Domenech_Analysis

Charlie Wilson

June 2021

Setup and packages

summarySE Function for plot DFs

Setup and count and order data Then extract it if needed Skip to next chunk if already saved

```r
monkey <- dlgInput("Enter monkey", Sys.info()["user"])$res

# Where are the data?
##For Charlie Macbook
path <- file.path("/Users/charliewilson/Dropbox/Reversible Cognition/Domenech_Analysis/",monkey); #Wher
save.path <- file.path("/Users/charliewilson/Dropbox/Reversible Cognition/Domenech_Analysis"); #Path to

#List and organise the data
Session.list <- dir(path = path, pattern = "*.txt"); #List the files in the date
Dates <- data.frame(Session.list); #Extract dates from filenames

colnames(Dates) <- c("fid"); #rename File ID column
Dates <- Dates %>% separate(fid, c("Monkey", "Date","_","Time","Ext"),sep=cumsum(c(2,6,1,4,4))); #Split
Dates$Date <- dmy(Dates$Date); #Put Date column into date format
Dates$fid <- Session.list;
Dates <- arrange(Dates, Date); #Put Dates into date order

Nsession <- length(Dates$Date); # How many sessions?

Data <- data.frame(); #Initialise big data frame

if(Nsession >0){ #Sanity check

    for(isession in 1:Nsession){ #Loop on each session

      fid <- Dates$fid[isession]; #Extract filename from Dates
      print(fid) #Show filename
      event.raw <- read.csv(file=paste(path,"/",fid, sep= ""),sep= "", dec =",",  header = FALSE, fill=
      colnames(event.raw) <- c("time","event"); #Name columns
      event.raw$session <- isession; #Record session number

      # Now some basic analysis
      event.raw$trial <- cumsum(event.raw$event == 100); #Trial number

      # Which stimulus was used in the trial
      event.raw <- event.raw %>%
        group_by(trial) %>%   #Apply this trial by trial (rather than line by line)
        dplyr:: mutate(Stim = nth(event,4)) %>%  #In every trial take teh nth entry in the Event column
```

```
      ungroup();   #Put it back with the new column

    # Which TS
    event.raw <- event.raw %>%
      group_by(trial) %>%
      dplyr:: mutate(TS = nth(event,2)) %>%
      ungroup();

    # This is the end of intiial analysis of a single day

    #Put this session (event.raw) into Data which will contain all of the sessions
    Data <- rbind(Data,event.raw);

   rm(event.raw)

  }
 }

#Save
save(file = paste(save.path,"/Data_Domenech_",monkey,".Rdata",sep = ""),Data, Dates) # Save monkey spec
```

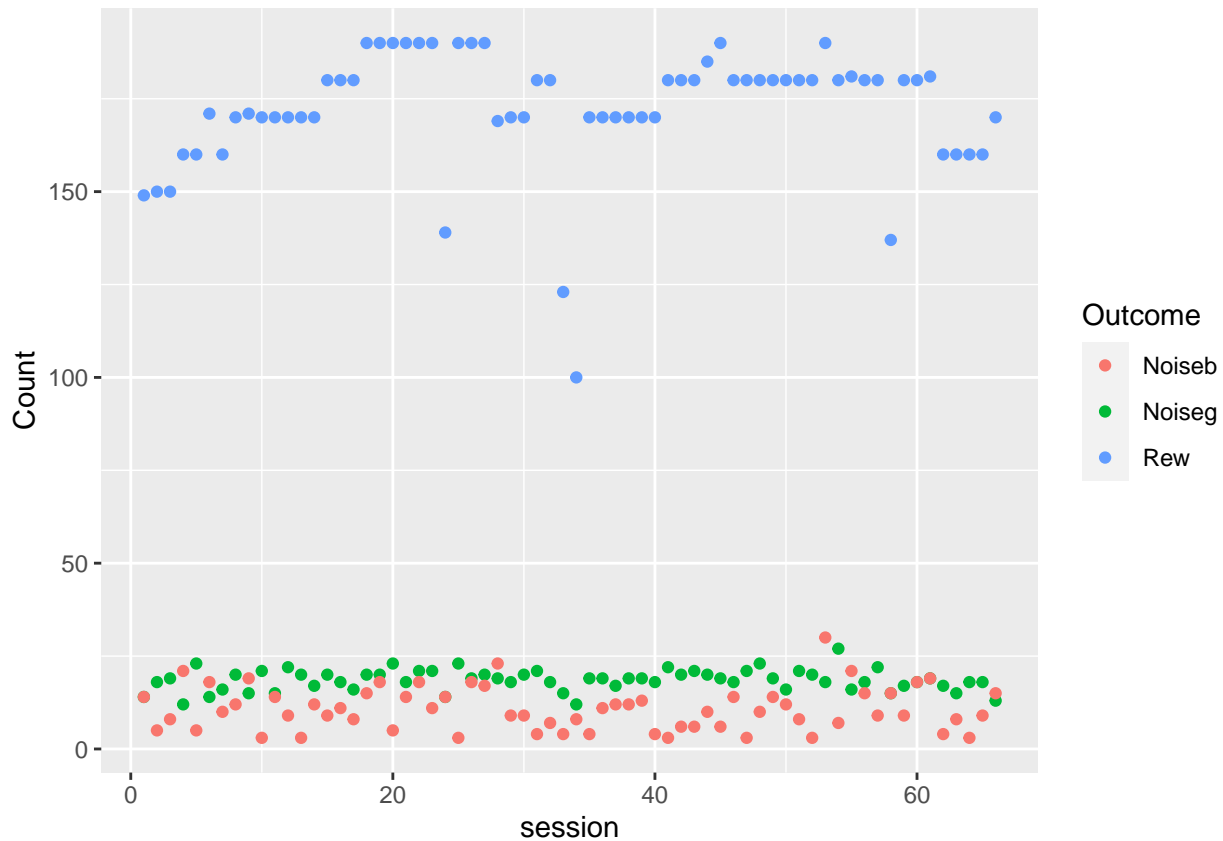If you don't need to run all of the above because you have already done it, you can load the data like this:

```
#monkey <- dlgInput("Enter monkey", Sys.info()["user"])$res
monkey <- "Dali"
# Where are the data?
save.path <- file.path("/Users/charliewilson/Dropbox/Brain Lyon/Reversible Cognition/Domenech_Analysis")

load(file = paste(save.path,"/Data_Domenech_",monkey,".Rdata",sep = ""))
```

What is in each session?

Let's build a trial by trial matrix to help the trial analysis, and populate it with teh necessary information on each trial and TS

```
# First some summary statistics from the whole trial
Data <- Data %>%
  dplyr::group_by(session,trial) %>%
  dplyr::mutate(GNorm = sum(event == 150), BNorm = sum(event == 151),GTrap = sum(event == 152), BTrap =
  dplyr::ungroup()

# And use that to extract if the response was correct or not
Data <- Data %>%
  dplyr::group_by(session,trial) %>%
  dplyr::mutate(COR = max(GNorm | GTrap), INC = max(BNorm | BTrap)) %>%
  dplyr::ungroup()

# Extract response
Data <- Data %>%
  dplyr::group_by(session,trial) %>%
  dplyr::mutate(Resp =
                  case_when(
                    sum(event == 61)>0 ~ 1,
                    sum(event == 62)>0 ~ 2,
                    sum(event == 63)>0 ~ 3)) %>%
  dplyr::ungroup()

# Now extract the first line of each trial
#rm ('Trials')
Trials <- Data %>%
```

3

```
  dplyr::group_by(session, trial) %>%
  slice(1)
Trials <- subset(Trials, COR>0 | INC>0) # Just trials with an outcome

# Extract the previous trial info for Stim and Rew
Trials <- Trials %>%
  dplyr::group_by(session) %>%
  dplyr::mutate(PrevSt=lag(Stim), PrevOut=lag(COR) ) %>%
  dplyr::ungroup()

# How many rewards in the last 3 trials?
Trials <- Trials %>%
  dplyr::group_by(session) %>%
  dplyr::mutate(roll_sum = roll_sum(Rew, 4, align = "right", fill = NA)) %>%
  dplyr::ungroup()

# We want a new DF with the outcomes around each trap
TrapI <- which(Trials$GTrap == 1 | Trials$BTrap == 1)

# Count trials since and to the Gtraps
Trials$GTrapCyc = cumsum(Trials$GTrap)
Trials <- Trials %>%
  group_by(GTrapCyc) %>%
  dplyr::mutate(GTrapC = row_number()-1,GTrapCback = row_number() - (n()+1))

# Subset down to trials just around Traps and set t around Traps
#rm ('Traps', 'TrapSE')
Traps <- subset(Trials, GTrapC <=2 | GTrapCback == -1)
Traps$t <-Traps$GTrapC
Traps$t[Traps$t>2] <- -1

# Is the response the same as the Trap?
Traps <- Traps %>%
  group_by(GTrapCyc) %>%
  dplyr::mutate(TrapR = dplyr::first(Resp), SameRT = as.integer(TrapR==Resp))

# Is the stimulus the same as the Trap?
Traps <- Traps %>%
  group_by(GTrapCyc) %>%
  dplyr::mutate(TrapS = dplyr::first(Stim), SameST = as.integer(TrapS==Stim))

TrapSSame <- subset(Traps, SameST == 1)
TrapSDif <- subset(Traps, SameST == 0)
```

Use this to produce a monkey version of the figures from PD's AI manuscript

```
# summarySE provides the standard deviation, standard error of the mean, and a (default 95%) confidence
# We need the summary for the different t values -1 to 2
# But for t=2 only we need to split on the basis of PrevOut
# The following is an ugly way of achieving this...
SS1 <- summarySE(TrapSSame, measurevar="SameRT", groupvars=c("t","PrevOut"))

##
## Attaching package: 'doBy'
```

```
## The following object is masked from 'package:dplyr':
##
##     order_by
```

```
SS1 <- subset(SS1, t==2 & !is.na(PrevOut),select=-PrevOut)
SS2 <- summarySE(TrapSSame, measurevar="SameRT", groupvars=c("t"))
SS2 <- subset(SS2, t<2 & t!=0)
SumSame <- rbind(SS2,SS1)

# Also for the Different stimulus...
SD1 <- summarySE(TrapSDif, measurevar="SameRT", groupvars=c("t","PrevOut"))
SD1 <- subset(SD1, t==2 & !is.na(PrevOut),select=-PrevOut)
SD2 <- summarySE(TrapSDif, measurevar="SameRT", groupvars=c("t"))
SD2 <- subset(SD2, t<2)
SumDif <- rbind(SD2,SD1)

# Reproduce figs 2a,b from the Domenech Manuscript
xxx <- c("t1-","t1+")

ggplot(SumSame, aes(x=t,y=SameRT)) + #What data to plot
    geom_point(size=3,color='red',fill='white')+
    geom_text(aes(label=ifelse(t>1,xxx,'')),hjust=2,vjust=-1) +
    geom_errorbar(aes(ymin=SameRT-se, ymax=SameRT+se), width=.1) +
    ylim(0,1) +
    geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
    geom_rect(aes(xmin=-0.1, xmax=0.1, ymin=0, ymax=1), color="grey", alpha=0.1) +
    ggtitle("Same Stimulus as 'Negative' Trap on t0") +
    theme(plot.title = element_text(hjust = 0.5)) +
    ylab("Proportion repetition same respons as 'Neg' Trap") +
    xlab("Trials since 'Neg' Trap")
```

Same Stimulus as 'Negative' Trap on t0
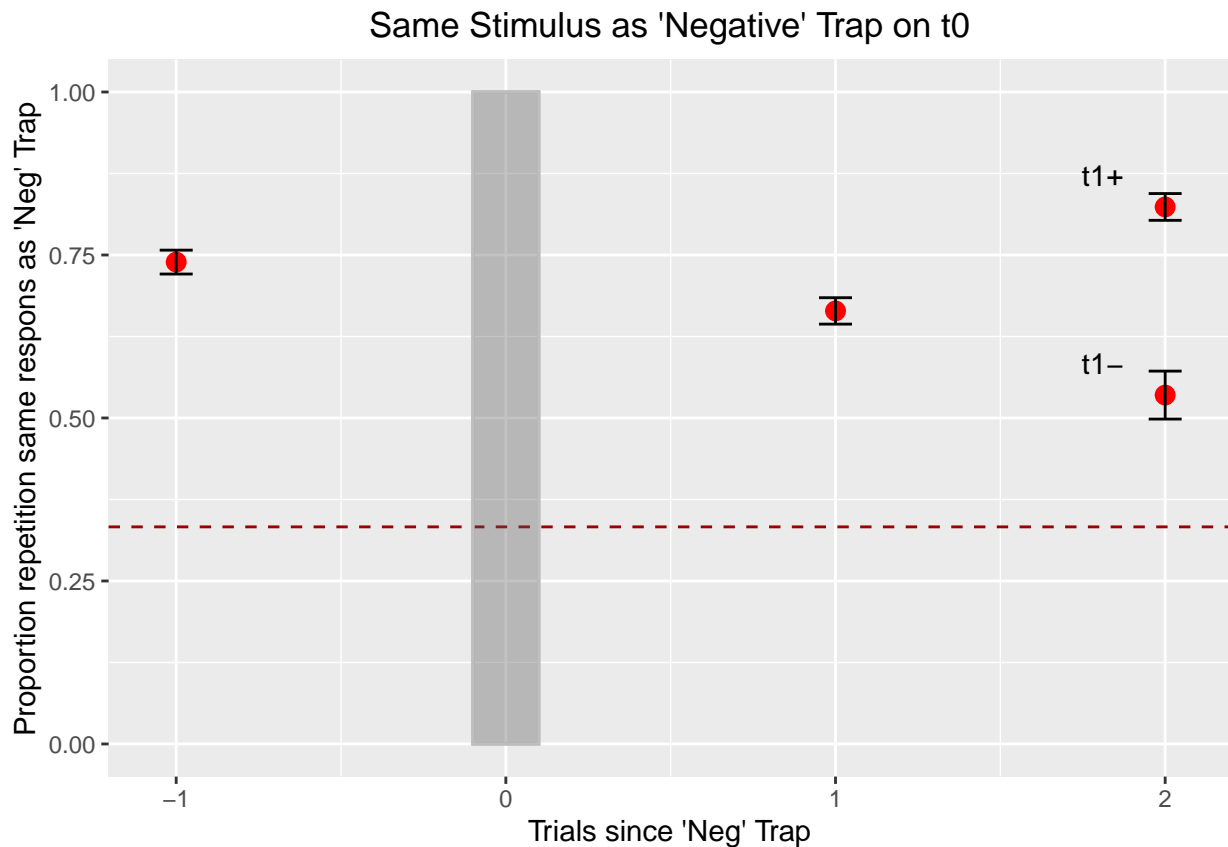

```r
ggplot(SumDif, aes(x=t,y=SameRT)) + #What data to plot
    geom_point(size=3,color='blue',fill='white')+
    geom_text(aes(label=ifelse(t>1,xxx,'')),hjust=2,vjust=-1) +
    geom_errorbar(aes(ymin=SameRT-se, ymax=SameRT+se), width=.1) +
    ylim(0,1) +
    geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
    geom_rect(aes(xmin=-0.1, xmax=0.1, ymin=0, ymax=1), color="grey", alpha=0.1) +
    ggtitle("Different Stimulus from 'Negative' Trap on t0") +
    theme(plot.title = element_text(hjust = 0.5)) +
    ylab("Proportion repetition same respons as 'Neg' Trap") +
    xlab("Trials since 'Neg' Trap")
```

Different Stimulus from 'Negative' Trap on t0
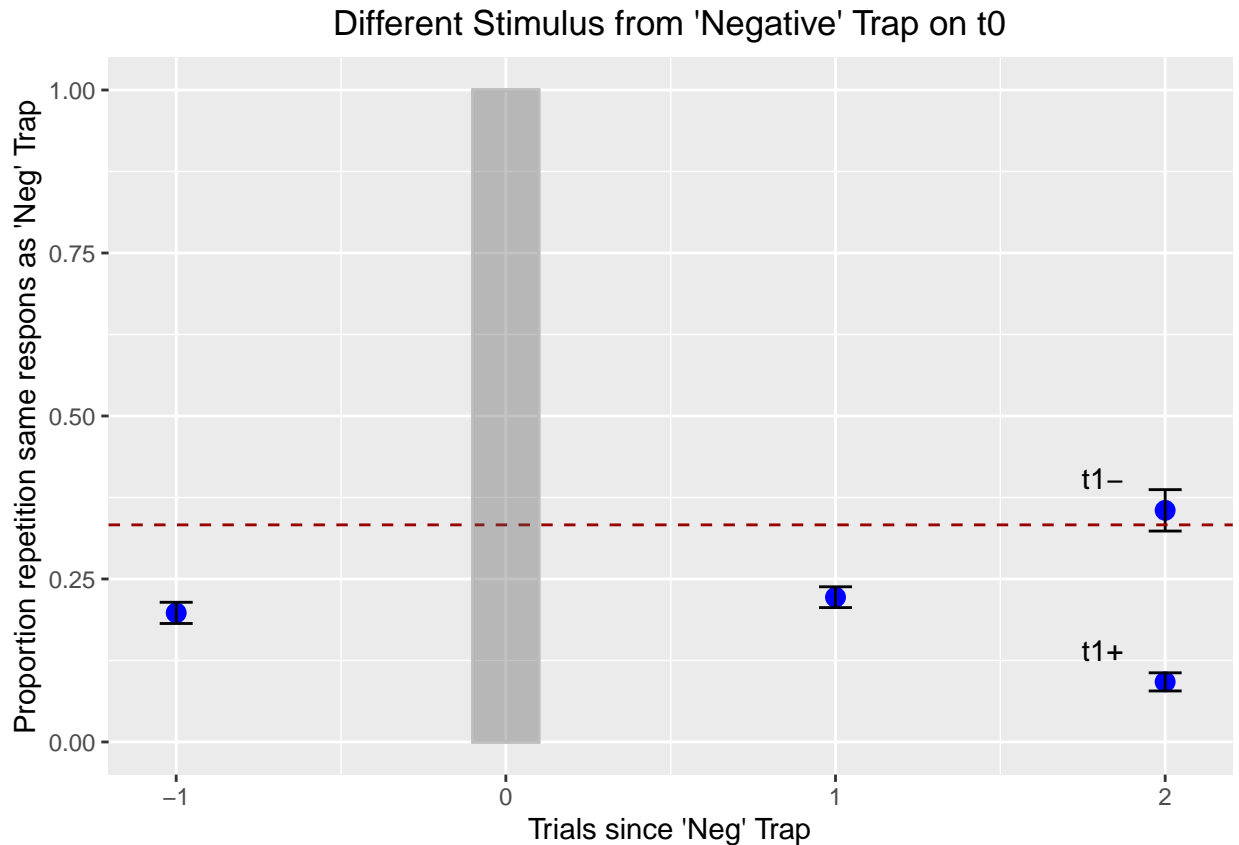

```
# Reproduce figs 2c,d from the Domenech Manuscript

t1same <- subset(TrapSSame,t==1)
Sumt1same <- summarySE(t1same, measurevar="SameRT", groupvars=c("roll_sum"))
t1dif <- subset(TrapSDif,t==1)
Sumt1dif <- summarySE(t1dif, measurevar="SameRT", groupvars=c("roll_sum"))

ggplot(Sumt1same, aes(x=roll_sum,y=SameRT)) + #What data to plot
  geom_point()+
  geom_line(color='red')+
  geom_errorbar(aes(ymin=SameRT-se, ymax=SameRT+se), width=.1) +
    ylim(0,1) +
    geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
    ggtitle("Same Stimulus as 'Negative' Trap on t0") +
    theme(plot.title = element_text(hjust = 0.5)) +
    ylab("Proportion repetition same response on t1 as 'Neg' Trap") +
    xlab("# Rewards in 3 trials before 'Neg' Trap")
```

Same Stimulus as 'Negative' Trap on t0
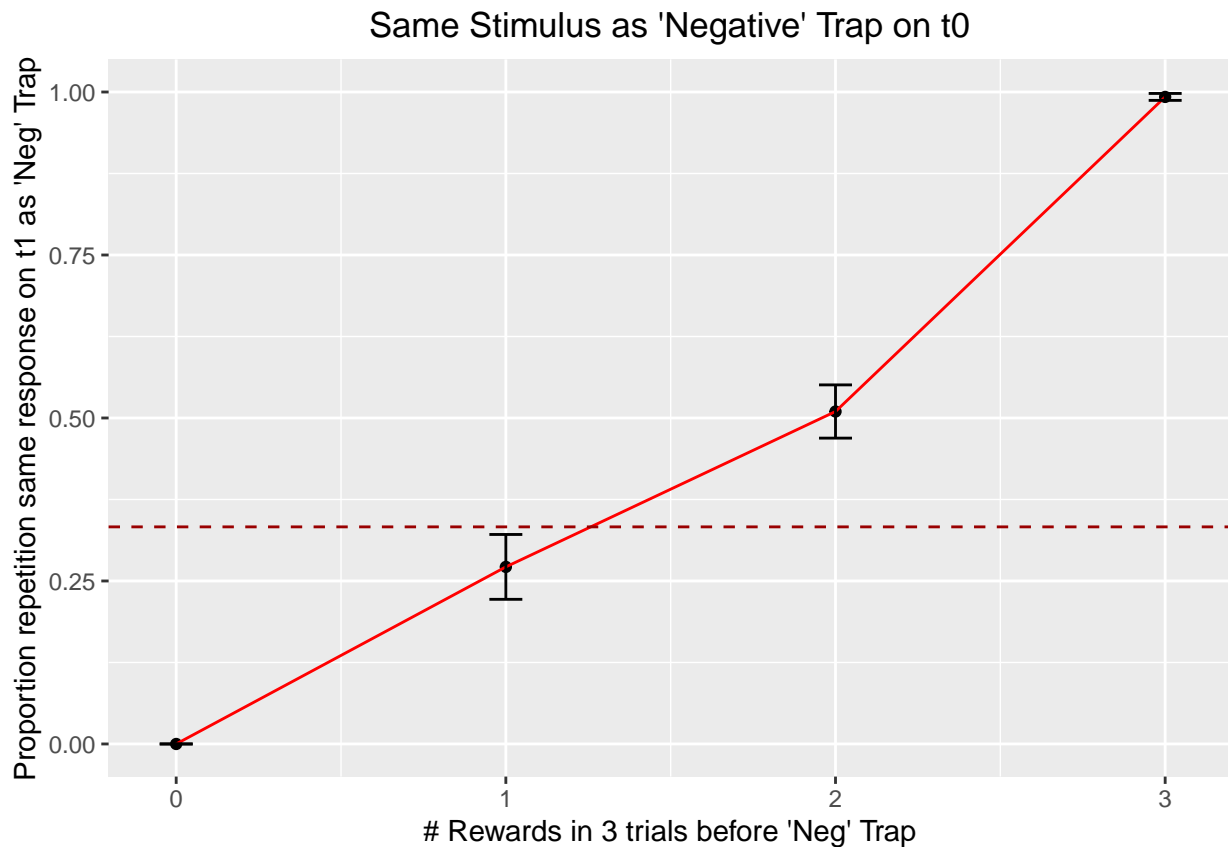

```
ggplot(Sumt1dif, aes(x=roll_sum,y=SameRT)) + #What data to plot
  geom_point()+
  geom_line(color='blue') +
  geom_errorbar(aes(ymin=SameRT-se, ymax=SameRT+se), width=.1) +
    ylim(0,1) +
    geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
    ggtitle("Different Stimulus from 'Negative' Trap on t0") +
    theme(plot.title = element_text(hjust = 0.5)) +
    ylab("Proportion repetition same response on t1 as 'Neg' Trap") +
    xlab("# Rewards in 3 trials before 'Neg' Trap")
```

Different Stimulus from 'Negative' Trap on t0
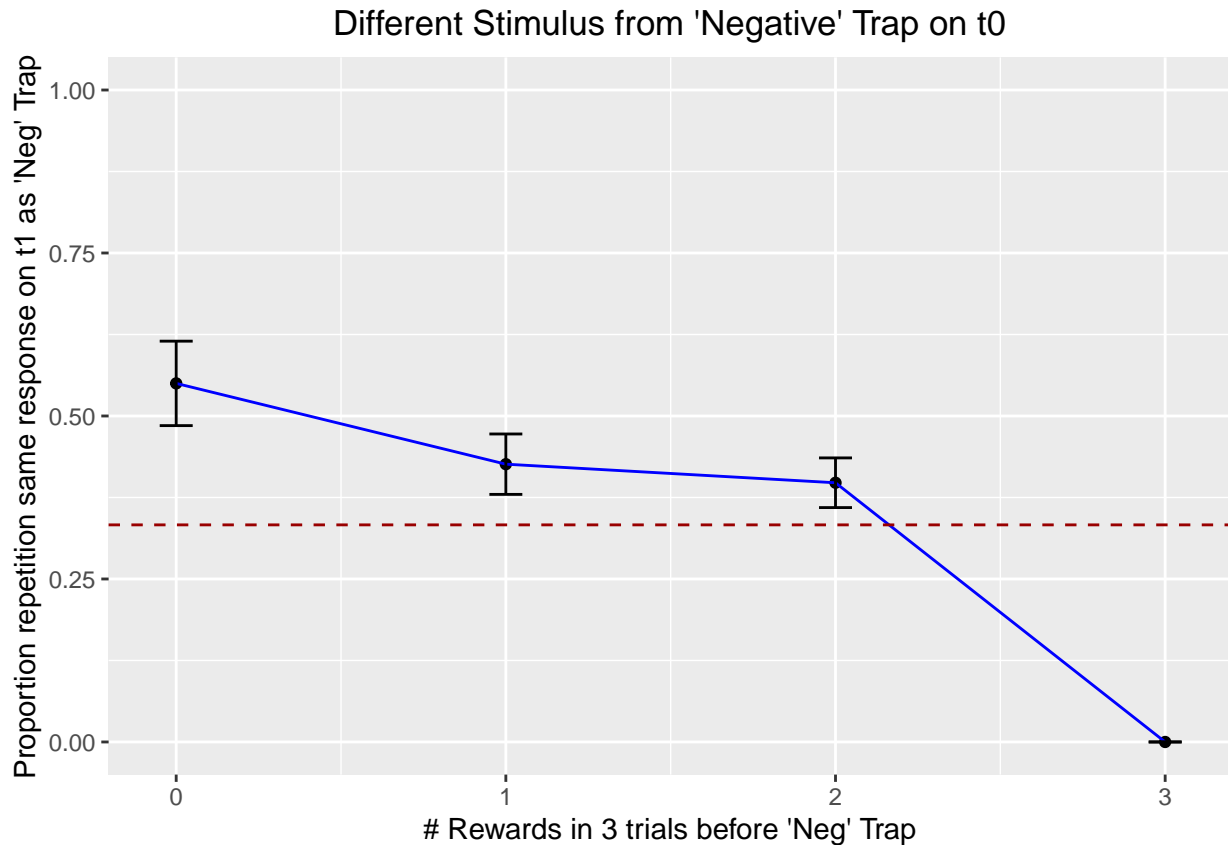
The monkeys in this later dataset look very much like the "good learner" humans in the PD manuscript, without the AI lesion This was unexpected for PD, who assumed from the Faraut et al paper that the Trap Reactivity of the monkeys shown therein was suggestive that they are doing a full reset on each Trap I am unconvinced that this was the case (their recovery was rapid), but in any case something might well have changed between the current dataset and the start of doing this Switch Task To test this we will need to compare the two datasets But first we want to look quickly at how the monkeys are learning in this dataset, and the reproduce the same figs as the Faraut et al study. Notably until now we have been looking at selection of same response after only GTraps (good resp, -ve FB). In Faraut we were more focussed on % COR

How are the monkeys learning in the TSs in this dataset?

```r
# Counter in each TS
Trials <- Trials %>%
  group_by(session, TS) %>%
  dplyr::mutate(TScount = row_number(), TSperc = mean(COR))
#Summarise % correct
TSsum <- summarySE(Trials, measurevar="COR", groupvars=c("TScount"))
#Plot % correct through TS
ggplot(TSsum, aes(x=TScount, y=COR)) +
  geom_point()+
  geom_line(color='blue') +
  geom_errorbar(aes(ymin=COR-se, ymax=COR+se), width=.1) +
  geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
  xlim(0,50)
```
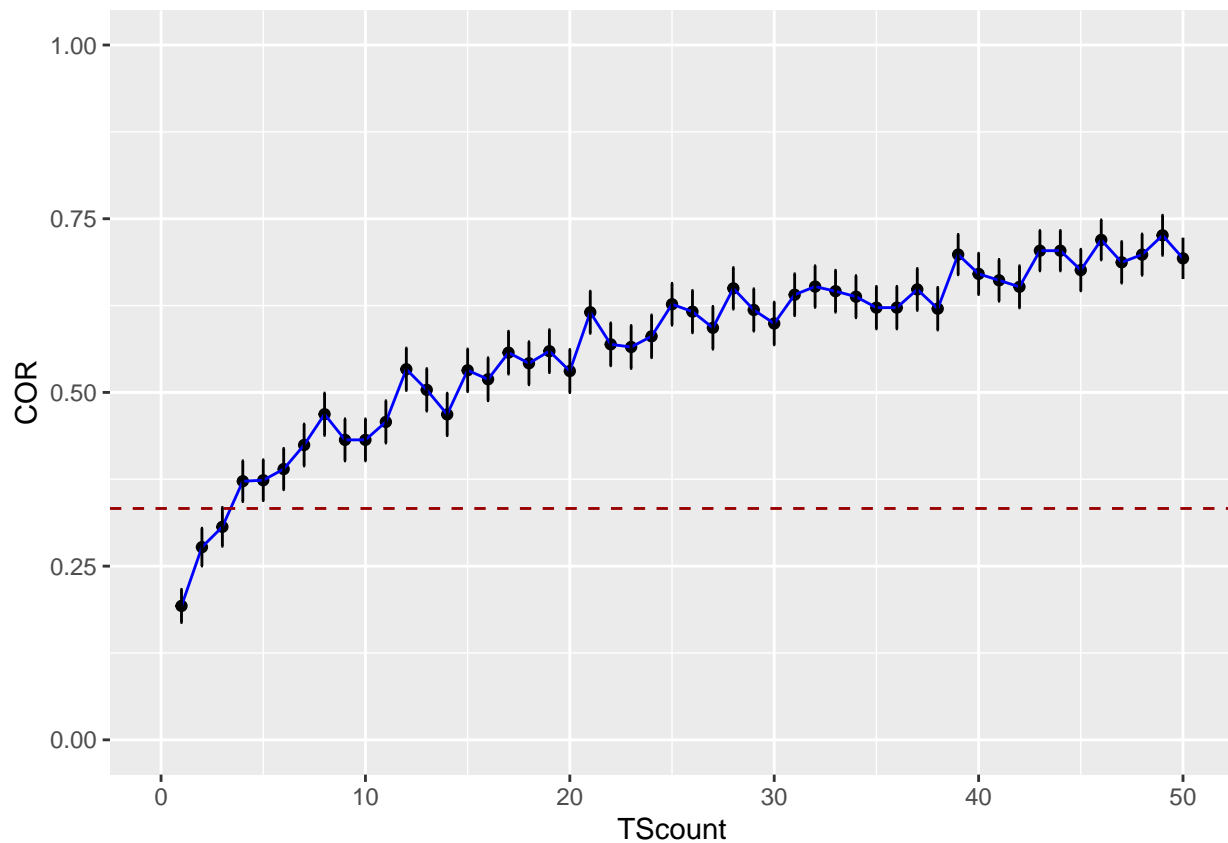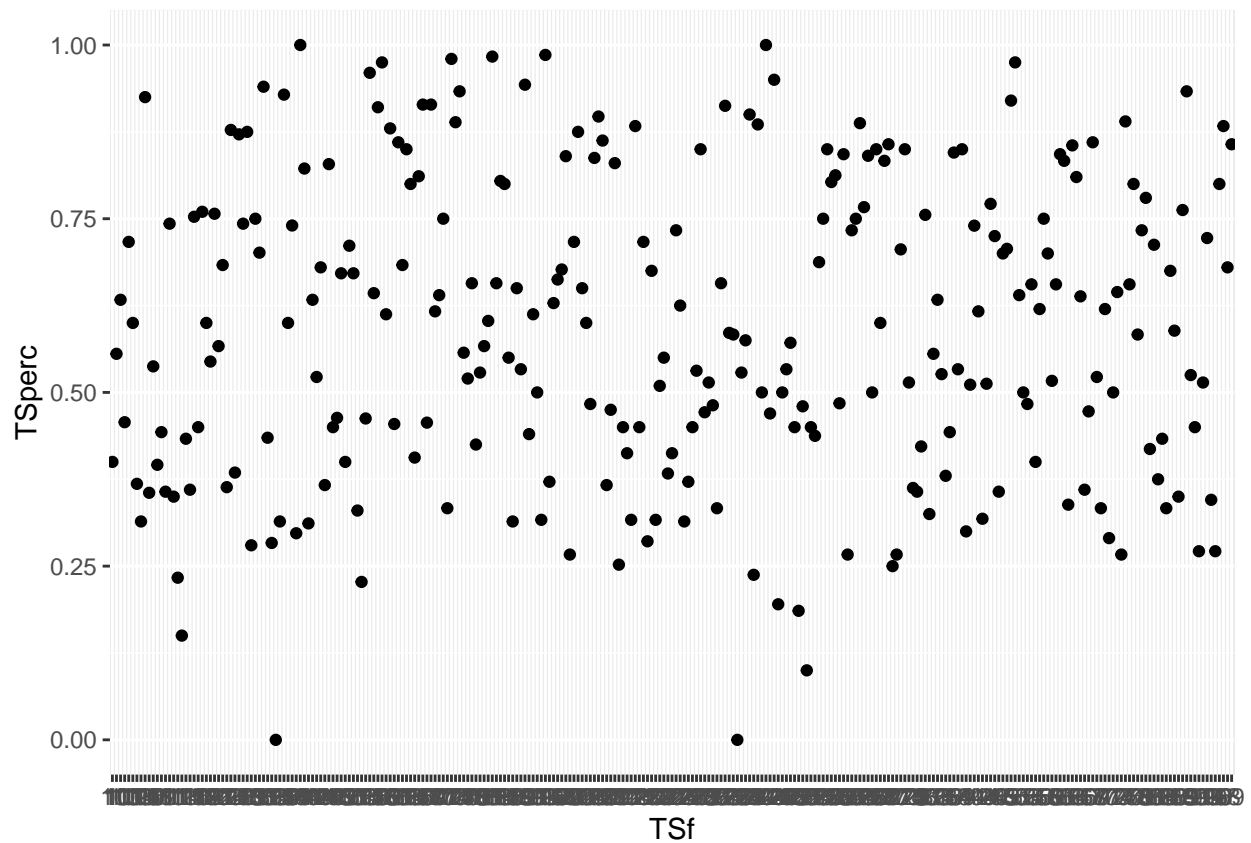
```
#How well are the TSs solved? Is there a pattern over time?
TSs <- Trials %>%
  dplyr::group_by(session, TS) %>%
  slice(1)
ggplot(TSs, aes(x=TSperc)) +
  geom_histogram()
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
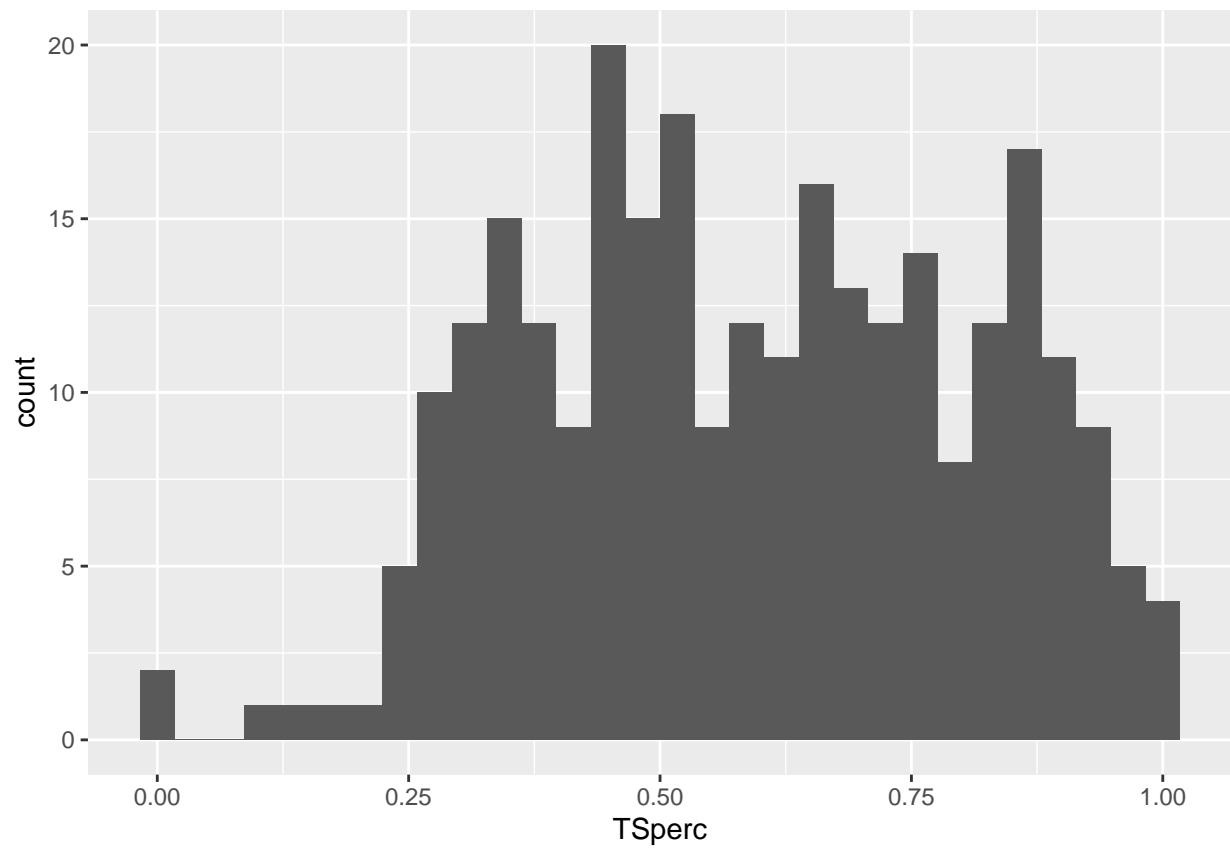
```
#Over time
TSs$TSf <- as.factor(rownames(TSs))
ggplot(TSs, aes(x=TSf, y=TSperc)) +
  geom_point()
```

```
#Look at only the good TSs
GoodTS <- subset(Trials, TSperc>0.75)
GoodTSsum <- summarySE(GoodTS, measurevar="COR", groupvars=c("TScount"))
ggplot(GoodTSsum, aes(x=TScount, y=COR)) +
  geom_point()+
  geom_line(color='green') +
  geom_errorbar(aes(ymin=COR-se, ymax=COR+se), width=.1) +
  geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
  xlim(0,50)
```
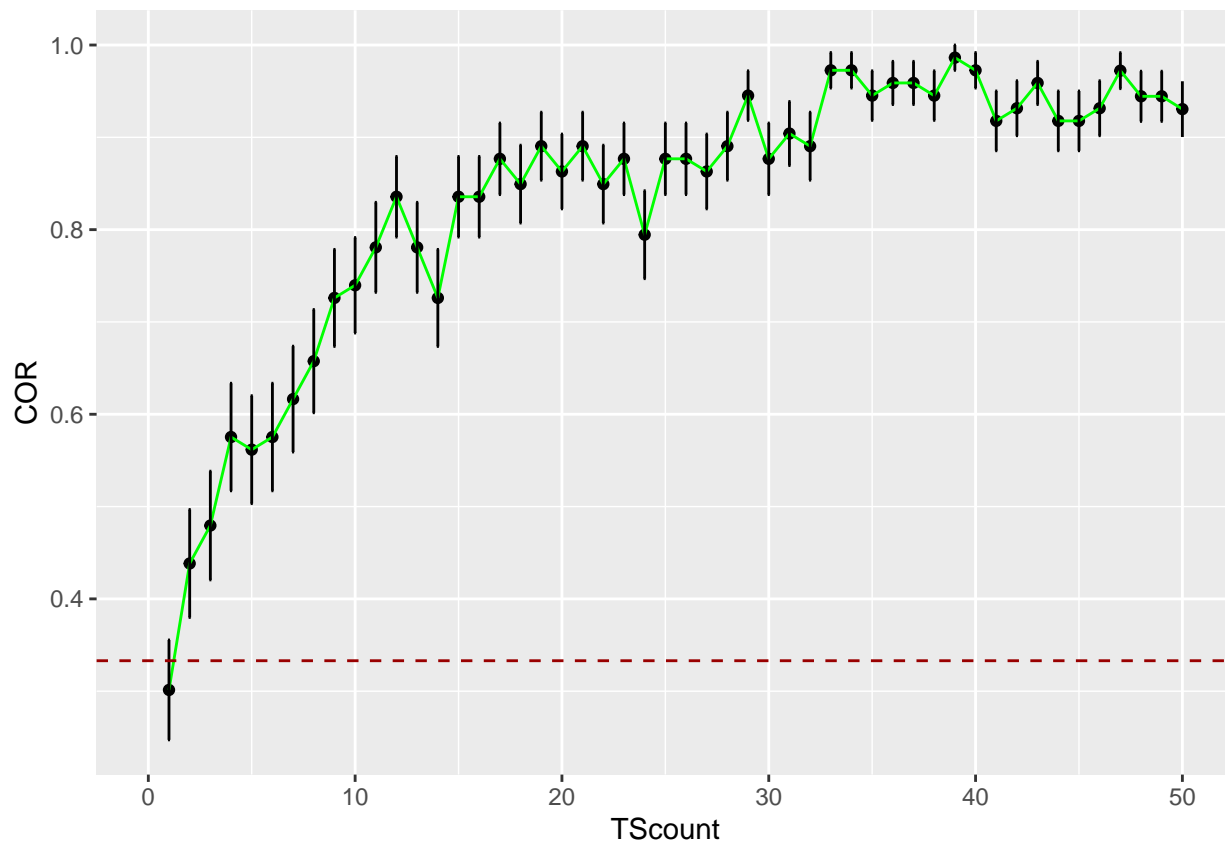
```
#Look at only the bad TSs
BadTS <- subset(Trials, TSperc<0.75)
BadTSsum <- summarySE(BadTS, measurevar="COR", groupvars=c("TScount"))
ggplot(BadTSsum, aes(x=TScount, y=COR)) +
  geom_point()+
  geom_line(color='red') +
  geom_errorbar(aes(ymin=COR-se, ymax=COR+se), width=.1) +
  geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
  xlim(0,50)
```
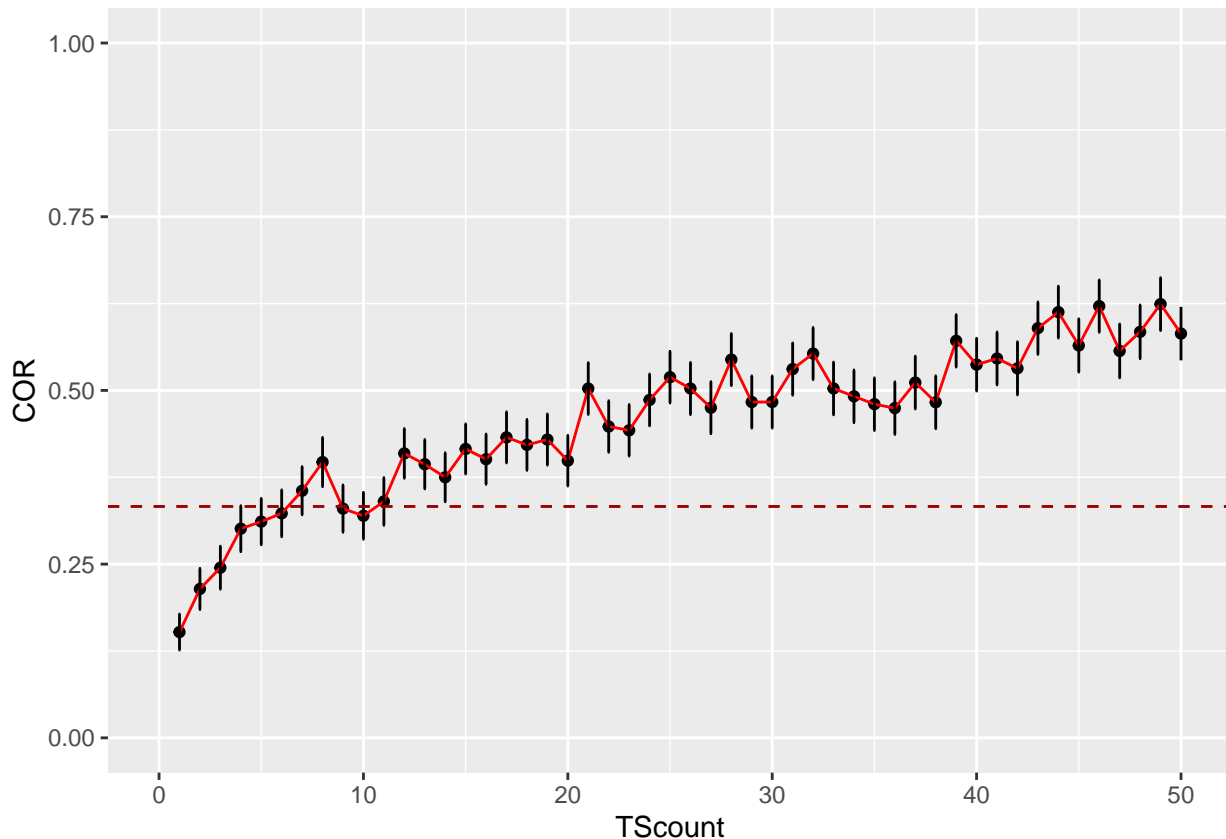
So learning is ok, but probably beneath what we had before Significant split between TS types. Clear and large number of TSs where the learning hit a 50% ceiling. They are probably only learning one of the stimuli

Now we want to reproduce some of the figures from Faraut et al 2016, but for the current dataset

```
#Let's look at trap reactivity as %COR (Fig 4A Faraut)

#Need to consider both Trap types, so need a Trap count that does that

# Count trials since and to the Gtraps

Trials$Traps = cumsum(Trials$GTrap+Trials$BTrap)
#max(Trials$Traps)
#sum(Trials$GTrap + Trials$BTrap)

Trials <- Trials %>%
  group_by(Traps) %>%
  dplyr::mutate(TrapC = row_number()-1,TrapCback = row_number() - (n()+1))

# Subset down to trials just around Traps and set t around Traps
#rm ('Traps', 'TrapSE')
AllTraps <- subset(Trials, TrapC <=4 | TrapCback >= -4)
AllTraps <- transform(AllTraps, tt = ifelse(TrapC>4,TrapCback,TrapC))

TrapReac <- summarySE(AllTraps, measurevar="COR", groupvars=c("tt"))

ggplot(TrapReac, aes(x=tt,y=COR)) + #What data to plot
    geom_point(size=3,color='red')+
    geom_line() +
```
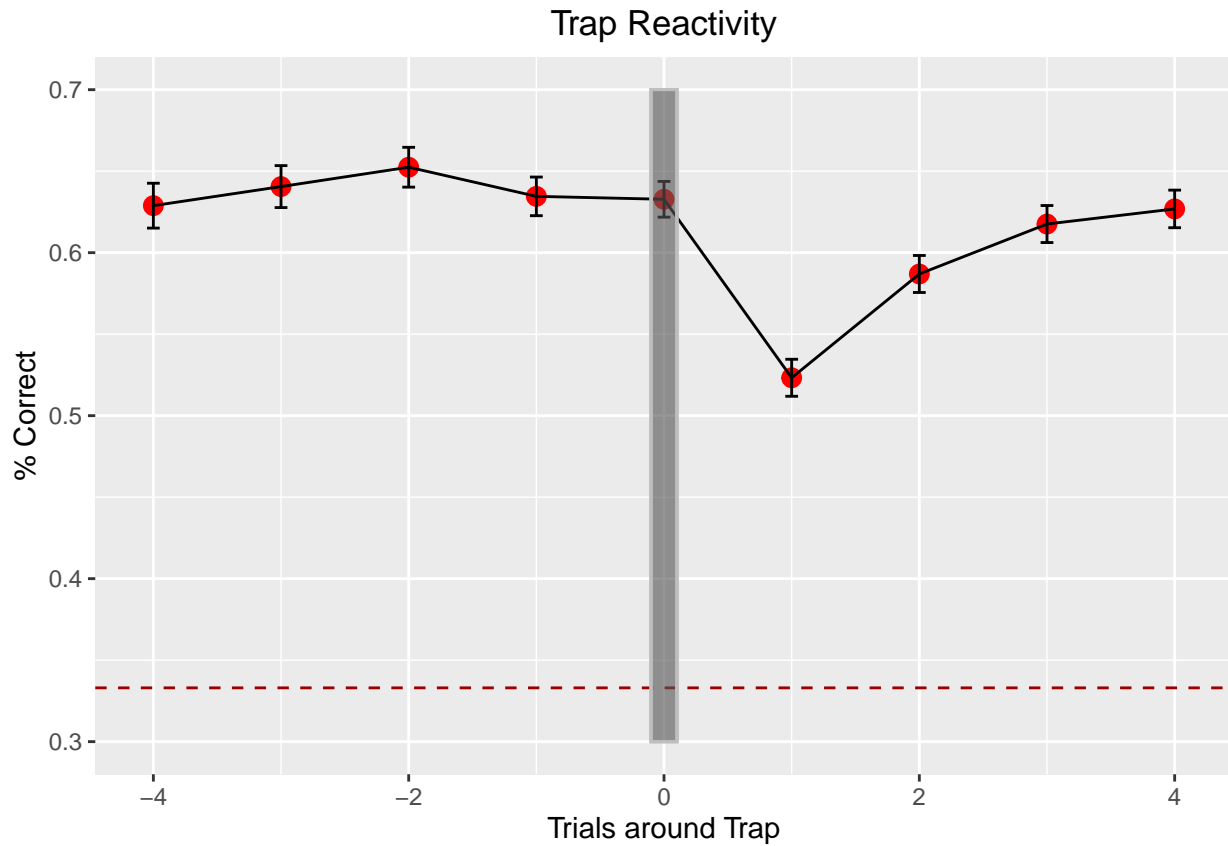
```
geom_errorbar(aes(ymin=COR-se, ymax=COR+se), width=.1) +
ylim(0.3,0.7) +
geom_hline(aes(yintercept=0.333), colour="#990000", linetype="dashed") +
geom_rect(aes(xmin=-0.1, xmax=0.1, ymin=0.3, ymax=0.7), color="grey", alpha=0.1) +
ggtitle("Trap Reactivity") +
theme(plot.title = element_text(hjust = 0.5)) +
ylab("% Correct") +
xlab("Trials around Trap")
```


Trap reactivity is clearly less than in the paper, on comparable pre-trap performance.