Dual Task

Eileen Petros

7/12/2022

Constants

```r
# Enums
WALK <- 1
VSTROOP <- 2
ASTROOP <- 3

# All in seconds
TASK_DURATION <- 2
VSTROOP_INTERVAL <- 1
ASTROOP_INTERVAL <- 1
WALK_TRIALS <- 1
VSTROOP_TRAILS <- 1
ASTROOP_TRIALS <- 1

VSTROOP_STIMULI_COUNT <- TASK_DURATION / VSTROOP_INTERVAL
ASTROOP_STIMULI_COUNT <- TASK_DURATION / ASTROOP_INTERVAL

EVENT_COUNT <- WALK_TRIALS + VSTROOP_STIMULI_COUNT + ASTROOP_STIMULI_COUNT
# StartTrials, WalkTrial, VStroopTrial, AStroopTrial
# TODO change counts to 1, 6, 12, 12
# Depends on interstimulus duration
EVENT_A <- "# EVENT A - COUNT 1"
EVENT_B <- "# EVENT B - COUNT 1"
EVENT_C <- "# EVENT C - COUNT 2"
EVENT_D <- "# EVENT D - COUNT 2"

filePath <- "./Data/Test/Test0004.csv"
```

Data Import and Cleaning

```r
data <- read.csv(filePath)

data <- as_tibble(data)

# Remove separation lines
data <- data %>% filter(Time != "#")

# Extract summary rows and remove from data
summary <- data %>% slice_tail(n = EVENT_COUNT)
```

```r
data <- data %>% slice(1:(n() - EVENT_COUNT))
```

Detect Stomp

```r
# https://newbedev.com/r-tidyverse-how-to-change-column-data-type-using-pipes-with-least-typing
stompTime <- data %>% filter(FP1.ForY == max(FP1.ForY, na.rm = TRUE))
(stompTime$Time <- stompTime$Time %>% as.double())
```

```
## [1] 3.746667 3.750000 3.753333
```

Subset by Stimulus using D-Flow Event Breaks

```r
# Find event detection rows
eventBreaks <- data %>%  with(which(is.na(FP1.CopX)))
eventBreaks <- eventBreaks %>% sort()

# break into sets for each event
sets <- list()
# Ignore first grouping from recording data to
for (i in 2:(EVENT_COUNT)) {
  # Offset to exclude labels themselves
  start <- (eventBreaks[i] + 1)
  end <- (eventBreaks[i + 1] - 1)
  varName <- paste0("set", i - 1)
  assign(varName, data[start:end,])
  sets[[i - 1]] <- data[start:end,]
}

# Last set
start <- eventBreaks[length(eventBreaks)] + 1
assign(paste0("set", EVENT_COUNT),
       data[start :nrow(data),])
sets[[EVENT_COUNT]] <- data[start:nrow(data),]
```

Calculate Average Belt Speed for each Stimulus

```r
walkSpeeds <- c()
vStroopSpeeds <- c()
aStroopSpeeds <- c()

for (i in 1:EVENT_COUNT){
  currentSet <- as_tibble(sets[[i]])
  task <- first(currentSet$Task)

  # With character entries in time removed, convert col to double
  currentSet$Time <- currentSet$Time %>% as.double()
  sets[[i]] <- currentSet

  # Compute average speeds
  avgSpeed <- mean(currentSet$LeftBelt.Speed)
```

```r
  if(task == WALK){
    walkSpeeds <- c(walkSpeeds, avgSpeed)
  }
  else if (task == VSTROOP){
    vStroopSpeeds <- c(vStroopSpeeds, avgSpeed)
  }
  else if (task == ASTROOP){
    aStroopSpeeds <- c(aStroopSpeeds, avgSpeed)
  }
  else {
    print("Invlaid task type")
  }
}

walkSpeeds
```

[1] 0

```r
vStroopSpeeds
```

[1] 0 0

```r
aStroopSpeeds
```

[1] 0 0

Single Speed Value for Each Task Type

```r
(mean(walkSpeeds))
```

[1] 0

```r
(mean(vStroopSpeeds))
```

[1] 0

```r
(mean(aStroopSpeeds))
```

[1] 0