

Multiple Views

INFO 526 Data Analysis and Visualization

Adriana Picoral

The temperature of emotions

We will be open science experimental data on the temperature of emotions for this part of the tutorial. I downloaded the csv file from the osf.io website and placed it inside the data folder in my project. As usual, we will load tidyverse and read the data in.

```
library(tidyverse)
emotion_temperature <- read_csv("data/dte_exp1.csv")
glimpse(emotion_temperature)

## Rows: 27,060
## Columns: 15
## $ rid      <chr> "R_1", "R_1", "R_1", "R_1", "R_1", "R_1", "R_1", "R_1", "~
## $ UserLanguage <chr> "English", "English", "English", "English", "English", "English", "E~
## $ platform  <chr> "prolific", "prolific", "prolific", "prolific", "prolific~
## $ duration  <dbl> 247, 247, 247, 247, 247, 247, 247, 247, 247, 247, 24~
## $ latitude  <dbl> 28.5274, 28.5274, 28.5274, 28.5274, 28.5274, 28.5274, 28.~
## $ longitude <dbl> -81.6012, -81.6012, -81.6012, -81.6012, -81.6012, -81.601~
## $ age       <dbl> 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 4~
## $ gender    <chr> "Female", "Female", "Female", "Female", "Female", "Female~
## $ firstlang <chr> "English", "English", "English", "English", "English", "English", "E~
## $ country   <chr> "United States", "United States", "United States", "Unite~
## $ emotion   <chr> "Energetic, Excited", "Secure, At ease", "Jittery, Nervou~
## $ valence   <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, ~
## $ arousal   <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ temp_cat  <chr> "temp0", "temp10", "temp40", "temp30", "temp10", "temp0",~
## $ response  <dbl> 1, 2, 5, 4, 2, 1, 3, 2, 3, 2, 1, 5, 5, 1, 3, 1, 5, 5, 2, ~
```

Data Inspection

Let's do a quick inspection on the categorical variables in the data. In addition to `count()` I will be using `kable()` to make the tables look nice in the output document.

How many observations per language do we have?

```
# create table with counts of data points per participant language, sort table
# since language is an unordered categorical variable
emotion_temperature %>%
```

```
count(UserLanguage, sort = TRUE) %>%
kable(caption = "Number of observations per language group")
```

Table 1: Number of observations per language group

UserLanguage	n
English	11460
Spanish	5820
Chinese	4920
Japanese	4860

How many participants per language group?

We have multiple observations per participant, so we need to first ensure we extract distinct participant IDs per language group to then count observations.

```
emotion_temperature %>%
  distinct(rid, UserLanguage) %>%
  count(UserLanguage, sort = TRUE) %>%
  kable(caption = "Number of participants per language group")
```

Table 2: Number of participants per language group

UserLanguage	n
English	191
Spanish	97
Chinese	82
Japanese	81

What are the demographics of the participants, besides language?

To get more in depth on participants' demographics, let's create a dataframe with distinct info related to each participant.

```
participant_info <- emotion_temperature %>%
  distinct(rid, UserLanguage, firstlang, age, gender, country)

participant_info %>%
  head() %>%
  kable(caption = "First six participants, and their demographic information")
```

Table 3: First six participants, and their demographic information

rid	UserLanguage	age	gender	firstlang	country
R_1	English	42	Female	English	United States
R_2	English	56	Male	English	United States
R_3	Japanese	16	Male	Arabic	Afghanistan

rid	UserLanguage	age	gender	firstlang	country
R_4	English	27	Male	English	United States
R_5	Spanish	39	Female	Spanish	Mexico
R_6	Chinese	39	Female	Chinese	Canada

Participant IDs should not repeat in the table above. Let's check that that's actually the case.

```
# are there any repeated participant IDs? Count IDs and filter to see if any
# ID has count higher than 1
participant_info %>%
  count(rid, sort = TRUE) %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: rid <chr>, n <int>
```

No repeated IDs, so everything looks good, we might proceed with our `participant_info` data analysis.

What is the average participant age?

```
participant_info %>%
  group_by(gender) %>%
  summarize(mean_age = mean(age)) %>%
  kable(caption = "Mean participant age per gender group",
        digits = 0)
```

Table 4: Mean participant age per gender group

gender	mean_age
Female	34
Male	35
Other	31

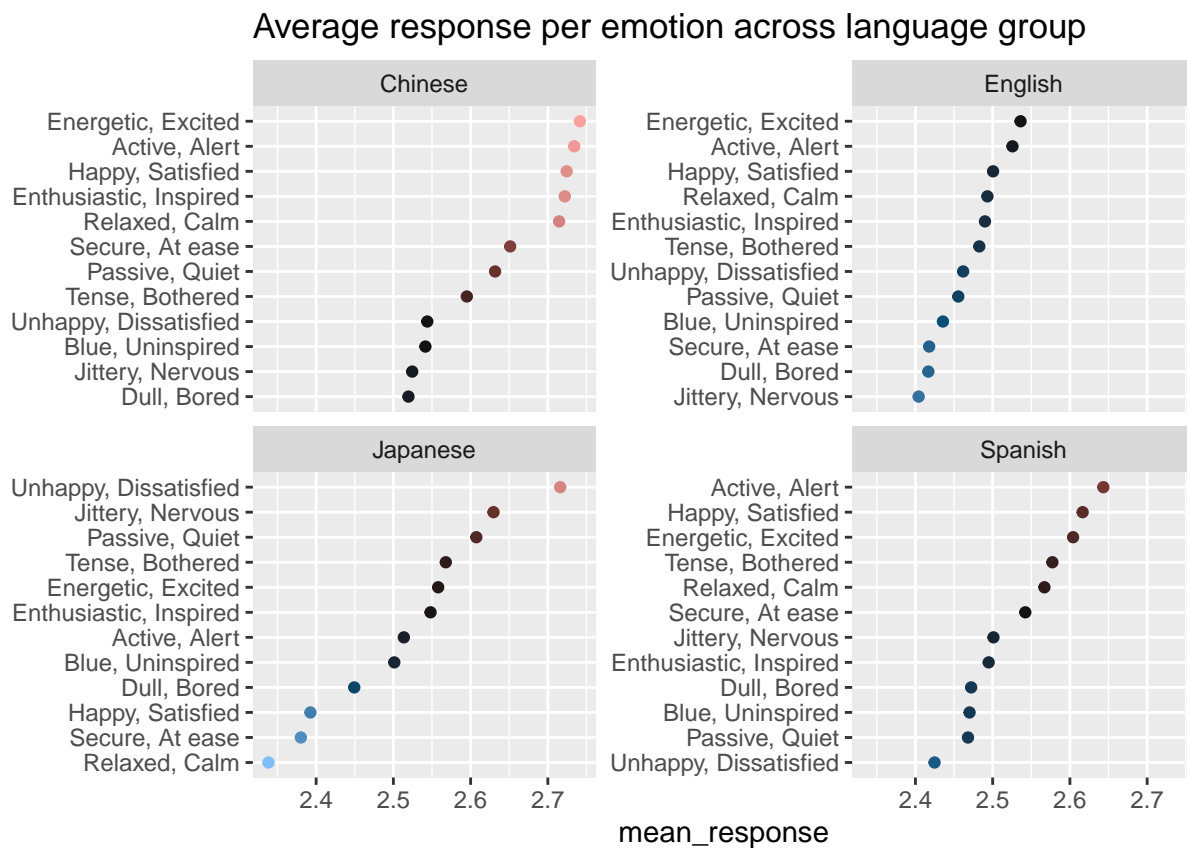
Now that we have a general idea of who the participants are (and I encourage you to explore the other categorical variables in the data in a similar way), we will proceed to the actual experimental data.

Visualizing responses by language and emotion

To better visualize the value of `response` across different categorical variables, we will be using multiple views to split our plot into subplots.

If you are curious about the work that goes into creating palettes, you can watch this conference presentation on the viridis palette.

```
library(tidytext)
emotion_temperature %>%
  group_by(emotion, UserLanguage) %>%
  summarize(mean_response = mean(response)) %>%
  ggplot(aes(y = reorder_within(emotion, mean_response, UserLanguage),
    x = mean_response,
    color = mean_response)) +
  geom_point() +
  facet_wrap(~UserLanguage, scales = "free_y") +
  #scale_color_continuous(type = "viridis") +
  #scale_color_gradient(low = "blue", high = "red") +
  scale_color_gradientn(colours = hcl.colors(5, "Berlin")) +
  scale_y_reordered() +
  theme(legend.position = "none") +
  labs(y = "",
    title = "Average response per emotion across language group")
```



```
emotion_temperature %>%
  filter(gender != "Other") %>%
  group_by(emotion, UserLanguage, gender) %>%
  summarize(mean_response = mean(response)) %>%
  ggplot(aes(y = UserLanguage,
    x = mean_response,
    color = mean_response)) +
  geom_point() +
  facet_grid(emotion~gender, scales = "free_y") +
```

```
#scale_color_continuous(type = "viridis") +  
#scale_color_gradient(low = "blue", high = "red") +  
scale_color_gradientn(colours = hcl.colors(5, "Berlin")) +  
theme(legend.position = "none",  
      strip.text = element_text(size = 5)) +  
labs(y = "",  
      title = "Average response per emotion across language group")
```

Average response per emotion across language group

