Small Multiples Full Scale Stimuli Generation

Helia Hosseinpour, Laura E. Matzen, Kristin M. Divis, Spencer C. Castro, and Lace Padilla

1/16/2022

Contents

Packages

The small multiples created in this document are our full scale versions, tested in Experiment 1. We simulated stock data (price over time) per company that we used to create small multiples with 18 different number of frames (two, six, ten, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 66, and 70). Each frame number had 20 versions; therefore, we created 360 small multiples in total. The context of our study revolved around energy grid operators. Therefore, we named the y-axis "Power (MW)" so that each small multiple resembled the trends of energy consumption per location (each frame would be a location) over time (one year per frame). All small multiples followed the same criteria but with different datasets. For our compare tasks, we used the same small multiples created here, but we added a blue highlight, using Adobe Illustrator, around the frames that we wanted the participants to compare. In Experiment 2, we tested a subset of these full scale small multiples (we used the same subset for fixed scale). The subset of the small multiples in Experiment 2 included one version for each of the five different numbers of frames tested (2, 6, 10, 30, and 70). The

simulated datasets and criteria we used to create our fixed scale small multiples for Experiment 3 (see our document named "Small Multiples Fixed Scale Stimuli Generation") were the same as those used here.

Note: Some datasets were modified by hand using Microsoft Excel so that they fit every task (e.g., getting the blue power line to reach above the dashed gray threshold line for the identify 3 task as not all generated plots naturally fitted this task)

Experiment 1

```
#Count the frames
frames <- seq(2, 70, by = 4)  # Generate a sequence from 2 to 70 by 4

# Repeat the values 20 times
repeated_values <- rep(frames, times = 20) #20 trials by the sum of frames

# Calculate the sum
sum_result <- sum(repeated_values) * 20 #20 participants
```

With 360 total trials (18 frame types by 20 trials) and the sum of individual frames (1.296×10^4) within 1 participant, the total number of observations is this value by 20 participants.

20 Participants * 648 individual frames per trial * 20 trials = 2.592×10^5 observations.

Identify 1

Instructions (Identify 1) Click on one graph with the highest peak power. Get Qualtrics output for 20 partcipants.

```
#Identify1 - remove unwanted columns
Identify1_LongV3 <- read.csv(file = "raw_data/Experiment1_Identify1_Raw.csv")[
    -c(1:8, 10:26) ] %>%
  dplyr::mutate(task = "Identify1") %>% #Task Id column
  tidyr::pivot_longer( #Pivot to long format for ID1 Trials and Frames
    cols = Look1_2GraphsFill1_1:Look1_70GraphsFill20_70, #Columns of all the
        Frames
    names_to = "trial_info", #Where the column names go
    values_to = "Response", #Where the values in thos columns go
    values_drop_na = TRUE)%>% #Drop NAs
    mutate(ResponseCode = as.numeric(ifelse(Response   == "On", "1", "0")))
      %>% #Make On and Off numeric (0,1)
  tidyr::separate(trial_info, into = c("task1","trialinfo2","Individual_
      FramesNum"), #names column split into 3
                  sep = "_",remove = FALSE) %>%  #split column to make
                      Individual_FramesNum
  mutate(Frame_Quantity = str_remove(trialinfo2, "Graphs.*"), #Create
      FrameQuantity
         Trial = str_remove(trialinfo2,'.*Fill'), #Create Trial
         Trial = str_remove(Trial, '\\.')) %>% #Remove extra periods from the
             end of trial
  select(-trialinfo2,-task1, -Gender_4_TEXT,-Educate_9_TEXT, #Remove extra
      columns
        -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>% #Put IFN
       and RC after Trial
```

```
  mutate(across(.cols = everything(),.fns = as.character)) #Make everything a
      character for joining purposes
```

Replace each data.frame (of 20) within the List of Lists for 2 frames (position 1) with filtered data.frame containing only company_name A and F.

TODO: Save all of the data.frames and lists needed into a single .RData file.

```
load(file = "stimuli/fullkey.RData") #Load the LIST OF LISTS (aka, the
    masterkey)
#Fix company name to uniform name scheme (in this case upper case letters)
listOlist_dats[[1]][[1]] <- data.frame(Date = listOlist_dats[[1]][[1]]$Date,
                                       my_vector = listOlist_dats[[1]][[1]]$my
                                           _vector,
                                       company_name = toupper(c(rep(letters
                                           [1],13),rep(letters[2],13),rep(
                                           letters[3],13),rep(letters[4],13),
                                                rep(letters[5],13),rep(
                                                    letters[6],13))))

#Get max values for each frame
frameMaxes <- lapply(listOlist_dats, function(x) lapply(x, function(y) tapply(
    y[,2],INDEX = y[,3],FUN = max)))
#Subset 2 frames from A,B,C,D,E,F to just A,F
frameMaxes[[1]] <-  lapply(frameMaxes[[1]], function(x) x[c("A", "F")])

#Rename trials from letters (eg A,F) to numbers
for (i in seq_along(frameMaxes)) { #Iterate through the list of 18 (1st level
    - frame#) in FrameMaxes
  frameMaxes[[i]] <- lapply(frameMaxes[[i]], function(x) { #Iterate through
      L20 (2nd level - trials)
    names(x) <- seq(1, length(x), 1) #Name each frame depending on frame#
    return(x)
  })
}

# Might come in handy later...
Maxes <- lapply(frameMaxes, function(x) lapply(x, max)) #Get the Max (ie
    winner) value for each trial

MaxPos <- lapply(frameMaxes, function(x) { #Get the Max (ie winner) FRAME for
    each trial
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(MaxPos) <- unique(Identify1_LongV3$Frame_Quantity) #Match the Frame
    Quantities to the answer key
```

DONE - Need to remove frames 2-5 from the frame 2 files (I forgot to do this) DONE - I also have max values of the winning frames, but not the winning frames themselves. You already did this calculation somewhere that returns the name of the frame (letter or number) that has the max value. Go find it.

Compare the maximum position to the response position to determine the accuracy. First, match the names

in the list MaxPos to the names in the FrameQuantity column, i.e., 2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 66, 70.

```
C_ansr <- Identify1_LongV3 %>%  # Get only the frames that were selected by a
    participant.
  filter(ResponseCode == 1)

# create an empty data.frame to store the results
key_Identify_df <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Correct_FramesNum = integer(),
                  stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MaxPos)) {
  # loop through each element in the list
  for (j in seq_along(MaxPos[[i]])) {
    # extract the values and add them to the data.frame
   key_Identify_df <- rbind(key_Identify_df, data.frame(Frame_Quantity = names
       (MaxPos)[i],
                                  Trial = as.character(j),
                                  Correct_FramesNum = MaxPos[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}

dfjoined <- left_join(C_ansr, key_Identify_df, by = c("Frame_Quantity","Trial"
    ))

Exp1_ID1_df <- dfjoined

# save(Exp1_ID1_df, file = "clean_data/main.RData")
```

Identify 2 (Combined Task)

```
#Identify2
Identify2_FullScaleData <- read.csv(file = "raw_data/Experiment1_Identify2_Raw
    .csv")
#removing unneeded columns
Identify2_FullScaleDataV2 <- Identify2_FullScaleData[ -c(1:8, 10:26) ] %>%
  mutate(task = "Identify2")


Identify2_Long<- Identify2_FullScaleDataV2 %>%
  pivot_longer(
    cols = Look1_2GraphsFill1_1:Look1_70GraphsFill20_70,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    mutate(ResponseCode = as.numeric(ifelse(Response    == "On", "1", "0")))

#split column
Identify2_LongV2<- cSplit(Identify2_Long, "trial_info", "_", drop = FALSE)
```

```
#name new column
colnames(Identify2_LongV2)[21] <- "Individual_FramesNum"

#getting frame # by it's self
Identify2_LongV2$Frame_Quantity <- sub('Graphs.*', '', Identify2_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Identify2_LongV2$Trial <- sub('.*Fill', '', Identify2_LongV2$trial_info_2)
Identify2_LongV2$Trial <- sub('\\.', '', Identify2_LongV2$Trial)

#drop unnecessary columns
Identify2_LongV3 <- Identify2_LongV2 %>%
  mutate(ResponseCode = ifelse(ResponseId != "R_1NcPPeiEdbNKRSp"& Frame_
      Quantity == 62 & Trial == 2,
                               as.numeric(ifelse(Response    == "On", "0", "1")
                                   ),ResponseCode)) %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))

# This gets joined with the output of creating the keys based on meanrange,
    mean, or range
C_ansr <- Identify2_LongV3 %>%
  filter(ResponseCode == 1)
```

Compare the meanrange, range, and mean scores to the recorded responses. First, we have to make the key from the raw stimuli data into the three score approaches.

```
f <- function(x) mean(x)*(max(x)-min(x)) # Multiply mean and range for
    meanrange score
# Put mean range score in our list of lists
frameMeanRange <- lapply(listOlist_dats, function(x) lapply(x, function(y)
    tapply(y[,2],INDEX = y[,3],FUN = f)))
# Put mean score in a list of lists
frameMean <- lapply(listOlist_dats, function(x) lapply(x, function(y) tapply(y
    [,2],INDEX = y[,3],FUN = mean)))

f_range <- function(x) max(x)-min(x) #  Get Range
# Put range score in a list of lists
frameRange <- lapply(listOlist_dats, function(x) lapply(x, function(y) tapply(
    y[,2],INDEX = y[,3],FUN = f_range)))

frameMeanRange[[1]] <-  lapply(frameMeanRange[[1]], function(x) x[c("A", "F")
    ]) #change 2 frames from 6 to 2

frameMean[[1]] <-  lapply(frameMean[[1]], function(x) x[c("A", "F")]) #change
    2 frames from 6 to 2

frameRange[[1]] <-  lapply(frameRange[[1]], function(x) x[c("A", "F")]) #
    change 2 frames from 6 to 2
```

Identify 2 MeanRange Processing

```
# MEAN RANGE
```

```
for (i in seq_along(frameMeanRange)) {
  frameMeanRange[[i]] <- lapply(frameMeanRange[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}


MeanRange <- lapply(frameMeanRange, function(x) lapply(x, max))

MeanRangePos <- lapply(frameMeanRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(MeanRangePos) <- unique(Identify2_LongV3$Frame_Quantity)


# create an empty data.frame to store the results
key_Identify_df <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanRangePos)) {

  # loop through each element in the list
  for (j in seq_along(MeanRangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df <- rbind(key_Identify_df, data.frame(Frame_Quantity = names
       (MeanRangePos)[i],
                                Trial = as.character(j),
                                Correct_FramesNum =MeanRangePos[[i]][[j]],
                                stringsAsFactors = FALSE))
  }
}

dfjoined <- left_join(C_ansr, key_Identify_df, by = c("Frame_Quantity","Trial"
    ))
Exp1_ID2_df <- dfjoined
# save(Exp1_ID1_df,Exp1_ID2_df, file = "clean_data/main.RData")
```

Identify 2 Secondary Data.Frames (Range vs Mean)

```
for (i in seq_along(frameMean)) {
  frameMean[[i]] <- lapply(frameMean[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}

Mean <- lapply(frameMean, function(x) lapply(x, max))

MeanPos <- lapply(frameMean, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(MeanPos) <- unique(Identify2_LongV3$Frame_Quantity)

# create an empty data.frame to store the results
key_Identify_df_mean <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanPos)) {

  # loop through each element in the list
  for (j in seq_along(MeanPos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_mean <- rbind(key_Identify_df_mean, data.frame(Frame_
      Quantity = names(MeanPos)[i],
                                Trial = as.character(j),
                                Correct_FramesNum =MeanPos[[i]][[j]],
                                stringsAsFactors = FALSE))
  }
}

dfjoinedMean <- left_join(C_ansr, key_Identify_df_mean, by = c("Frame_Quantity
    ","Trial"))



# RANGE
```

```
for (i in seq_along(frameRange)) {
  frameRange[[i]] <- lapply(frameRange[[i]], function(x) {
```

```
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}


Range <- lapply(frameRange, function(x) lapply(x, max))

RangePos <- lapply(frameRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})


names(RangePos) <- unique(Identify2_LongV3$Frame_Quantity)

# create an empty data.frame to store the results
key_Identify_df_range <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(RangePos)) {

  # loop through each element in the list
  for (j in seq_along(RangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_range <- rbind(key_Identify_df_range, data.frame(Frame_
       Quantity = names(RangePos)[i],
                                Trial = as.character(j),
                                Correct_FramesNum =RangePos[[i]][[j]],
                                stringsAsFactors = FALSE))
  }
}

dfjoinedRange <- left_join(C_ansr, key_Identify_df_range, by = c("Frame_
    Quantity","Trial"))

# save(dfjoinedMean,dfjoinedRange,file = "clean_data/ID2Analysis.RData")
```

Identify 3

```
#Identify3
Identify3_FullScaleData <- read.csv(file = "raw_data/Experiment1_Identify3_Raw
    .csv")
#removing unneeded columns
```

```
Identify3_FullScaleDataV2 <- Identify3_FullScaleData[ -c(1:8, 10:26) ] %>%
  mutate(task = "Identify3")

Identify3_Long<- Identify3_FullScaleDataV2 %>%
  pivot_longer(
    cols = Look1_2GraphsFill1_1:Look1_70GraphsFill20_70,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    mutate(ResponseCode = as.numeric(ifelse(Response    == "On", "1", "0")))

#split column
Identify3_LongV2<- cSplit(Identify3_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Identify3_LongV2)[21] <- "Individual_FramesNum"

#getting frame # by it's self
Identify3_LongV2$Frame_Quantity <- sub('Graphs.*', '', Identify3_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Identify3_LongV2$Trial <- sub('.*Fill', '', Identify3_LongV2$trial_info_2)
Identify3_LongV2$Trial <- sub('\\.', '', Identify3_LongV2$Trial)

#drop unnecessary columns
Identify3_LongV3 <- Identify3_LongV2 %>%
  # mutate(ResponseCode = ifelse(ResponseId != "R_1NcPPeiEdbNKRSp"& Frame_
      Quantity == 62 & Trial == 2,
  #                              as.numeric(ifelse(Response    == "On", "0",
      "1")),ResponseCode)) %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))



hline <- read_csv("raw_data/Geom_hline.csv") %>%
  dplyr::rename("Trial" = Seed) %>%
  dplyr::mutate(across(c("Frame_Quantity","Trial"), .fns = as.character))

names(frameMaxes) <- unique(Identify2_LongV3$Frame_Quantity)

# create an empty data.frame to store the results
key_Identify_df_maxes <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Individual_FramesNum = integer(),
                  Correct_FramesNum = integer(),
                  stringsAsFactors = FALSE)


# loop through each list in the main list
for (i in seq_along(frameMaxes)) {
```

```
  # loop through each element in the list
  for (j in seq_along(frameMaxes[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_maxes <- rbind(key_Identify_df_maxes,
                                  data.frame(Frame_Quantity = names(frameMaxes
                                      )[i],
                                  Trial = as.character(j),
                                  Individual_FramesNum = names(frameMaxes[[i]][[j
                                      ]]),
                                  Correct_FramesValue =frameMaxes[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}


# end Maxes ------------------------------------------------------------
dfjoinedMaxes1 <- left_join(key_Identify_df_maxes,hline, by = c("Frame_
    Quantity","Trial"))

dfjoinedMaxes <- left_join(Identify3_LongV3, dfjoinedMaxes1, by = c("Frame_
    Quantity","Trial","Individual_FramesNum"))

Exp1_ID3_df <- dfjoinedMaxes
# save(Exp1_ID1_df,Exp1_ID2_df,Exp1_ID3_df, file = "clean_data/main.RData")
```

Compare 1

(Compare 1) Of these two graphs highlighted in blue, click on one graph with the highest peak power. Reduce the key to only the two frame that were compared before the task.

```
comparelist <- list(c(1,2),c(1,6),c(1,7),c(2,9),c(6,13),
      c(2,10),c(12,21),c(16,24),c(24,33),
      c(18,28),c(22,31),c(21,31),c(1,12),
      c(4,15),c(33,44),c(48,59),c(44,56),c(54,66))

for(i in 1:length(frameMaxes)) {
frameMaxes[[i]] <- lapply(frameMaxes[[i]], function(x) x[c(comparelist[[i]])
    ])
}

for(i in 1:length(frameMaxes)) {
  for(j in 1:length(frameMaxes[[1]])){
names(frameMaxes[[i]][[j]]) <- c("1","2")
  }
}
Maxes <- lapply(frameMaxes, function(x) lapply(x, max))

MaxPos <- lapply(frameMaxes, function(x) {
```

```
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})


#Compare1
Compare1_FullScaleData <- read.csv(file = "raw_data/Experiment1_Compare1_Raw.
    csv")
Compare1_FullScaleDataV2 <- Compare1_FullScaleData[ -c(1:8, 10:26) ]

Compare1_Long<- Compare1_FullScaleDataV2 %>%
  pivot_longer(
    cols = Look1_2GraphsFill1_1:Look1_70GraphsFill20_2,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    mutate(ResponseCode = as.numeric(ifelse(Response    == "On", "1", "0")))

#split column
Compare1_LongV2<- cSplit(Compare1_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Compare1_LongV2)[20] <- "Individual_FramesNum"

#getting frame # by it's self
Compare1_LongV2$Frame_Quantity <- sub('Graphs.*', '', Compare1_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Compare1_LongV2$Trial <- sub('.*Fill', '', Compare1_LongV2$trial_info_2)
Compare1_LongV2$Trial <- sub('\\.', '', Compare1_LongV2$Trial)

#drop unnecessary columns
Compare1_LongV3 <- Compare1_LongV2 %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))



names(frameMaxes) <- unique(Compare1_LongV3$Frame_Quantity)

C_ansr <- Compare1_LongV3 %>%
  filter(ResponseCode == 1)

# create an empty data.frame to store the results
key_Identify_df_compare <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Correct_FramesNum = integer(),
                  stringsAsFactors = FALSE)
```

```
# loop through each list in the main list
for (i in seq_along(MaxPos)) {

  # loop through each element in the list
  for (j in seq_along(MaxPos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_compare <- rbind(key_Identify_df_compare, data.frame(Frame_
      Quantity = names(MaxPos)[i],
                                Trial = as.character(j),
                                Correct_FramesNum = MaxPos[[i]][[j]],
                                stringsAsFactors = FALSE))
  }
}

dfjoined <- left_join(C_ansr, key_Identify_df_compare, by = c("Frame_Quantity"
    ,"Trial"))

Exp1_C1_df <- dfjoined
# save(Exp1_ID1_df,Exp1_ID2_df,Exp1_ID3_df,Exp1_C1_df, file = "clean_data/main
    .RData")
```

Compare 2

(Compare 2) Of these two graphs highlighted in blue, click on one graph with both the biggest change and the highest average power.

```
#Compare2
Compare2_FullScaleData <- read.csv(file = "raw_data/Experiment1_Compare2_Raw.
    csv")
Compare2_FullScaleDataV2 <- Compare2_FullScaleData [ -c(1:8, 10:26) ]

Compare2_Long<- Compare2_FullScaleDataV2 %>%
  pivot_longer(
    cols = Look1_2GraphsFill1_1:Look1_70GraphsFill20_2,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    mutate(ResponseCode = as.numeric(ifelse(Response    == "On", "1", "0")))

#split column
Compare2_LongV2<- cSplit(Compare2_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Compare2_LongV2)[20] <- "Individual_FramesNum"

#getting frame # by it's self
Compare2_LongV2$Frame_Quantity <- sub('Graphs.*', '', Compare2_LongV2$trial_
    info_2)
```

```
#getting frame stimuli group by it's self
Compare2_LongV2$Trial <- sub('.*Fill', '', Compare2_LongV2$trial_info_2)
Compare2_LongV2$Trial <- sub('\\.', '', Compare2_LongV2$Trial)

#drop unnecessary columns
Compare2_LongV3 <- Compare2_LongV2 %>%
  # mutate(ResponseCode = ifelse(ResponseId != "R_1NcPPeiEdbNKRSp"& Frame_
      Quantity == 62 & Trial == 2,
  #                              as.numeric(ifelse(Response    == "On", "0",
      "1")),ResponseCode)) %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))

C_ansr <- Compare2_LongV3 %>%
  filter(ResponseCode == 1)
```

Make compare key Compare the meanrange, range, and mean scores to the recorded responses. First, we have to make the key from the raw stimuli data for the highlighted frames into the three score approaches.

```
comparelist <- list(c(1,2),c(1,6),c(1,7),c(2,9),c(6,13),
     c(2,10),c(12,21),c(16,24),c(24,33),
     c(18,28),c(22,31),c(21,31),c(1,12),
     c(4,15),c(33,44),c(48,59),c(44,56),c(55,66))

f <- function(x) mean(x)*(max(x)-min(x)) # Multiply mean and range for
    meanrange score
# Put mean range score in our list of lists
frameMeanRange <- lapply(listOlist_dats, function(x) lapply(x, function(y)
    tapply(y[,2],INDEX = y[,3],FUN = f)))
# Put mean score in a list of lists
frameMean <- lapply(listOlist_dats, function(x) lapply(x, function(y) tapply(y
    [,2],INDEX = y[,3],FUN = mean)))

f_range <- function(x) max(x)-min(x) #  Get Range
# Put range score in a list of lists
frameRange <- lapply(listOlist_dats, function(x) lapply(x, function(y) tapply(
    y[,2],INDEX = y[,3],FUN = f_range)))

frameMeanRange[[1]] <-  lapply(frameMeanRange[[1]], function(x) x[c("A", "F")
    ]) #change 2 frames from 6 to 2

frameMean[[1]] <-  lapply(frameMean[[1]], function(x) x[c("A", "F")]) #change
    2 frames from 6 to 2

frameRange[[1]] <-  lapply(frameRange[[1]], function(x) x[c("A", "F")]) #
    change 2 frames from 6 to 2
```

Compare 2 MeanRange Processing

```
# MEAN RANGE
```

```
for (i in seq_along(frameMeanRange)) {
  frameMeanRange[[i]] <- lapply(frameMeanRange[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
  }

for(i in 1:length(frameMeanRange)) {
frameMeanRange[[i]] <-  lapply(frameMeanRange[[i]], function(x) x[c(
    comparelist[[i]])])
}

for(i in 1:length(frameMeanRange)) {
  for(j in 1:length(frameMeanRange[[1]])){
     names(frameMeanRange[[i]][[j]]) <- c("1","2")
  }
}


MeanRange <- lapply(frameMeanRange, function(x) lapply(x, max))

MeanRangePos <- lapply(frameMeanRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(MeanRangePos) <- unique(Identify2_LongV3$Frame_Quantity)


# create an empty data.frame to store the results
key_Identify_df <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanRangePos)) {

  # loop through each element in the list
  for (j in seq_along(MeanRangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df <- rbind(key_Identify_df, data.frame(Frame_Quantity = names
      (MeanRangePos)[i],
                                Trial = as.character(j),
                                Correct_FramesNum =MeanRangePos[[i]][[j]],
                                stringsAsFactors = FALSE))
```

```
  }
}


dfjoined <- left_join(C_ansr, key_Identify_df, by = c("Frame_Quantity","Trial"
    ))

Exp1_C2_df <- dfjoined
# save(Exp1_ID1_df,Exp1_ID2_df,Exp1_ID3_df,Exp1_C1_df,Exp1_C2_df, file = "
    clean_data/main.RData")
```

Compare 2 Secondary Data.Frames (Range vs Mean)

```
for (i in seq_along(frameMean)) {
  frameMean[[i]] <- lapply(frameMean[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
  }

for(i in 1:length(frameMean)) {
frameMean[[i]] <-  lapply(frameMean[[i]], function(x) x[c(comparelist[[i]])])
}

for(i in 1:length(frameMean)) {
  for(j in 1:length(frameMean[[1]])){
     names(frameMean[[i]][[j]]) <- c("1","2")
  }
}

Mean <- lapply(frameMean, function(x) lapply(x, max))

MeanPos <- lapply(frameMean, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})


names(MeanPos) <- unique(Identify2_LongV3$Frame_Quantity)


# create an empty data.frame to store the results
key_Identify_df_mean <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Correct_FramesNum = integer(),
                  stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanPos)) {

  # loop through each element in the list
```

```
  for (j in seq_along(MeanPos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_mean <- rbind(key_Identify_df_mean, data.frame(Frame_
       Quantity = names(MeanPos)[i],
                                  Trial = as.character(j),
                                  Correct_FramesNum =MeanPos[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}

dfjoinedMean <- left_join(C_ansr, key_Identify_df_mean, by = c("Frame_Quantity
    ","Trial"))



for (i in seq_along(frameRange)) {
  frameRange[[i]] <- lapply(frameRange[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
  }

for(i in 1:length(frameRange)) {
frameRange[[i]] <-  lapply(frameRange[[i]], function(x) x[c(comparelist[[i]])
    ])
}

for(i in 1:length(frameRange)) {
  for(j in 1:length(frameRange[[1]])){
     names(frameRange[[i]][[j]]) <- c("1","2")
  }
}


RangePos <- lapply(frameRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(RangePos) <- unique(Compare2_LongV3$Frame_Quantity)

# create an empty data.frame to store the results
key_Identify_df_range <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(RangePos)) {

  # loop through each element in the list
```

```
  for (j in seq_along(RangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_range <- rbind(key_Identify_df_range,
                                  data.frame(Frame_Quantity = names(RangePos),
                                     Trial = as.character(j),
                                     Correct_FramesNum =RangePos[[i]][[j]],
                                     stringsAsFactors = FALSE))
  }
}


dfjoinedRange <- left_join(C_ansr, key_Identify_df_range, by = c("Frame_
    Quantity","Trial"))

# save(dfjoinedMean,dfjoinedRange,file = "clean_data/C2Analysis.RData")
```

Summarize 1

(Summarize 1) Is the general trend in the graphs going up or down? (This was a multiple-choice question with choices Up and Down).

```
#Summarize1
Summarize1_FullScaleData <- read.csv(file = "raw_data/Experiment1_Summarize1_
    Words.csv")
Summarize1_FullScaleDataV2 <- Summarize1_FullScaleData [-c(1:2), -c(1:8,
    10:26) ]

Summarize1_Long<- Summarize1_FullScaleDataV2 %>%
  pivot_longer(
    cols = Ense1_2GraphsFill1:Ense1_70GraphsFill20,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    mutate(ResponseCode = as.numeric(ifelse(Response == "Up",1,0)))


#split column
Summarize1_LongV2<- cSplit(Summarize1_Long, "trial_info", "_", drop = FALSE)
#name new column
# colnames(Summarize1_LongV2)[18] <- "Individual_FramesNum"

#getting frame # by it's self
Summarize1_LongV2$Frame_Quantity <- sub('Graphs.*', '', Summarize1_LongV2$
    trial_info_2)


#getting frame stimuli group by it's self
```

```
Summarize1_LongV2$Trial <- sub('.*Fill', '', Summarize1_LongV2$trial_info_2)
Summarize1_LongV2$Trial <- sub('\\.', '', Summarize1_LongV2$Trial)

#drop unnecessary columns
Summarize1_LongV3 <- Summarize1_LongV2 %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  relocate(c(ResponseCode) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))



f <- function(x) last(x) - first(x)

framedifs <- lapply(listOlist_dats, function(x) lapply(x,  function(y) {
  ifelse(sum(tapply(y[,2], y[,3],f))>0,1,0)
  }))

names(framedifs) <- unique(Summarize1_LongV3$Frame_Quantity)



# create an empty data.frame to store the results
key_Identify_df_summ <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(framedifs)) {

  # loop through each element in the list
  for (j in seq_along(framedifs[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_summ <- rbind(key_Identify_df_summ, data.frame(Frame_
      Quantity = names(framedifs)[i],
                                 Trial = as.character(j),
                                 Correct_FramesNum =framedifs[[i]][[j]],
                                 stringsAsFactors = FALSE))
  }
}

# create an empty data.frame to store the results
key_Identify_df_range <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 MaxRange = integer(),
                 stringsAsFactors = FALSE)


dfjoined <- left_join(Summarize1_LongV3, key_Identify_df_summ, by = c("Frame_
    Quantity","Trial"))
```

```
Exp1_S1_df <- dfjoined
# save(Exp1_ID1_df, Exp1_ID2_df, Exp1_ID3_df, Exp1_C1_df, Exp1_C2_df, Exp1_S1_df,
    file = "clean_data/main.RData")
```

Summarize 2

```
#Summarize2
Summarize2_FullScaleData <- read.csv(file = "raw_data/Experiment1_Summarize2_
    Raw.csv")
Summarize2_FullScaleDataV2 <- Summarize2_FullScaleData [ -c(1:8, 10:26) ]

Summarize2_Long<- Summarize2_FullScaleDataV2 %>%
  pivot_longer(
    cols = Ense1_2GraphsFill1:Ense1_70GraphsFill20,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)

#split column
Summarize2_LongV2<- cSplit(Summarize2_Long, "trial_info", "_", drop = FALSE)
#name new column
# colnames(Summarize2_LongV2)[18] <- "Individual_FramesNum"

#getting frame # by it's self
Summarize2_LongV2$Frame_Quantity <- sub('Graphs.*', '', Summarize2_LongV2$
    trial_info_2)


#getting frame stimuli group by it's self
Summarize2_LongV2$Trial <- sub('.*Fill', '', Summarize2_LongV2$trial_info_2)
Summarize2_LongV2$Trial <- sub('\\.', '', Summarize2_LongV2$Trial)

#drop unnecessary columns
Summarize2_LongV3 <- Summarize2_LongV2 %>%
  select(-trial_info_1,-trial_info_2) %>%
  select(-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT, Response) %>%
  relocate(c(Response) , .after = Trial) %>%
  mutate(across(.cols = everything(),.fns = as.character))



for(i in 1:length(listOlist_dats[[1]])) {
listOlist_dats[[1]][[i]] <- listOlist_dats[[1]][[i]][listOlist_dats[[1]][[i]]$
    company_name =="A"|listOlist_dats[[1]][[i]]$company_name =="F",]
}
```

```
# MEAN
-------------------------------------------------------------------------------

frameMean <- lapply(listOlist_dats, function(x) lapply(x, function(y) mean(y
    [,2])))
names(frameMean) <- unique(Summarize2_LongV3$Frame_Quantity)

# create an empty data.frame to store the results
key_Identify_df_mean <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_Mean = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(frameMean)) {

  # loop through each element in the list
  for (j in seq_along(frameMean[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_mean <- rbind(key_Identify_df_mean, data.frame(Frame_
       Quantity = names(frameMean)[i],
                                 Trial = as.character(j),
                                 Correct_Mean =frameMean[[i]][[j]],
                                 stringsAsFactors = FALSE))
  }
}

# create an empty data.frame to store the results
key_Identify_df_range <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 MaxRange = integer(),
                 stringsAsFactors = FALSE)

Range <- lapply(frameRange, function(x) lapply(x, max))
names(Range) <- unique(Summarize2_LongV3$Frame_Quantity)


# loop through each list in the main list
for (i in seq_along(Range)) {

  # loop through each element in the list
  for (j in seq_along(Range[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_range <- rbind(key_Identify_df_range, data.frame(Frame_
       Quantity = names(Range)[i],
                                 Trial = as.character(j),
                                 MaxRange =Range[[i]][[j]],
                                 stringsAsFactors = FALSE))
  }
}
```

```
dfjoinedRange1 <- left_join(Summarize2_LongV3, key_Identify_df_mean, by = c("
    Frame_Quantity","Trial"))

dfjoined <- left_join(dfjoinedRange1, key_Identify_df_range, by = c("Frame_
    Quantity","Trial"))

Exp1_S2_df <- dfjoined
# save(Exp1_ID1_df, Exp1_ID2_df, Exp1_ID3_df, Exp1_C1_df, Exp1_C2_df, Exp1_S1_df,
    Exp1_S2_df, file = "clean_data/main.RData")
```

Experiment 2

Description: 90 participants per each (4) between subjects condition 360 total. The conditions of experiment 2 are 4 between subjects conditions:

Between: - Time Constraint vs None - Full scale vs Fixed Scale Within: - 7 Tasks: + ID1 + ID2 + ID3 + Comp1 + Comp2 + Sum1 + Sum2 - 2,6,10,30,70 frames - No Repeated Trials!

Identify 1

```
#New key ------
trialListId1 <- list(c("2"= "10"), c("6"= "13"), c("10"= "19"), c("30"= "17")
    , c("70"= "12"))
trialListId2 <- list(c("2"= "1" ), c("6"= "5"), c("10"= "17"), c("30"= "20"),
     c("70"= "13"))
trialListId3 <- list(c("2"= "18"), c("6"= "17"), c("10"= "19"), c("30"= "17")
    , c("70"= "3"))
trialListComp1 <-list(c("2"= "4" ), c("6"= "17"), c("10"= "19"), c("30"= "18")
    , c("70"= "19"))
trialListComp2 <-list(c("2"= "12"), c("6"= "3"), c("10"= "1"), c("30"= "2"), c
    ("70"= "1"))
trialListSumm1 <-list(c("2"= "20"), c("6"= "17"), c("10"= "16"), c("30"= "15")
    , c("70"= "8"))
trialListSumm2 <-list(c("2"= "11"), c("6"= "6"), c("10"= "6"), c("30"= "5"), c
    ("70"= "11"))


names(listOlist_dats) <-  seq(2,70,by=4)

for(i in 1:length(listOlist_dats)) {
  for(j in 1:length(listOlist_dats[[1]])) {
names(listOlist_dats[[i]]) <- seq(1,20,1)
  }
}

Id1key <- data.frame(Trial = unlist(trialListId1)[seq(1, length(trialListId1),
     by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")
```

```
exp2subsetId1 <- listOlist_dats[names(listOlist_dats) %in% Id1key$Frame_
    Quantity]


for(i in 1:length(exp2subsetId1)) {
  for(j in 1:length(exp2subsetId1[[i]])){
  exp2subsetId1[[i]][[j]] <- exp2subsetId1[[i]][[j]][j == Id1key$Trial[i]]
  }
 exp2subsetId1[[i]] <-  exp2subsetId1[[i]][lengths(exp2subsetId1[[i]]) != 0]
}

exp2subsetId1[[1]] <- lapply(exp2subsetId1[[1]], function(x) x[x$company_name
    %in% c("A", "F"), ])



frameMaxes <- lapply(exp2subsetId1, function(x) lapply(x, function(y) tapply(y
    [,2],INDEX = y[,3],FUN = max)))

for (i in seq_along(frameMaxes)) {
  frameMaxes[[i]] <- lapply(frameMaxes[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}


Maxes <- lapply(frameMaxes, function(x) lapply(x, max))


MaxPos <- lapply(frameMaxes, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})



Exp2Identify1_FixedScaleData_NT <- read_csv(file = "raw_data/Exp2/Matched+Size
    +Graphs_No+Time+Constraint_May+3,+2023_09.58.csv")[-c(1:2),] %>%
  mutate(scale = "Fixed", time_lim = "0")

Exp2Identify1_FixedScaleData_T <- read_csv(file = "raw_data/Exp2/Matched+Size+
    Graphs_Time+Constraint_May+3,+2023_10.00.csv")[-c(1:2),]%>%
  mutate(scale = "Fixed", time_lim = "1")

Exp2Identify1_FullScaleData_NT <- read_csv(file = "raw_data/Exp2/Full+Size+
    Graphs_No+Time+Constraint_May+3,+2023_09.51.csv")[-c(1:2),]%>%
  mutate(scale = "Full", time_lim = "0")

Exp2Identify1_FullScaleData_T <- read_csv(file = "raw_data/Exp2/Full+Size+
    Graphs_Time+Constraint_May+3,+2023_09.55.csv")[-c(1:2),]%>%
```

```
  mutate(scale = "Full", time_lim = "1")

dftemp <- rbind(rbind(rbind(Exp2Identify1_FixedScaleData_NT, Exp2Identify1_
    FixedScaleData_T),
                Exp2Identify1_FullScaleData_NT), Exp2Identify1_FullScaleData_T
                    )

#removing unneeded columns
Exp2FullV2 <- dftemp[, -c(1:8, 10:37) ] %>%
  dplyr::select(ResponseId,everything(),-contains("Click"),-contains("Submit")
      ,
                -contains("T_"),-contains("6Graphs2"))

Exp2Full_Long<- Exp2FullV2 %>%
  pivot_longer(
    cols = c(Look1_2Graphs10_1:Ense2_70Graphs11) ,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    dplyr::mutate(ResponseCode = as.numeric(ifelse(Response    == "On", "1", "0
        ")))

#split column
Exp2Full_LongV2<- cSplit(Exp2Full_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Exp2Full_LongV2)[colnames(Exp2Full_LongV2) =="trial_info_3"] <- "
    Individual_FramesNum"

#getting frame # by it's self
Exp2Full_LongV2$Frame_Quantity <- sub('Graphs.*', '', Exp2Full_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Exp2Full_LongV2$Trial <- sub('.*Graphs', '', Exp2Full_LongV2$trial_info_2)
Exp2Full_LongV2$Trial <- sub('\\.', '', Exp2Full_LongV2$Trial)

#drop unnecessary columns
Exp2Identify1_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
        -trial_info, -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything(),.fns = as.character)) %>%
  dplyr::filter(task == "Look1", ResponseCode == "1")



# create an empty data.frame to store the results
key_Identify_df <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  acc = integer(),
                  stringsAsFactors = FALSE)

# loop through each list in the main list
```

```
for (i in seq_along(MaxPos)) {

  # loop through each element in the list
  for (j in seq_along(MaxPos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df <- rbind(key_Identify_df, data.frame(Frame_Quantity = names
      (MaxPos)[i],
                                  Trial = names(MaxPos[[i]][j]),
                                  acc =MaxPos[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}



dfjoined <- left_join(Exp2Identify1_LongV3, key_Identify_df, by = c("Frame_
    Quantity","Trial"))

Exp2_ID1_df <- dfjoined
# save(Exp2_ID1_df, file = "clean_data/exp2main.RData")
```

Identify 2 (Combined Task)

```
#drop unnecessary columns
Exp2Identify2_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything() ,.fns = as.character)) %>%
  dplyr::filter(task == "Look2", ResponseCode == "1")

Id2key <- data.frame(Trial = unlist(trialListId2)[seq(1, length(trialListId1),
     by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetId2 <- listOlist_dats[names(listOlist_dats) %in% Id1key$Frame_
    Quantity]


for(i in 1:length(exp2subsetId2)) {
  for(j in 1:length(exp2subsetId2[[i]])){
  exp2subsetId2[[i]][[j]] <- exp2subsetId2[[i]][[j]][j == Id2key$Trial[i]]
  }
 exp2subsetId2[[i]] <-  exp2subsetId2[[i]][lengths(exp2subsetId2[[i]]) != 0]
}
```

```
exp2subsetId2[[1]] <- lapply(exp2subsetId2[[1]], function(x) x[x$company_name
    %in% c("A", "F"), ])


f <- function(x) mean(x)*(max(x)-min(x)) # Multiply mean and range for
    meanrange score
# Put mean range score in our list of lists
frameMeanRange <- lapply(exp2subsetId2, function(x) lapply(x, function(y)
    tapply(y[,2],INDEX = y[,3],FUN = f)))

frameMeanRange[[1]] <- lapply(frameMeanRange[[1]], function(x) x[c("A", "F")
    ]) #change 2 frames from 6 to 2


# MEAN RANGE
    ------------------------------------------------------------------------


for (i in seq_along(frameMeanRange)) {
  frameMeanRange[[i]] <- lapply(frameMeanRange[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}


MeanRange <- lapply(frameMeanRange, function(x) lapply(x, max))

MeanRangePos <- lapply(frameMeanRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

names(MeanRangePos) <- unique(Exp2Identify2_LongV3$Frame_Quantity)


# create an empty data.frame to store the results
key_Identify_df <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanRangePos)) {

  # loop through each element in the list
  for (j in seq_along(MeanRangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df <- rbind(key_Identify_df, data.frame(Frame_Quantity = names
       (MeanRangePos)[i],
                                Trial = names(MeanRangePos[[i]][j]),
                                Correct_FramesNum =MeanRangePos[[i]][[j]],
```

```
                                    stringsAsFactors = FALSE))
  }
}


dfjoined <- left_join(Exp2Identify2_LongV3, key_Identify_df, by = c("Frame_
    Quantity","Trial"))

Exp2_ID2_df <- dfjoined
# save(Exp2_ID1_df,Exp2_ID2_df, file = "clean_data/exp2main.RData")
```

Identify 3

```
#drop unnecessary columns
Exp2Identify3_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything() ,.fns = as.character)) %>%
  dplyr::filter(task == "Look3")



Id3key <- data.frame(Trial = unlist(trialListId3)[seq(1, length(trialListId3),
     by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetId3 <- listOlist_dats[names(listOlist_dats) %in% Id3key$Frame_
    Quantity]


for(i in 1:length(exp2subsetId3)) {
  for(j in 1:length(exp2subsetId3[[i]])){
  exp2subsetId3[[i]][[j]] <- exp2subsetId3[[i]][[j]][j == Id3key$Trial[i]]
  }
 exp2subsetId3[[i]] <-  exp2subsetId3[[i]][lengths(exp2subsetId3[[i]]) != 0]
}

exp2subsetId3[[1]] <- lapply(exp2subsetId3[[1]], function(x) x[x$company_name
    %in% c("A", "F"), ])



frameMaxes <- lapply(exp2subsetId3, function(x) lapply(x, function(y) tapply(y
    [,2],INDEX = y[,3],FUN = max)))

for (i in seq_along(frameMaxes)) {
```

```
  frameMaxes[[i]] <- lapply(frameMaxes[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}

hline <- read_csv("raw_data/Geom_hline.csv") %>%
  dplyr::rename("Trial" = Seed) %>%
  dplyr::mutate(across(c("Frame_Quantity","Trial"), .fns = as.character))


# create an empty data.frame to store the results
key_Identify_df_maxes <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Individual_FramesNum = integer(),
                  Correct_FramesValue = integer(),
                  stringsAsFactors = FALSE)


# loop through each list in the main list
for (i in seq_along(frameMaxes)) {

  # loop through each element in the list
  for (j in seq_along(frameMaxes[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_maxes <- rbind(key_Identify_df_maxes,
                                  data.frame(Frame_Quantity = names(frameMaxes
                                      )[i],
                                  Trial = names(frameMaxes[[i]][j]),
                                  Individual_FramesNum = names(frameMaxes[[i]][[j
                                      ]]),
                                  Correct_FramesValue =frameMaxes[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}


# end Maxes ------------------------------------------------------------


dfjoinedMaxes1 <- left_join(key_Identify_df_maxes,hline, by = c("Frame_
    Quantity","Trial"))

dfjoinedMaxes <- left_join(Exp2Identify3_LongV3, dfjoinedMaxes1, by = c("Frame
    _Quantity","Trial","Individual_FramesNum"))

Exp2_ID3_df <- dfjoinedMaxes
# save(Exp2_ID1_df,Exp2_ID2_df,Exp2_ID3_df, file = "clean_data/exp2main.RData
    ")
```

Compare 1

```
#drop unnecessary columns
Compare1_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(Individual_FramesNum,ResponseCode), .after = Trial) %>%
  dplyr::mutate(across(.cols = everything(),.fns = as.character)) %>%
  dplyr::filter(task == "Comp1")

comparelist <- list(c(1,2),c(1, 6),c(1,7),c(16,24),c(55,66))

Comp1key <- data.frame(Trial = unlist(trialListComp1)[seq(1, length(
    trialListComp1), by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetComp1 <- listOlist_dats[names(listOlist_dats) %in% Comp1key$Frame_
    Quantity]


for(i in 1:length(exp2subsetComp1)) {
  for(j in 1:length(exp2subsetComp1[[i]])){
  exp2subsetComp1[[i]][[j]] <- exp2subsetComp1[[i]][[j]][j == Comp1key$Trial[i
      ]]
  }
 exp2subsetComp1[[i]] <-  exp2subsetComp1[[i]][lengths(exp2subsetComp1[[i]]) !
    = 0]
}

exp2subsetComp1[[1]] <- lapply(exp2subsetComp1[[1]], function(x) x[x$company_
    name %in% c("A", "F"), ])



frameMaxes <- lapply(exp2subsetComp1, function(x) lapply(x, function(y) tapply
    (y[,2],INDEX = y[,3],FUN = max)))

for (i in seq_along(frameMaxes)) {
  frameMaxes[[i]] <- lapply(frameMaxes[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
}




for(i in 1:length(frameMaxes)) {
frameMaxes[[i]] <-  lapply(frameMaxes[[i]], function(x) x[c(comparelist[[i]])
    ])
}
```

```
for(i in 1:length(frameMaxes)) {
  for(j in 1:length(frameMaxes[[1]])){
names(frameMaxes[[i]][[j]]) <- c("1","2")
  }
}
Maxes <- lapply(frameMaxes, function(x) lapply(x, max))

MaxPos <- lapply(frameMaxes, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})



C_ansr <- Compare1_LongV3 %>%
  filter(ResponseCode == 1)


# create an empty data.frame to store the results
key_Identify_df_compare <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MaxPos)) {

  # loop through each element in the list
  for (j in seq_along(MaxPos[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_compare <- rbind(key_Identify_df_compare, data.frame(Frame_
      Quantity = names(MaxPos)[i],
                                Trial = names(MaxPos[[i]][j]),
                                Correct_FramesNum = MaxPos[[i]][[j]],
                                stringsAsFactors = FALSE))
  }
}



dfjoined <- left_join(C_ansr, key_Identify_df_compare, by = c("Frame_Quantity"
    ,"Trial"))

Exp2_C1_df <- dfjoined
# save(Exp2_ID1_df,Exp2_ID2_df,Exp2_ID3_df,Exp2_C1_df, file = "clean_data/
    exp2main.RData")
```

Compare 2

```
#drop unnecessary columns
Compare2_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info, -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(Individual_FramesNum,ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything() ,.fns = as.character)) %>%
  dplyr::filter(task == "Comp2")

comparelist <- list(c(1,2),c(1, 6),c(1,7),c(16,24),c(55,66))

Comp2key <- data.frame(Trial = unlist(trialListComp2)[seq(1, length(
    trialListComp2), by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetComp2 <- listOlist_dats[names(listOlist_dats) %in% Comp2key$Frame_
    Quantity]


for(i in 1:length(exp2subsetComp2)) {
  for(j in 1:length(exp2subsetComp2[[i]])){
  exp2subsetComp2[[i]][[j]] <- exp2subsetComp2[[i]][[j]][j == Comp2key$Trial[i
      ]]
  }
 exp2subsetComp2[[i]] <-  exp2subsetComp2[[i]][lengths(exp2subsetComp2[[i]]) !
    = 0]
}

exp2subsetComp2[[1]] <- lapply(exp2subsetComp2[[1]], function(x) x[x$company_
    name %in% c("A", "F"), ])

f <- function(x) mean(x)*(max(x)-min(x)) # Multiply mean and range for
    meanrange score

# Put mean range score in our list of lists
frameMeanRange <- lapply(exp2subsetComp2, function(x) lapply(x, function(y)
    tapply(y[,2],INDEX = y[,3],FUN = f)))

frameMeanRange[[1]] <-  lapply(frameMeanRange[[1]], function(x) x[c("A", "F")
    ]) #change 2 frames from 6 to 2


# ------------------------------------------------
for (i in seq_along(frameMeanRange)) {
  frameMeanRange[[i]] <- lapply(frameMeanRange[[i]], function(x) {
    names(x) <- seq(1, length(x), 1)
    return(x)
  })
  }
```

```
for(i in 1:length(frameMeanRange)) {
frameMeanRange[[i]] <- lapply(frameMeanRange[[i]], function(x) x[c(
    comparelist[[i]])])
}

for(i in 1:length(frameMeanRange)) {
  for(j in 1:length(frameMeanRange[[1]])){
     names(frameMeanRange[[i]][[j]]) <- c("1","2")
  }
}


MeanRange <- lapply(frameMeanRange, function(x) lapply(x, max))

MeanRangePos <- lapply(frameMeanRange, function(x) {
  lapply(x, function(y) {
    max_index <- which.max(y)
    names(y)[max_index]
  })
})

# create an empty data.frame to store the results
key_comp_df <- data.frame(Frame_Quantity = integer(),
                  Trial = integer(),
                  Correct_FramesNum = integer(),
                  stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(MeanRangePos)) {

  # loop through each element in the list
  for (j in seq_along(MeanRangePos[[i]])) {

    # extract the values and add them to the data.frame
   key_comp_df <- rbind(key_comp_df, data.frame(Frame_Quantity = names(
       MeanRangePos)[i],
                                  Trial = names(MeanRangePos[[i]][j]),
                                  Correct_FramesNum =MeanRangePos[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}



C_ansr <- Compare2_LongV3 %>%
  filter(ResponseCode == 1)


dfjoined <- left_join(C_ansr, key_comp_df, by = c("Frame_Quantity","Trial"))
Exp2_C2_df <- dfjoined
# save(Exp2_ID1_df,Exp2_ID2_df,Exp2_ID3_df,Exp2_C1_df,Exp2_C2_df, file = "
    clean_data/exp2main.RData")
```

Summarize 1

(Summarize 1) Is the general trend in the graphs going up or down? (This was a multiple-choice question with choices Up and Down).

```
Exp2Full_Long <- Exp2FullV2 %>%
  select(-contains("Look"),-contains("Comp"),-contains("Ense2")) %>%
  pivot_longer(
    cols = c(Ense1_2Graphs20:Ense1_70Graphs8) ,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)%>%
    dplyr::mutate(ResponseCode = ifelse(Response == "Up",1,0))

#split column
Exp2Full_LongV2<- cSplit(Exp2Full_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Exp2Full_LongV2)[colnames(Exp2Full_LongV2) =="trial_info_3"] <- "
    Individual_FramesNum"

#getting frame # by it's self
Exp2Full_LongV2$Frame_Quantity <- sub('Graphs.*', '', Exp2Full_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Exp2Full_LongV2$Trial <- sub('.*Graphs', '', Exp2Full_LongV2$trial_info_2)
Exp2Full_LongV2$Trial <- sub('\\.', '', Exp2Full_LongV2$Trial)

#drop unnecessary columns
Summarize1_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
         -trial_info , -Race_8_TEXT,-Response) %>%
  dplyr::relocate(c(ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything() ,.fns = as.character)) %>%
  dplyr::filter(task == "Ense1")



summ1key <- data.frame(Trial = unlist(trialListSumm1)[seq(1, length(
    trialListSumm1), by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetsumm1 <- listOlist_dats[names(listOlist_dats) %in% summ1key$Frame_
    Quantity]

for(i in 1:length(exp2subsetsumm1)) {
  for(j in 1:length(exp2subsetsumm1[[i]])){
  exp2subsetsumm1[[i]][[j]] <- exp2subsetsumm1[[i]][[j]][j == summ1key$Trial[i
      ]]
```

```
  }
 exp2subsetsumm1[[i]] <-  exp2subsetsumm1[[i]][lengths(exp2subsetsumm1[[i]]) !
    = 0]
}

exp2subsetsumm1[[1]] <- lapply(exp2subsetsumm1[[1]], function(x) x[x$company_
   name %in% c("A", "F"), ])


f <- function(x) last(x) - first(x)

framedifs <- lapply(exp2subsetsumm1, function(x) lapply(x,  function(y) {
  ifelse(sum(tapply(y[,2], y[,3], f))>0,1,0)
  }))

names(framedifs) <- unique(Summarize1_LongV3$Frame_Quantity)



# create an empty data.frame to store the results
key_Identify_df_summ <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_FramesNum = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(framedifs)) {

  # loop through each element in the list
  for (j in seq_along(framedifs[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_summ <- rbind(key_Identify_df_summ, data.frame(Frame_
      Quantity = names(framedifs)[i],
                              Trial =names(framedifs[[i]][j]) ,
                              Correct_FramesNum =framedifs[[i]][[j]],
                              stringsAsFactors = FALSE))
  }
}


dfjoined <- left_join(Summarize1_LongV3, key_Identify_df_summ, by = c("Frame_
   Quantity","Trial"))

Exp2_S1_df <- dfjoined
# save(Exp2_ID1_df,Exp2_ID2_df,Exp2_ID3_df,Exp2_C1_df,Exp2_C2_df,Exp2_S1_df,
    file = "clean_data/exp2main.RData")
```

Summarize 2

```
Exp2Full_Long <- Exp2FullV2 %>%
  select(-contains("Look"),-contains("Comp"),-contains("Ense1")) %>%
  pivot_longer(
    cols = c(Ense2_2Graphs11:Ense2_70Graphs11) ,
    names_to = "trial_info",
    values_to = "Response",
    values_drop_na = TRUE)#%>%
    # dplyr::mutate(ResponseCode = ifelse(Response == "Up",1,0))

#split column
Exp2Full_LongV2<- cSplit(Exp2Full_Long, "trial_info", "_", drop = FALSE)
#name new column
colnames(Exp2Full_LongV2)[colnames(Exp2Full_LongV2) =="trial_info_3"] <- "
    Individual_FramesNum"

#getting frame # by it's self
Exp2Full_LongV2$Frame_Quantity <- sub('Graphs.*', '', Exp2Full_LongV2$trial_
    info_2)

#getting frame stimuli group by it's self
Exp2Full_LongV2$Trial <- sub('.*Graphs', '', Exp2Full_LongV2$trial_info_2)
Exp2Full_LongV2$Trial <- sub('\\.', '', Exp2Full_LongV2$Trial)

#drop unnecessary columns
Summarize2_LongV3 <- Exp2Full_LongV2 %>%
  dplyr::filter(!(Frame_Quantity == "6" & Trial == "2")) %>%
  dplyr::mutate(task = trial_info_1) %>%
  dplyr::select(-trial_info_1, -trial_info_2,-Gender_4_TEXT,-Educate_9_TEXT,
        -trial_info, -Race_8_TEXT) %>%
  # dplyr::relocate(c(ResponseCode) , .after = Trial) %>%
  dplyr::mutate(across(.cols = everything() ,.fns = as.character))



for(i in 1:length(listOlist_dats[[1]])) {
listOlist_dats[[1]][[i]] <- listOlist_dats[[1]][[i]][listOlist_dats[[1]][[i]]$
    company_name =="A"|listOlist_dats[[1]][[i]]$company_name =="F",]
}

summ2key <- data.frame(Trial = unlist(trialListSumm2)[seq(1, length(
    trialListSumm2), by = 1)]) %>%
  rownames_to_column(var = "Frame_Quantity")

exp2subsetsumm2 <- listOlist_dats[names(listOlist_dats) %in% summ2key$Frame_
    Quantity]

for(i in 1:length(exp2subsetsumm2)) {
  for(j in 1:length(exp2subsetsumm2[[i]])){
  exp2subsetsumm2[[i]][[j]] <- exp2subsetsumm2[[i]][[j]][j == summ2key$Trial[i
      ]]
  }
 exp2subsetsumm2[[i]] <-  exp2subsetsumm2[[i]][lengths(exp2subsetsumm2[[i]]) !
    = 0]
}
```

```
exp2subsetsumm2[[1]] <- lapply(exp2subsetsumm2[[1]], function(x) x[x$company_
    name %in% c("A", "F"), ])



# MEAN
-------------------------------------------------------------------------------

frameMean <- lapply(exp2subsetsumm2, function(x) lapply(x, function(y) mean(y
    [,2])))



names(frameMean) <- unique(Summarize2_LongV3$Frame_Quantity)


# create an empty data.frame to store the results
key_Identify_df_mean <- data.frame(Frame_Quantity = integer(),
                 Trial = integer(),
                 Correct_Mean = integer(),
                 stringsAsFactors = FALSE)

# loop through each list in the main list
for (i in seq_along(frameMean)) {

  # loop through each element in the list
  for (j in seq_along(frameMean[[i]])) {

    # extract the values and add them to the data.frame
   key_Identify_df_mean <- rbind(key_Identify_df_mean, data.frame(Frame_
       Quantity = names(frameMean)[i],
                                  Trial = names(frameMean[[i]])[j],
                                  Correct_Mean =frameMean[[i]][[j]],
                                  stringsAsFactors = FALSE))
  }
}



dfjoined <- left_join(Summarize2_LongV3, key_Identify_df_mean, by = c("Frame_
    Quantity","Trial"))

Exp2_S2_df <- dfjoined
# save(Exp2_ID1_df, Exp2_ID2_df, Exp2_ID3_df, Exp2_C1_df, Exp2_C2_df, Exp2_S1_df,
    Exp2_S2_df, file = "clean_data/exp2main.RData")
```