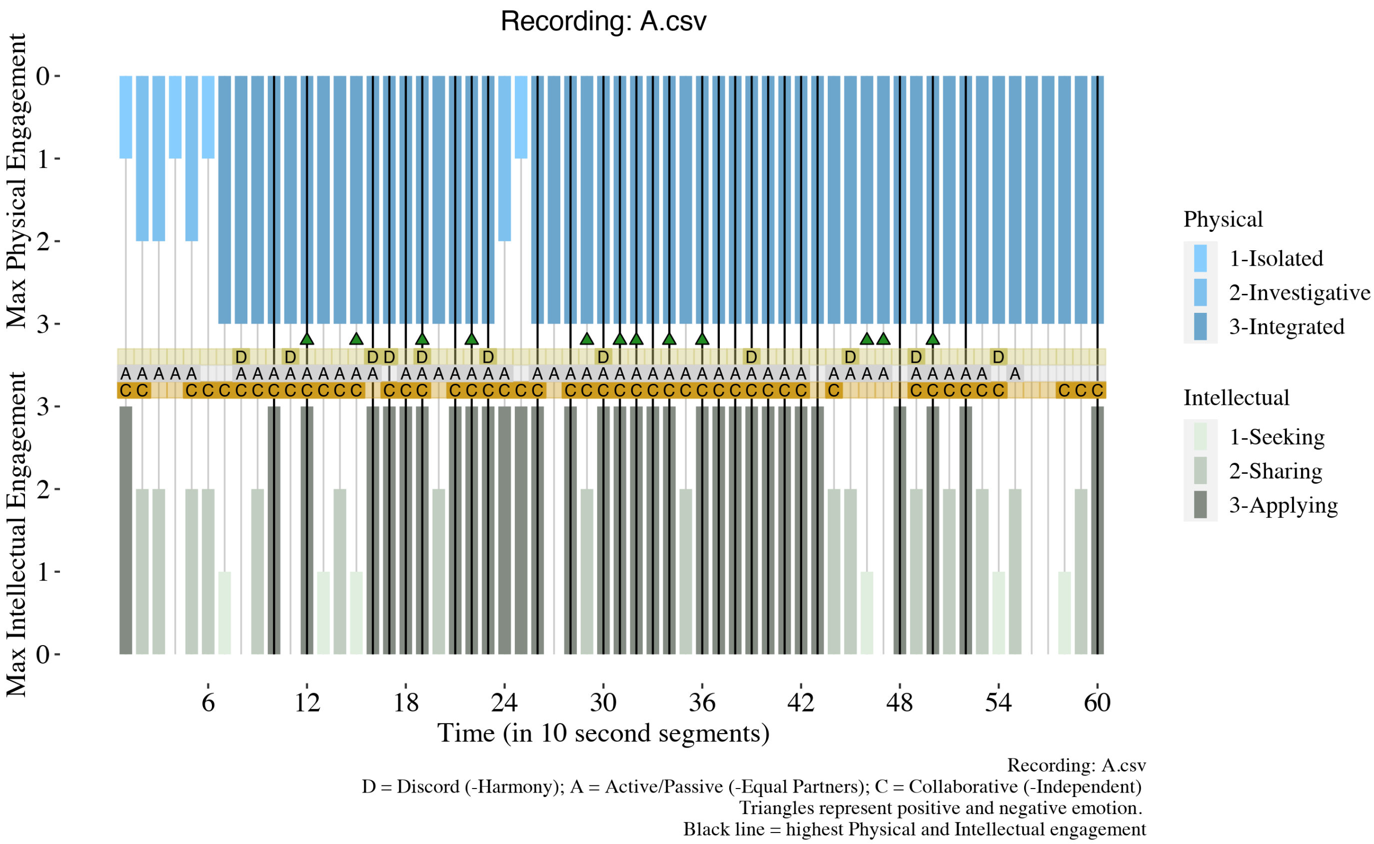
Build APEX Plots

APEX Evaluation Team

8/7/2020

Here are step-by-step instructions for recreating the APEX plots in R. The final product will look something like this.



Raw Data Formatting

This example contains three sample .csv files saved in a directory called “raw data”.

Raw data are generally saved in .csv files (one file per recording) and include the following columns.

- TimeInterval: Each recording is broken into 10-second segments and this variable is a sequential numbering for each segment.

Physical and social codes: each column takes either a 1 or a 0. The coding scheme allows for the presence of two subcategories per major category. For example, one segment might display both Physical Isolated Manipulation and Physical Investigative Manipulation.

- P_Isolated: Physical level 1 (isolated manipulation)
- P_Investigative: Physical level 2 (investigative manipulation)
- P_Integrated: Physical level 3 (integrated manipulation)
- I_Seeking: Intellectual level 1 (seeking)
- I_Sharing: Intellectual level 2 (sharing)
- I_Applying: Intellectual level 3 (applying)

Social codes are comprised of three binary codes and raters determine whether each segment exhibits either ActivePassive or EqualPartners; Harmony or Discord; and Independent or Collaborative

- S_ActivePassive: Social code; 1 if the segment exhibits an active/passive roles among participants
- S_ EqualPartners: Social code (opposite to S_ActivePassive) in which the participants interact as equal partners
- S_Harmony: Social code in which participants work together harmoniously
- S_Discord: Social code (opposite to S_Harmony) in which participants exhibit discord.
- S_Independent: Social code in which participants primarily work independently
- S_Collaborative: Social code (opposite to S_Independent) in which participants collaborate

Emotional Code

- Emotional: Positive emotion is expressed as 1; negative emotion is -1; 0 is neutral

```
# Load the necessary libraries
library(tidyverse) # because nobody should live without it
library(data.table) # for rbindlist
library(kableExtra) # for rendering tables
library(ggnewscale) #allows multiple legends

# This reads raw data files and compiles them into one dataframe
csv_path <- "raw data/"
files <- list.files(path = csv_path, pattern = ".csv")
dat <- NULL
for(fileName in files){
  dat[[fileName]] <- read.csv(paste0(csv_path,fileName),
                              header = TRUE,
                              sep = ",",
                              na.strings = "-",
                              stringsAsFactors = FALSE,
                              fill=TRUE) %>%
    mutate(recording = fileName) # identify the fileName in the dataset
}

dat<-rbindlist(dat)
```

The code above creates the following data table:

TimeInterval	P_Isolated	P_Investigative	P_Integrated	I_Seeking	I_Sharing	I_Applying	S_ActivePa
1	1	0	0	0	1	1	
2	1	1	0	1	1	0	
3	1	1	0	1	1	0	
4	1	0	0	0	0	0	
5	1	1	0	1	1	0	

Data Cleaning

Each APEX value is either a 1 or 0 (except for emotional which is either -1 for negative emotion, 0 for neutral, and 1 for positive emotion). To render each value on the chart, we need to replace the value of each 1 with the appropriate Y axis value. For example, all of the emotional codes will get placed on the Y axis at value 3.8. The binary social codes (Discord vs. Harmony; ActivePassive vs. Equal Partners; Collaborative vs. Independent) each get placed on the Y axis at 3.6, 3.4, and 3.2, respectively. Most of the manipulation in this chunk is establishing the appropriate Y axis values for each APEX code.

```
# Replace 1s with levels
dat <- dat %>%
  mutate(P_Investigative = P_Investigative * 2,
         P_Integrated = P_Integrated * 3,
         I_Sharing = I_Sharing * 2,
         I_Applying = I_Applying * 3)

# Create max intellectual and max physical.  This will be used in the APEX plots
set_max <- function(x)
    if((sum(x)>0) & !is.na(sum(x))) {
      max(x[x > 0])
    }else{
      return(NA)}

# Collect highest recorded physical and intellectual values for each segment
dat <- dat %>%
  rowwise() %>%
  mutate(max_P = set_max(c(P_Isolated, P_Investigative, P_Integrated)),
         max_I = set_max(c(I_Seeking, I_Sharing, I_Applying))) %>%

  # The plot contains both physical and intellectual bars
  # Physical bars descend from the top; intellectual bars go up from the bottom of the plot
  # Therefore, we need to renumber each physical value to count down from the top (7)
  mutate(max_P = 7 - max_P) %>%

  # also identify segments where intellectual and physical are at 3
  mutate(max_ip = ifelse(P_Integrated == 3 & I_Applying == 3, 1, NA)) %>%

  # Create separate positive and negative emotional categories
  # .8 helps us place this on the chart - it will appear at y position 3.8
  mutate(neg_emotional = as.numeric(ifelse(Emotional == -1, .8, NA)),
         pos_emotional = as.numeric(ifelse(Emotional == 1, .8, NA))) %>%

  # Set Y axis values for each social binary
  mutate(S_Discord = as.numeric(ifelse(S_Discord == 1, 0.6, NA)),
         S_Harmony = as.numeric(ifelse(S_Harmony == 1, 0.6, NA)),
         S_ActivePassive = as.numeric(ifelse(S_ActivePassive == 1, 0.4, NA)),
         S_EqualPartners = as.numeric(ifelse(S_EqualPartners == 1, 0.4, NA)),
         S_Collaborative = as.numeric(ifelse(S_Collaborative == 1, 0.2, NA)),
         S_Independent = as.numeric(ifelse(S_Independent == 1, 0.2, NA)))
```

Build the Plot

We build the plot one element at a time. Let’s begin with setting some initial variables (e.g. caption, axis scales, colors) along with the plot background and thematic elements.

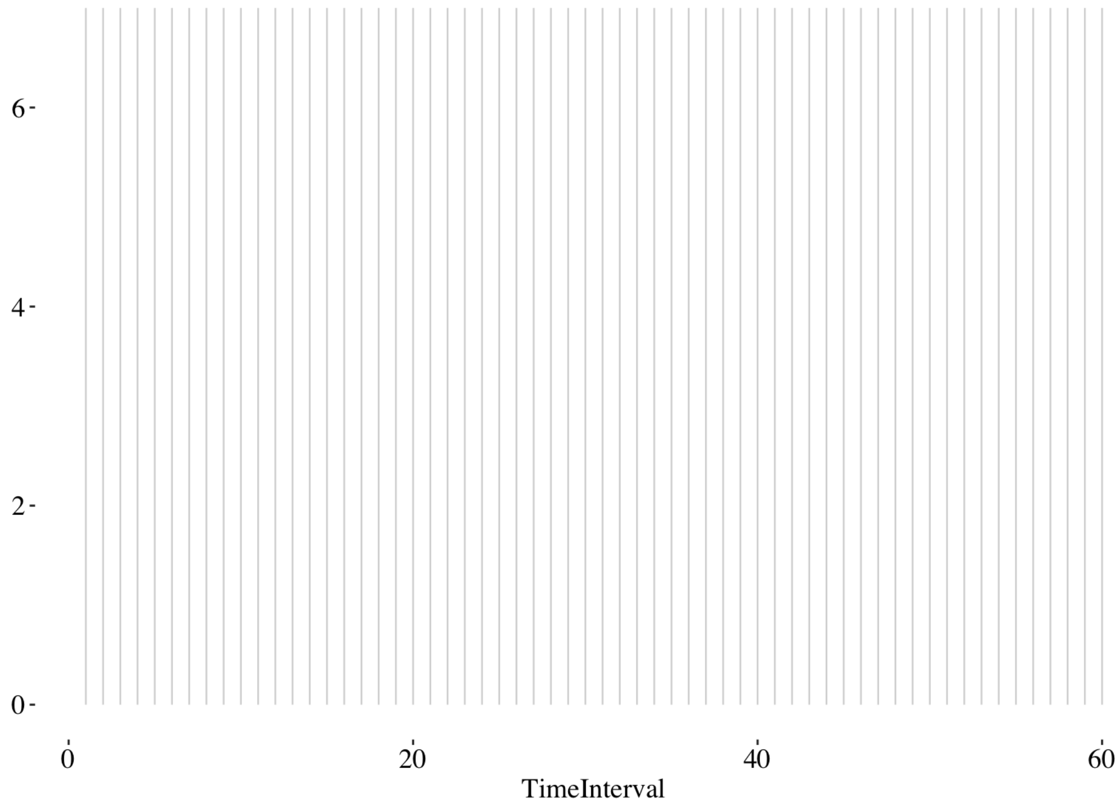
Set Initial Variables

```
# Let's just get one recording (A.csv) from our dataframe
single_plot <- dat %>% filter(recording == "A.csv")
recording_name <- unique(single_plot$recording)

# Set initial plot variables
num_segments <- max(dat$TimeInterval)
common_x_scale <- scale_x_continuous(limits = c(0, num_segments), breaks = seq(6, num_segments,
 by = 6)) #common scale
plot_caption <- paste("Recording:", recording_name, sep = " ")
plot_caption <- paste(plot_caption, "D = Discord (-Harmony); A = Active/Passive (-Equal Partner
s); C = Collaborative (-Independent)
                           Triangles represent positive and negative emotion.
                           Black line = highest Physical and Intellectual engagement", sep =
"\n")

# Set color scheme for Physical and Intellectual Bars
cols_p <- c("6" = "skyblue1","5" = "skyblue2","4" = "skyblue3")
cols_i <- c("1" = "honeydew2","2" = "honeydew3","3" = "honeydew4")

# Set colors for social codes
color_A <- "lightgrey"    #for ActivePassive vs. EqualPartners
color_D <- "khaki3"       #for Discord vs. Harmony
color_C <- "goldenrod3"   #for Collaborative vs. Independent
```

Plot Background and Theme

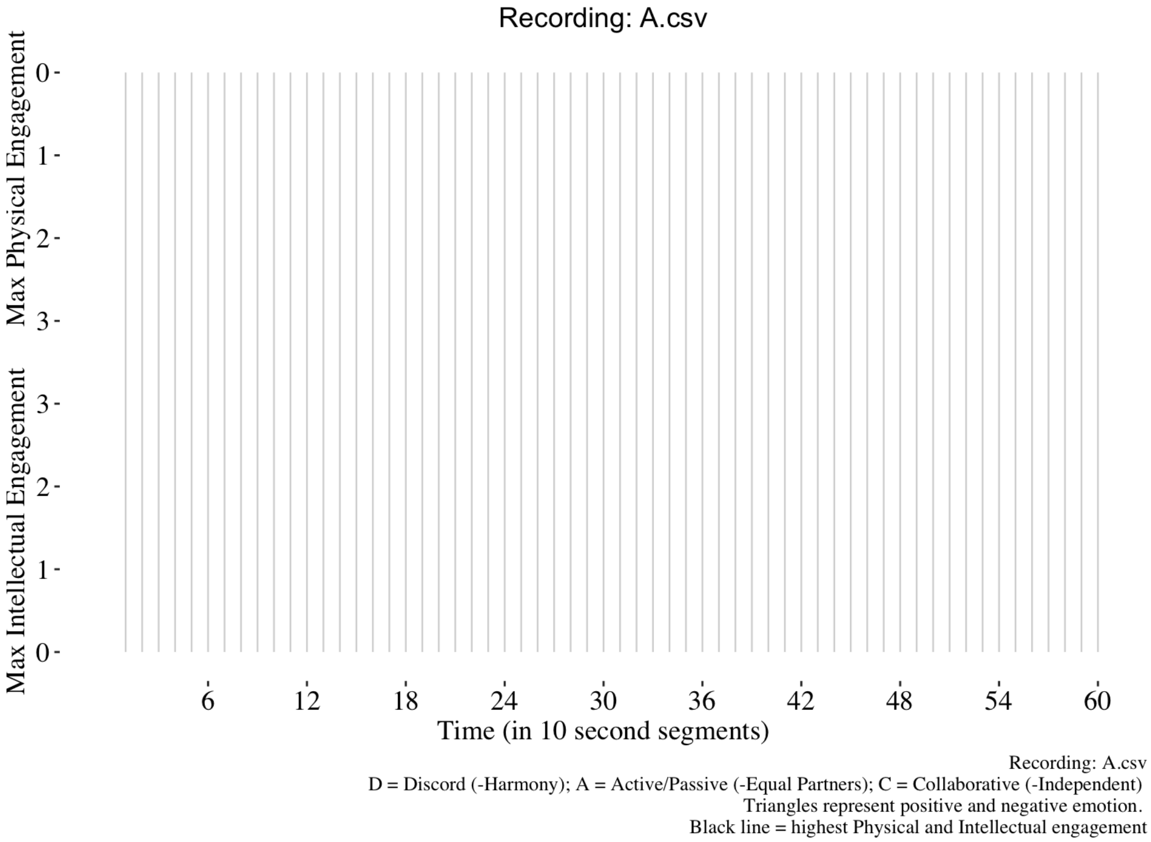
In the R Markdown header, the figure width is set to 10 (fig.width=10), and the height is set to 6 (fig.height=6).

```
# This chunk builds a single plot from data from one recording.
# To create multiple plots, place this in a for loop that iterates through each recording
apex_plot <- ggplot(single_plot, aes(x = TimeInterval)) +
      geom_linerange(aes(ymin = 0, ymax = 7),  alpha = 0.2, size = 0.4) +  #background bars

      ### General Plot Aesthetics
      theme(panel.background = element_blank(),  #removes the gray background R uses as a defaul
t
            axis.text = element_text(size = 14, family = "serif", color = "black"),
            axis.title = element_text(size = 14, family = "serif", color = "black"),
            plot.title = element_text(hjust = 0.5, size = 14), #Center the title
            plot.caption = element_text(size = 10, family = "serif", color = "black"),
            plot.margin = unit(c(.2,ifelse(nrow(single_plot)>(num_segments*.9), 5, 0),.2,0), "c
m"), # make room for the legend if necessary
            legend.text = element_text(size = 12, family = "serif", color = "black"),
            legend.title = element_text(size = 12, family = "serif", color = "black"),
            legend.position = c((nrow(single_plot)/num_segments)+.12,.5)) # put the legend close
to the plot

apex_plot
```



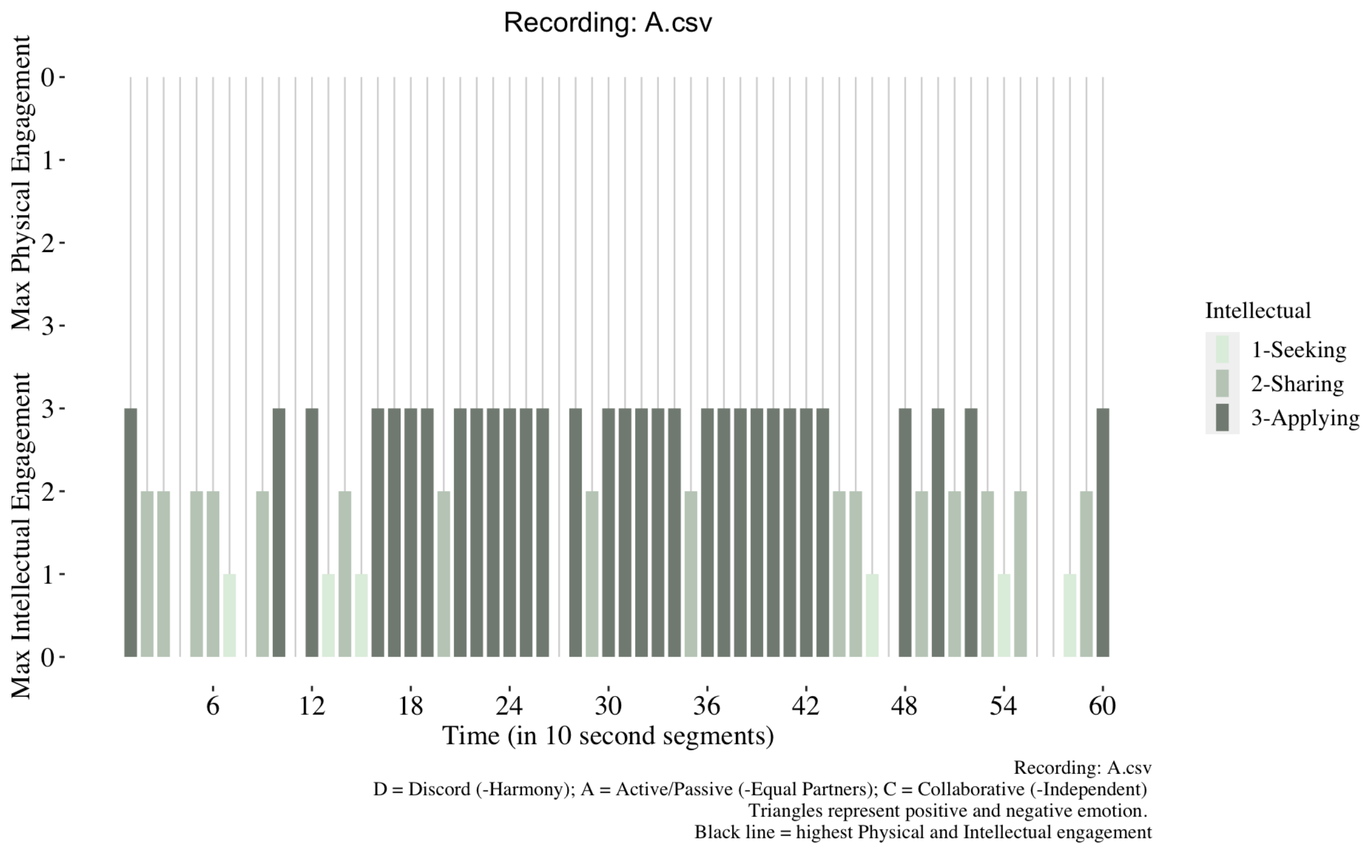
Title and Labels

```
apex_plot <- apex_plot +

      ### Axis Labels, Title, and Caption
      labs(y = "Max Intellectual Engagement      Max Physical Engagement", x = "Time (in 10 seco
nd segments)",
           title = paste("Recording:", recording_name, sep = " "),
          caption = plot_caption, size = 6) +
      scale_y_continuous(breaks = c(0, 1, 2, 3, 4, 5, 6, 7),
                          labels = c("0", "1", "2", "3", "3", "2", "1", "0")) +
      common_x_scale   #x axis time increments


apex_plot
```



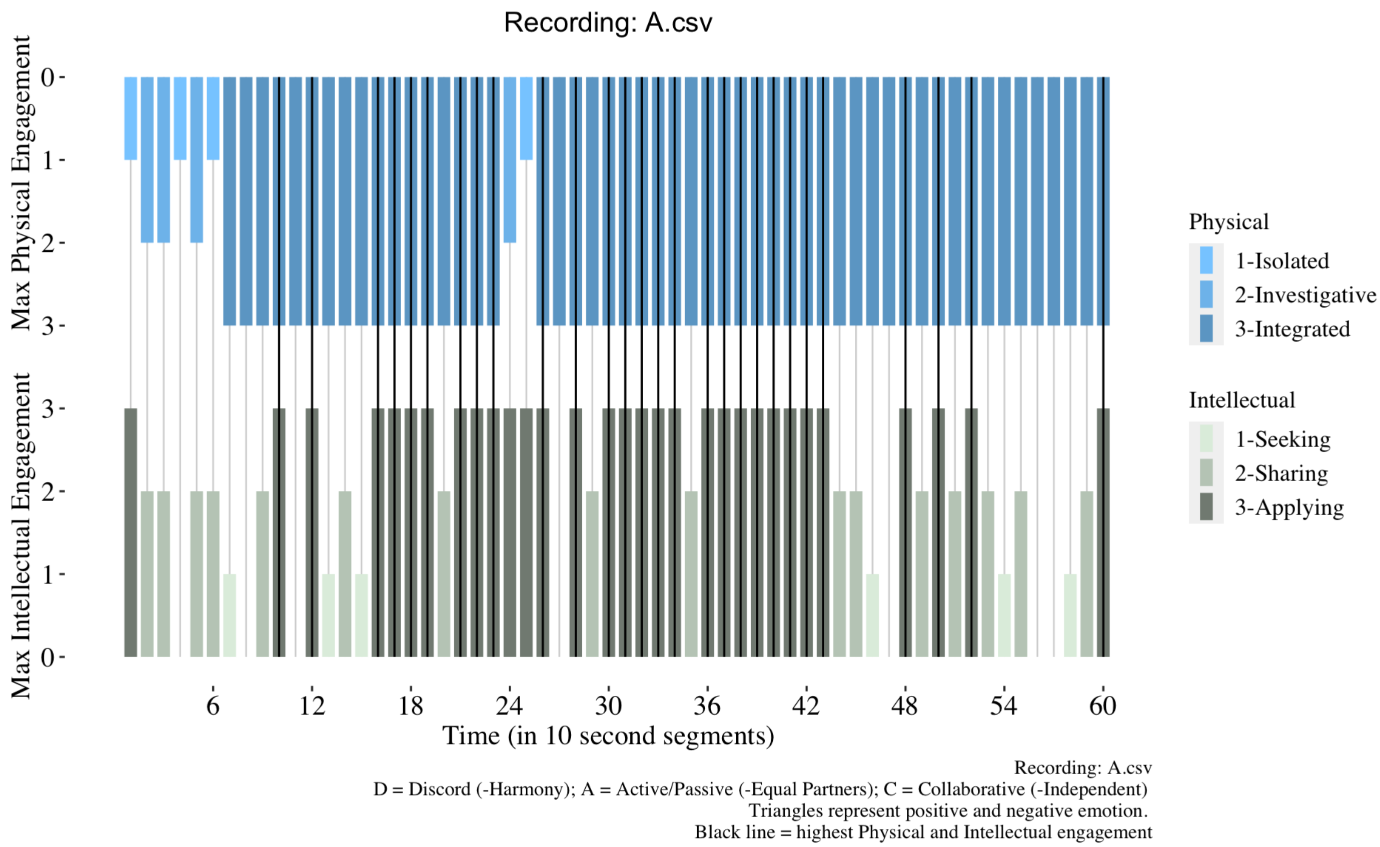
Add Intellectual Codes

```
apex_plot <- apex_plot +

      ### Intellectual
      geom_linerange(aes(ymin = 0, ymax = max_I, colour = as.factor(max_I)), size = 3) +
      scale_colour_manual(name = "Intellectual", breaks = c("1", "2", "3"),
                          values = cols_i,
                          limits = c("1", "2", "3"),
                          labels = c("1-Seeking", "2-Sharing", "3-Applying")) +
      new_scale_color()

apex_plot
```



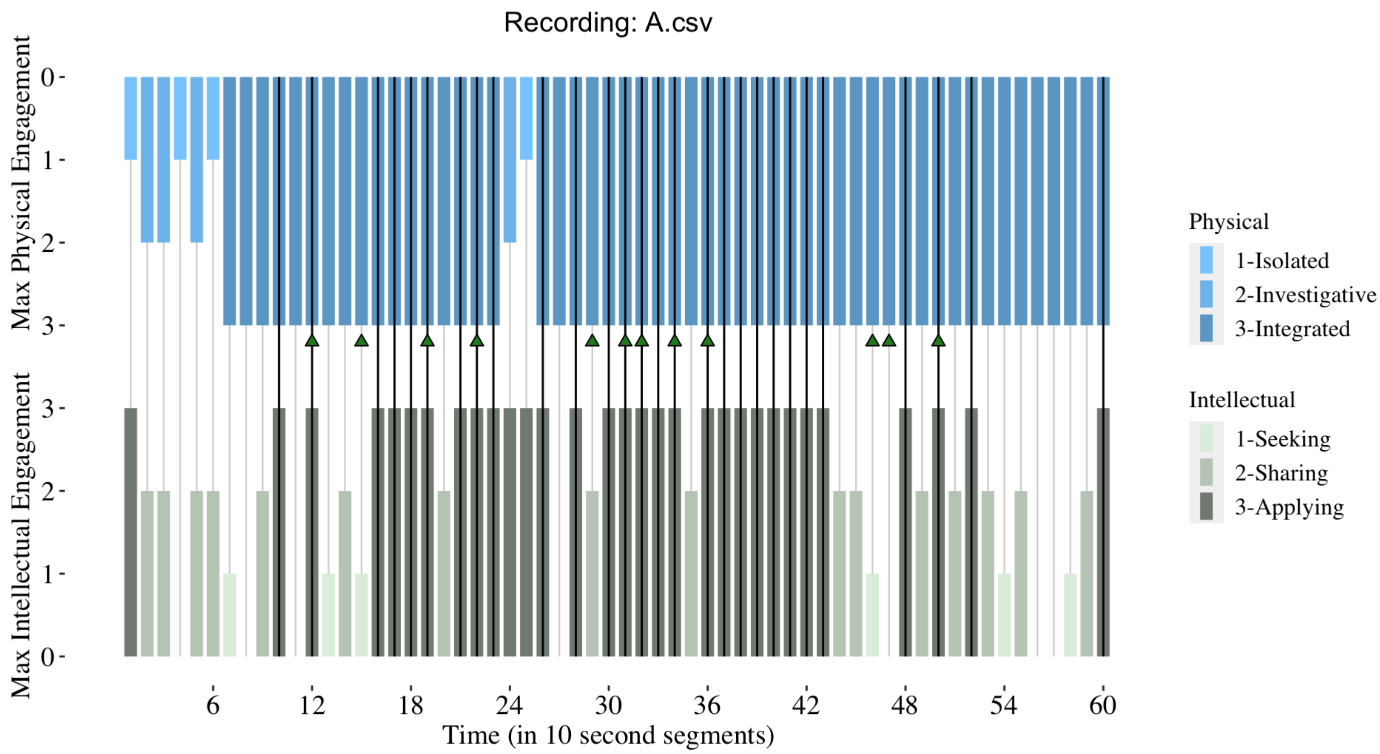
Add Physical

```
apex_plot <- apex_plot +

      ### Physical
      geom_linerange(aes(ymin = 7, ymax = max_P, color = as.factor(max_P)), size = 3) +
      scale_colour_manual(name = "Physical", breaks = c("6", "5", "4"),
                          values = cols_p,
                          limits = c("6", "5", "4"),
                          labels = c("1-Isolated", "2-Investigative", "3-Integrated")) +
      geom_linerange(aes(ymin = 0, ymax = max_ip*7), color="black")   # both physical and intell
ectual at highest



apex_plot
```



Add Emotional

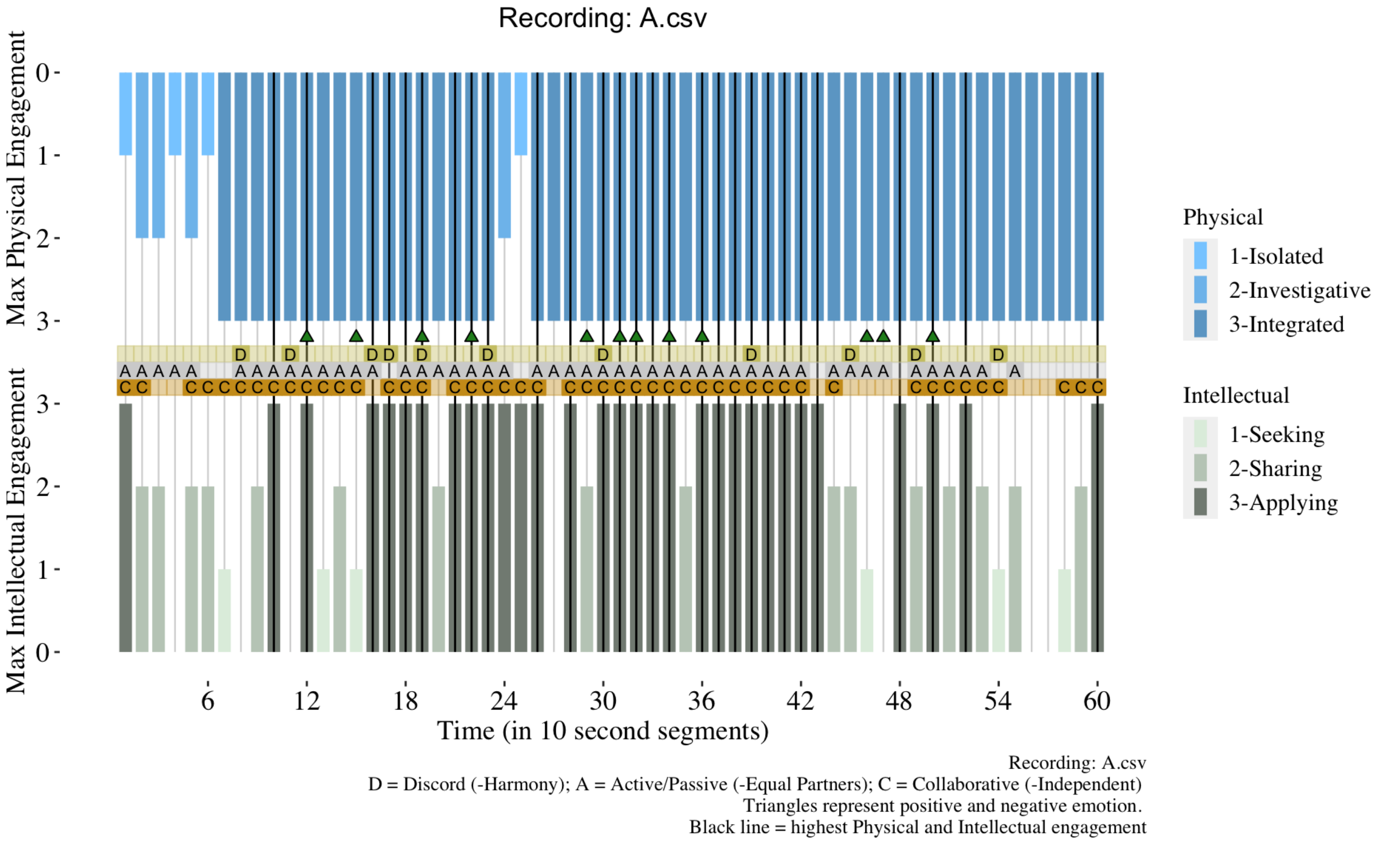
```
apex_plot <- apex_plot +

      ### Emotional
      geom_point(aes(y = pos_emotional + 3), size = 2, shape = 24, fill = "forestgreen") + # pos
itive emotional
      geom_point(aes(y = neg_emotional + 3), size = 2, shape = 25, fill = "firebrick1")   # nega
tive emotional


apex_plot
```



Recording: A.csv
D = Discord (-Harmony); A = Active/Passive (-Equal Partners); C = Collaborative (-Independent)
Triangles represent positive and negative emotion.
Black line = highest Physical and Intellectual engagement

Add Social

```
apex_plot <- apex_plot +

      ### Social
      geom_point(aes(y = S_Harmony + 3), shape = 22, color = color_D, fill=color_D, size = 4, al
pha = .4) +
      geom_point(aes(y = S_Discord + 3), shape = 22, color = color_D, fill=color_D, size = 4) +
      geom_text(aes(y = S_Discord + 3, label = "D"), size = 3) +
      geom_point(aes(y = S_Independent + 3), shape = 22, color = color_C, fill=color_C, size = 4
, alpha = .4) +
      geom_point(aes(y = S_Collaborative + 3), shape = 22, color = color_C, fill=color_C, size =
4) +
      geom_text(aes(y = S_Collaborative + 3, label = "C"), size = 3) +
      geom_point(aes(y = S_EqualPartners + 3), shape = 22, color = color_A, fill=color_A, size =
4, alpha = .4) +
      geom_point(aes(y = S_ActivePassive + 3), shape = 22, color = color_A, fill=color_A, size =
4) +
      geom_text(aes(y = S_ActivePassive + 3, label = "A"), size = 3)


apex_plot
```

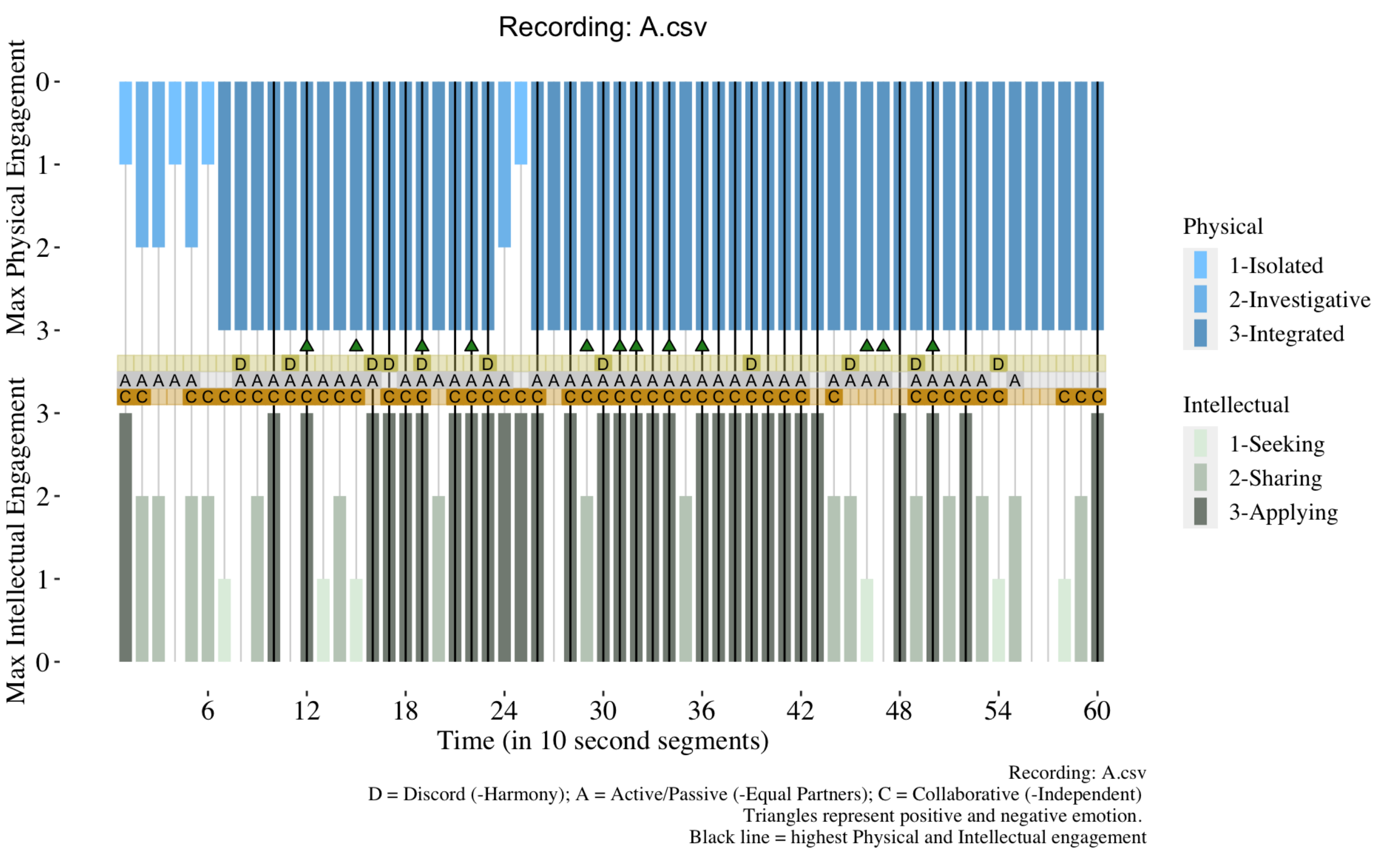


Whole Plot

Here's the code for the entire plot.

```
apex_plot <- ggplot(single_plot, aes(x = TimeInterval)) +
      geom_linerange(aes(ymin = 0, ymax = 7),  alpha = 0.2, size = 0.4) +  #background bars

      ### Intellectual
      geom_linerange(aes(ymin = 0, ymax = max_I, colour = as.factor(max_I)), size = 3) +
      scale_colour_manual(name = "Intellectual", breaks = c("1", "2", "3"),
                          values = cols_i,
                          limits = c("1", "2", "3"),
                          labels = c("1-Seeking", "2-Sharing", "3-Applying")) +
      new_scale_color() +

      ### Physical
      geom_linerange(aes(ymin = 7, ymax = max_P, color = as.factor(max_P)), size = 3) +
      scale_colour_manual(name = "Physical", breaks = c("6", "5", "4"),
                          values = cols_p,
                          limits = c("6", "5", "4"),
                          labels = c("1-Isolated", "2-Investigative", "3-Integrated")) +
      geom_linerange(aes(ymin = 0, ymax = max_ip*7), color="black") +  # both physical and intel
lectual at highest

      ### Emotional
      geom_point(aes(y = pos_emotional + 3), size = 2, shape = 24, fill = "forestgreen") + # pos
itive emotional
      geom_point(aes(y = neg_emotional + 3), size = 2, shape = 25, fill = "firebrick1") +  # neg
ative emotional

      ### Social
      geom_point(aes(y = S_Harmony + 3), shape = 22, color = color_D, fill=color_D, size = 4, al
pha = .4) +
      geom_point(aes(y = S_Discord + 3), shape = 22, color = color_D, fill=color_D, size = 4) +
      geom_text(aes(y = S_Discord + 3, label = "D"), size = 3) +
      geom_point(aes(y = S_Independent + 3), shape = 22, color = color_C, fill=color_C, size = 4
, alpha = .4) +
      geom_point(aes(y = S_Collaborative + 3), shape = 22, color = color_C, fill=color_C, size =
4) +
      geom_text(aes(y = S_Collaborative + 3, label = "C"), size = 3) +
      geom_point(aes(y = S_EqualPartners + 3), shape = 22, color = color_A, fill=color_A, size =
4, alpha = .4) +
      geom_point(aes(y = S_ActivePassive + 3), shape = 22, color = color_A, fill=color_A, size =
4) +
      geom_text(aes(y = S_ActivePassive + 3, label = "A"), size = 3) +

      ### Axis Labels, Title, and Caption
      labs(y = "Max Intellectual Engagement      Max Physical Engagement", x = "Time (in 10 seco
nd segments)",
           title = paste("Recording:", recording_name, sep = " "),
          caption = plot_caption, size = 6) +
      scale_y_continuous(breaks = c(0, 1, 2, 3, 4, 5, 6, 7),
                          labels = c("0", "1", "2", "3", "3", "2", "1", "0")) +  #add code label
s
      common_x_scale +  #x axis time increments

      ### General Plot Aesthetics
```

```
      theme(panel.background = element_blank(),  #removes the gray background R uses as a default
            axis.text = element_text(size = 14, family = "serif", color = "black"),
            axis.title = element_text(size = 14, family = "serif", color = "black"),
            plot.title = element_text(hjust = 0.5, size = 14), #Center the title
            plot.caption = element_text(size = 10, family = "serif", color = "black"),
            plot.margin = unit(c(.2,ifelse(nrow(single_plot)>(num_segments*.9), 5, 0),.2,0), "cm"), # make room for the legend if necessary
            legend.text = element_text(size = 12, family = "serif", color = "black"),
            legend.title = element_text(size = 12, family = "serif", color = "black"),
            legend.position = c((nrow(single_plot)/num_segments)+.12,.5)) # put the legend close to the plot

apex_plot
```




```
######## Save publication-ready plot using:
ggsave(filename = paste(str_remove(recording_name, ".csv"), "-APEX.png", sep = ""),
       type = "cairo",
       #path = "", # use this to place the plot in a separate directory
       width = 10,
       height = 6,
       units = "in")
```