
ts.acc.pes$Current[6]/ts.acc.pes$Original[6]
```

```
## [1] 0.09049606
```

```

#Compare current vs. original effect sizes
interaction.original <-
  get.ci.partial.eta.squared(output_Smith_exp2_acc$F[6],
                             output_Smith_exp2_acc$DFn[6],
                             output_Smith_exp2_acc$DFd[6],
                             conf.level = 0.90)

ts.acc.pes.int <- data.frame(matrix(nrow = 2, ncol = 5))
colnames(ts.acc.pes.int) <- c("Exp", "Effect.size", "CI.Lower", "CI.Upper", "name")
ts.acc.pes.int[1] <- c("Current", "Original")
ts.acc.pes.int[1,2:4] <- c(ts.acc.pes$Current[6], data.frame(interaction_effect_CI))
ts.acc.pes.int[2,2:4] <- c(ts.acc.pes$Original[6], data.frame(interaction.original))
ts.acc.pes.int[1:2,5] <- c("Accuracy", "Accuracy")
#Graph comparison of key effects for all three experiments- Similar to forest plot in sig
#filtering paper?
forest.colors <- c("black", "red")

ts.acc.pes.int$Exp <- as.factor(ts.acc.pes.int$Exp)

str(ts.acc.pes.int)

## 'data.frame': 2 obs. of 5 variables:
## $ Exp : Factor w/ 2 levels "Current","Original": 1 2
## $ Effect.size: num 0.0145 0.1604
## $ CI.Lower : num 0 0.011
## $ CI.Upper : num 0.107 0.347
## $ name : chr "Accuracy" "Accuracy"

forest.comp <- mod.forestplot(df = ts.acc.pes.int,
                             estimate = Effect.size,
                             ci.lower = CI.Lower,
                             ci.upper = CI.Upper,
                             colour = Exp,
                             xlab = "Partial Eta-Squared"
) +
  scale_color_manual(values = forest.colors)

## Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the
## existing scale.

forest.comp

```

