Determining Cognitive Load difficulty - Pilot Analysis

This is a short script that combines the pilot data files and saves a CSV that you can plot with.

Loading the Data

First, we want to add the files into one big data frame. For each participant we have two files. One ‘EndofTrial’ file that records counts, and one ‘WithinTrial’ file that records responses to stimuli.

```
library("tidyverse") #load code libraries for analysis

FilePath <- "~/SP_18-19/Pilot_Data/" #filepath (~ means to look in user's working directory)
Exp_ID <- "Orca18_Pilot_" #experiment code that gets prepended to any data file.
EoT <- "_EndofTrial.csv" #text at end of EndofTrial files
WiT <- "_WithinTrial.csv" #text at end of WithinTrial files
PPs <- 1:3 #vector of participant codes

for (pp in PPs){ #this for loop will loop through our participant vector
  filename_EoT = paste(FilePath,Exp_ID, pp, EoT, sep="") #creates a string from our variables
  filename_WiT = paste(FilePath,Exp_ID, pp, WiT, sep="") #creates a string from our variables

  if (pp == 1){ #if it is the first participant, create a new data frame.

    EoT_dataframe <- read.csv(filename_EoT) #load data
    WiT_dataframe <- read.csv(filename_WiT)

  } else{ #if it isn't the first participant, add loaded files to existing data frame
    EoT_newdata <- read.csv(filename_EoT) #load data
    WiT_newdata <- read.csv(filename_WiT)

    EoT_dataframe <- rbind(EoT_newdata, EoT_dataframe) #add to existing datframe
    WiT_dataframe <- rbind(WiT_newdata, WiT_dataframe) #add to existing datframe
  }
}

head(EoT_dataframe) #view start of dataframe.
```

```
##   X ppid targetoccurence targetnumber trialn EoTScore1 TargetCount1
## 1 0    3             0.8            2      0         5            5
## 2 1    3             0.2            3      1         0            0
## 3 2    3             0.2            1      2         5            5
## 4 3    3             0.6            2      3         5            4
## 5 4    3             0.2            1      4         3            3
## 6 5    3             0.2            3      5         1            1
##   EoTScore2 TargetCount2 EoTScore3 TargetCount3
## 1         7            7        NA           NA
## 2         0            0         0            0
## 3        NA           NA        NA           NA
## 4         2            2        NA           NA
## 5        NA           NA        NA           NA
## 6         1            1         0            0
```

```
head(WiT_dataframe) #view start of dataframe.
```

```
##   X ppid targetoccurence targetnumber trialn CurrentAudio         RT
## 1 0    3             0.8            2      0            o  1.0326844
## 2 1    3             0.8            2      0            b -1.0000000
## 3 2    3             0.8            2      0            b  0.7820289
## 4 3    3             0.8            2      0            b  0.5154195
## 5 4    3             0.8            2      0            h -1.0000000
## 6 5    3             0.8            2      0            b  0.5988033
##   ResponseCategory Target1 Target2 Target3
## 1                1       o       b
## 2                2       o       b
## 3                1       o       b
## 4                1       o       b
## 5                4       o       b
## 6                1       o       b
```

You can see that we have two data frames.

The first, **EoT_dataframe** has 12 columns. Each row is a trial (*trialn*) . The first column *X* is the data frame index so each row has a unique identifier. The next four columns (*ppid, targetoccurence, targetnumber, trialn*) contains all the information necessary for that particular trial. The last three rows (*EoTScore1, TargetCount1…EoTScore3, TargetCount3*) contains the user inputted scores, and the actual target counts for each target. NA is inputted to any cells of trials that had fewer targets than three.

The second dataframe, **WiT_dataframe**, also has 12 columns. Here each row is a stimulus presentation. The first five columns are the same as **EoT_dataframe**. *CurrentAudio* is the heard stimulus. *RT* is the response time (-1 if not responded). *ResponseCategory* is a code for how appropriate the response is: 1 = true positive, 2 = false negative, 3 = false positive, 4 = true negative. *Target1…Target3* indicates the actual targets for that trial.

Let's save these files so we don't have to load them again.

```
write.csv(EoT_dataframe, "Pilot_Data/Orca18_Pilot_EndofTrial_ALLPPs.csv")
write.csv(WiT_dataframe, "Pilot_Data/Orca18_Pilot_WithinTrial_ALLPPs.csv")
```

Calculating Measures

The dataframes now hold all the data needed for calculating measures for each trial.

The experiment design allows many ways to capture performance. There are three obvious ways of measures performance: the speed of response, whether the participant responded appropriately, and how far off they were in their recorded target counts.

From the **WiT_dataframe** the following measures need to be calculated: * MeanRT_TruePos (MeanRT for True Positives) * StdRT_TruePos (Standard deviation of RT for True Positives) * Perc_Correct (True Positives and True Negatives) * Perc_Incorrect(False Positives and False Negatives)

First, we want to remove any RTs that are unfeasibly quick, since they are probably responding to the previously head stimuli. Based on Luce (1986), this value is set at 100ms.

```
WiT_RTfiltered <- filter(WiT_dataframe, RT>.1) # Returns dataframe for rows where RT was >.1.
```

To calculate the RT measures, only the True Positives should be selected. In R, using the *tidyverse piping syntax* (https://style.tidyverse.org/pipes.html), we can use *group_by()* and *summarise()* to calculate the mean and standard deviation of a particular trial's RTs.

```
  WiT_TruePos <- filter(WiT_RTfiltered, ResponseCategory == 1) #create new dataframe only including tru

  SummaryRTs <- WiT_TruePos %>% group_by(ppid, trialn) %>% summarise(meanRT = mean(RT),
                                                                    stdRT = sd(RT))
```

```
head(SummaryRTs) #view start of dataframe. For every participant and trial there will be a meanRT and
```

```
## # A tibble: 6 x 4
## # Groups:   ppid [1]
##    ppid trialn meanRT   stdRT
##   <int>  <int>  <dbl>   <dbl>
## 1     1      0  0.916 NaN
## 2     1      1  0.977   0.119
## 3     1      2  0.901   0.266
## 4     1      3  0.880   0.213
## 5     1      5  1.01    0.268
## 6     1      6  0.852   0.188
```

All the response categories are needed calculating Perc_Correct and Perc_Incorrect, not just the True Positives.

```
#SummaryPerc <- WiT_RTfiltered %>% group_by(ppid, trialn) %>% summarise(meanRT = mean(RT),
                                                                      #stdRT = sd(RT))
```