Assignment 3 - First Steps

Joey Heffner
Jae-Young Son
James Wilmott

Initialize working environment

The original study

We are replicating portions of the study 'Maps of subjective feelings' by Lauri Nummenmaa, Riitta Hari, Jari K. Hietanen, and Enrico Glerean. This study can be found at: https://www.pnas.org/content/115/37/9198

The goal of this study was to elucidate organizing principles of subjective experiences of emotions across individuals, and to uncover whether/how emotions can be grouped together to form broad categories of subjective experience.

In Experiment 1, the authors' specific aim was to map the basic dimensions of emotional states. To do so, the authors presented 339 volunteers with 100 words expressing core feelings and bodily processes. The volunteers were then asked to rate these words using five different scales corresponding to the following five basic dimensions:

1. bodily saliency
2. mental saliency
3. emotional intensity
4. controllability
5. lapse (relative frequency of experiencing each emotion)

After collecting the data, the authors investigated whether each basic dimension is related to the others, as visualized in Figure 1. This figure depicts the results of two analyses. First, as shown by the scatterplots, the authors used Spearman's correlations to assess how strongly each word's emotion ratings are associated with the corresponding ratings of all other words. Second, as shown by the histograms, the authors assessed the distributional spread of the words' ratings for each of the five basic dimensions. For these analyses, the authors took all volunteers' ratings, calculated the median rating for each of the 100 words, then z-scored those ratings.

Goal of assignment

For the "first pass" of this assignment, we have replicated Figure 1, which depicts the results of Experiment 1. As a reminder, this study examined how various words (related to a variety of subjective feelings) were rated along five basic dimensions (thought to be important for the experience of emotion). This step involved cleaning the data into a format that we can use for our replication purposes, an important and time-consuming process.

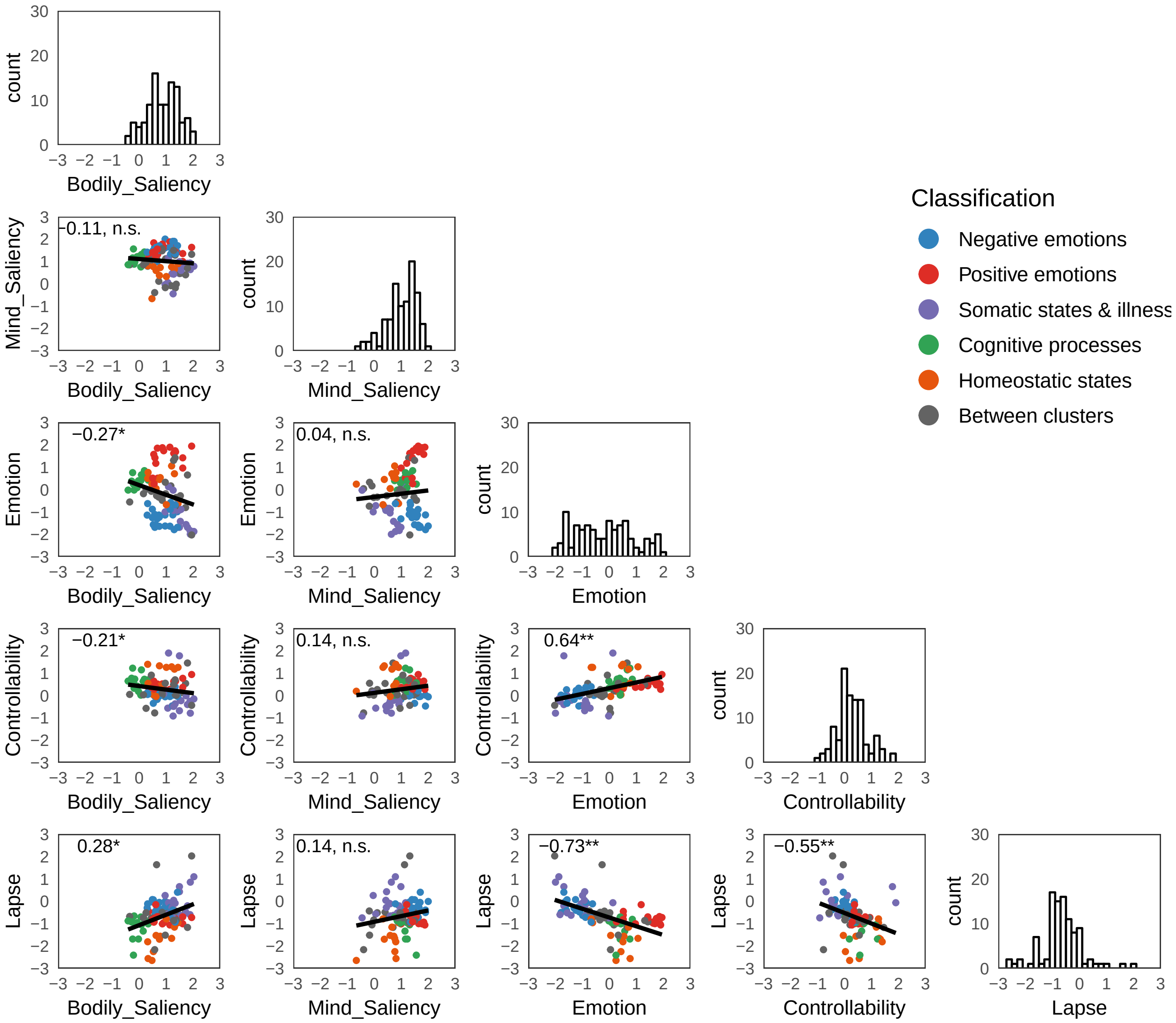
Step 0. Clean and import the data

As an introductory note, the raw data for this study was larger than 2gb, and is all but impossible to access without Matlab. In order to make the analysis replicable using non-proprietary software, we wrote custom Matlab code to clean the ratings used in the study, and to aggregate those ratings into a .csv that can be read into R. That Matlab code has been made available for the interested reader.

Step 1. Get data

Based on the raw data we found, the authors calculated and examined mean and median ratings, and also z-scored ratings. However, they only reported the results using z-scored median ratings in the final paper. To replicate Figure 1, we will subset the dataset to include only those variables.

Step 2. Recreate Figure 1



Next steps

Our project deviates somewhat from the typical project in that our analysis is less regression-based, and more machine-learning-based. Therefore, our main interest is in trying out different types of classification and clustering algorithms, and seeing whether use of these algorithms produces different organizations of subjective feelings. The original paper used the DBSCAN algorithm, and so we are interested in seeing the extent to which other algorithms (such as k-means and k-nearest-neighbors) produce similar patterns of results. In this way, we will not only replicate the analyses from the original paper, but extend them. A secondary goal is to examine the “goodness-of-fit” for the clusters identified in the original paper. One method for accomplishing this is to train a classifier to predict clusters based on subjective feeling ratings. If the clusters identified in the original paper are robust, the classifier should have high accuracy. If we have time (and given the complexity of what we are hoping to accomplish, we may not), we may try to replicate the body heatmaps produced in Figure 3. However, given how much time it took us to replicate Figure 1, we suspect that we will not have sufficient time.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
# Initialize environment
libraryBooks <- c("knitr", "tidyverse", "cowplot")
invisible(lapply(libraryBooks, require, character.only = TRUE)); rm(libraryBooks)
# scriptPath <- dirname(rstudioapi::getSourceEditorContext()$path)
# dataPath <- paste0(scriptPath, "/AllDataForFig1.csv")


# Graph aesthetics
pnas_theme = theme_bw(base_size = 10) +
  theme(text = element_text(size = 10), # Increase the font size
        panel.grid = element_blank(),
        axis.ticks = element_blank()) # remove x & y ticks


## Histogram function
emotion_hist <- function(df, colName) {

  df <- df %>% select(colName)

  # Histogram plot
  emotion_hist_plot <- ggplot(df, aes_string(x = names(df))) + #aes_string() necessary here
    geom_histogram(fill = "white", color = "black", binwidth = .2) +
    coord_cartesian(ylim = c(0, 30), xlim = c(-3, 3), expand = FALSE) +
    pnas_theme

  return(emotion_hist_plot)
}


## Scatter plot function
emotion_scatter <- function(df, xName, yName, grouping) {

  df <- df %>% select(xName, yName, grouping)

  # Correlation number and significance
  cor_value <- sprintf("%.2f", round(cor(df[,1], df[,2], method="spearman"), 2))
  cor_test <- cor.test(df[,1], df[,2], method="spearman")$p.value
  cor_test_text <- if_else(cor_test < 0.005, "**",
                           if_else(cor_test < 0.05, "*", ", n.s."))
  displaySig <- paste0(cor_value, cor_test_text)
```

```r
  # Scatterplot
  emotion_scatterplot <- ggplot(df) +
    aes_string(x = names(df)[1], y = names(df)[2], color=names(df)[3]) + #aes_string() necessary here
    geom_point(size = 1) +
    geom_smooth(method = "lm", se = FALSE, linetype = "solid", color = "black", size = 1) +
    coord_cartesian(ylim = c(-3, 3), xlim = c(-3, 3), expand = FALSE) +
    annotate("text", x = -1.5, y = 2.5, label = displaySig, size = 3) +
    scale_color_manual(name="Classification",
                       labels=c("Negative emotions",
                                "Positive emotions",
                                "Somatic states & illnesses",
                                "Cognitive processes",
                                "Homeostatic states",
                                "Between clusters"),
                       values=c("#3182bd", "#de2d26", "#756bb1",
                                "#31a354", "#e6550d", "#636363")) +
    pnas_theme +
    guides(color = guide_legend(override.aes = list(size=4))) +
    theme(legend.title = element_text(size = 12),
          legend.text = element_text(size = 10))

  emotion_scatterplot_noLegend <- emotion_scatterplot +
    theme(legend.position = "none")

  legend <- get_legend(emotion_scatterplot)

  output <- list(emotion_scatterplot_noLegend, legend)

  return(output)
}

# Load data
# alldata <- read.csv(paste0(dataPath), header = TRUE)
alldata <- read.csv("AllDataForFig1.csv", header = TRUE)

data <- alldata %>%
  select(Bodily_Saliency = BodilySensationStrength_medianZ,
         Mind_Saliency = MindSensationStrength_medianZ,
         Emotion = EmotionIntensity_medianZ,
         Controllability = Controllability_medianZ,
         Lapse = LastTime_medianZ,
         classification = DBSCAN_class) %>%
  mutate(classification = replace(classification, classification=="-1", "Unclassifiable"),
         classification = replace(classification, classification=="1", "1"),
         classification = replace(classification, classification=="2", "2"),
         classification = replace(classification, classification=="3", "3"),
         classification = replace(classification, classification=="4", "4"),
         classification = replace(classification, classification=="5", "5"))

# Create list to hold grobs, which will be smushed together later using gridExtra
grobs <- list() # blank list to hold the grobs
legends <- list()
```

```r
# Generate list of plots to iterate through, in the order they will appear
names_pairs <- list(c('Bodily_Saliency', 'Bodily_Saliency'),
                    c('Bodily_Saliency', 'Mind_Saliency'),
                    c('Mind_Saliency', 'Mind_Saliency'),
                    c('Bodily_Saliency', 'Emotion'),
                    c('Mind_Saliency', 'Emotion'),
                    c('Emotion', 'Emotion'),
                    c('Bodily_Saliency', 'Controllability'),
                    c('Mind_Saliency', 'Controllability'),
                    c('Emotion', 'Controllability'),
                    c('Controllability', 'Controllability'),
                    c('Bodily_Saliency', 'Lapse'),
                    c('Mind_Saliency', 'Lapse'),
                    c('Emotion', 'Lapse'),
                    c('Controllability', 'Lapse'),
                    c('Lapse', 'Lapse'))

# Now, iterate through the name pairs.
for (i in 1:length(names_pairs)) {
  # First, get the name pair
  names <- names_pairs[[i]];

  # Next, check if the names are the same
  name_match_bool = names[1]==names[2];

  # If the names match, create a histogram; if they don't, create a correlation plot
  if (name_match_bool){
    grobs[[i]] <- emotion_hist(data, names[[1]])
  } else {
    scatterData <- emotion_scatter(data, names[[1]], names[[2]], "classification")
    grobs[[i]] <- scatterData[[1]]
    scatterLegend <- scatterData[[2]]
  }
}

# Now, the grobs list holds each of the 15 figures, with the first index corresponding to the topmost p

plot_grid(grobs[[1]], NULL, NULL, NULL, NULL,
          grobs[[2]], grobs[[3]], NULL, NULL, scatterLegend,
          grobs[[4]], grobs[[5]], grobs[[6]], NULL, NULL,
          grobs[[7]], grobs[[8]], grobs[[9]], grobs[[10]], NULL,
          grobs[[11]], grobs[[12]], grobs[[13]], grobs[[14]], grobs[[15]],
          ncol = 5)
```