Pre-Processing

Case Study 1

November 16, 2024

Setup

Install Packages

```
options(repos = c(CRAN = "https://cran.r-project.org"))


if (!requireNamespace("groundhog", quietly = TRUE)) {
    install.packages("groundhog")
    library("groundhog")
}

pkgs <- c("magrittr", "data.table", "knitr", "stringr", "jsonlite")

groundhog::groundhog.library(pkg = pkgs,
                             date = "2024-10-01")

rm(pkgs)
```

```
t1 <- Sys.time()
```

Data

```
raw    <- data.table::fread(file = "../data/raw/all_apps_wide_2024-08-15.csv")
input  <- data.table::fread(file = "../stimuli/brazil.csv")
# dice   <- data.table::fread(file = "../data/processed/DICE-processed-2024-05-10.csv")
qualtrics  <- data.table::fread(file = "../data/raw/DICE_Brand_Safety_Aug_2024_August+15,+
page_times <- data.table::fread(file = "../data/raw/PageTimes-2024-08-15.csv")
```

```
set.seed(42)
```

Process DICE Data

```
cols <- str_detect(string = names(raw),
                   pattern = "session.code|participant.label|participant.code|likes_data|r

data <- raw[participant._index_in_pages >= 4 & nchar(participant.label) == 24 & participan
rm(cols)

# Preprocessing: rename cols
names(data) %>%
  str_replace_all(pattern = ".*player\\.", replacement = "") %>%
  str_replace_all(pattern = "\\.", replacement = "_") %>%
  str_replace_all(pattern = "feed_", replacement = "") %>%
  str_to_lower() %>%
  setnames(x = data)

# Edit Date Format
data[, participant_time_started_utc := participant_time_started_utc %>% str_sub(start = 1,

# Create Item Sequence DT
display <- data[,
                .(tweet = unlist(base::strsplit(x = sequence,
                                                split = ", ")) %>%
                    as.integer()),
                by = participant_code]

display[, displayed_sequence := 1:.N, by = participant_code]

# Create Flow (Scroll Sequence DT)
```

```
sequence <- data[nchar(viewport_data) > 1,
                 viewport_data %>%
                   str_replace_all(pattern = '"',
                                   replacement = '') %>%
                   fromJSON,
                 by = participant_code][!is.na(doc_id)][, scroll_sequence := 1:.N, by = pa

setnames(sequence, old = 'doc_id', new = 'tweet')

flow <- sequence[display, on = .(participant_code, tweet)]

flow[, duplicate := duplicated(tweet), by = participant_code]

setorder(flow, participant_code, scroll_sequence)

flow_collapsed <- flow[,
                       .(scroll_sequence = paste(scroll_sequence, collapse = ",")),
                       by = .(participant_code, tweet)]

flow_collapsed[scroll_sequence == "NA", scroll_sequence := NA]



# Create Dwell Time DT
viewport <- data[nchar(viewport_data) > 1,
                 fromJSON(str_replace_all(string = viewport_data,
                                          pattern = '"',
                                          replacement = '')),
                 by = participant_code][!is.na(doc_id)]

# -- sum durations by tweet (in case someone scrolled back and forth)
viewport <- viewport[,
                     .(seconds_in_viewport = sum(duration,
                                                 na.rm = TRUE)),
                     by = c('participant_code', 'doc_id')]


# -- rename
setnames(x = viewport,
         old = 'doc_id',
         new = 'tweet')
```

```r
# Create Reactions DT
likes <- data[nchar(likes_data) > 1,
              fromJSON(str_replace_all(string = likes_data,
                                       pattern = '""',
                                       replacement = '"')),
              by = participant_code][!is.na(doc_id)]

if(data[nchar(replies_data) > 3, .N] > 0){
  replies <- data[nchar(replies_data) > 3,
                  fromJSON(str_replace_all(string = replies_data,
                                           pattern = '""',
                                           replacement = '")),
                  by = participant_code][!is.na(doc_id)]
  reactions <- merge(likes, replies, by = c("participant_code", "doc_id"), all = TRUE)

} else {
  reactions <- likes
  reactions[, replies := NA]

}

# make sure doc_id is numeric as is the case for the other data.tables
reactions[, doc_id := as.numeric(doc_id)]

# rename
setnames(x = reactions,
         old = 'doc_id',
         new = 'tweet')


# Create Rowheight DT
rowheight <- data[nchar(rowheight_data) > 1,
                  fromJSON(str_replace_all(string = rowheight_data,
                                           pattern = '""',
                                           replacement = '")),
                  by = participant_code][!is.na(doc_id)]

# rename
setnames(x = rowheight,
         old = 'doc_id',
```

```
         new = 'tweet')


# Merge to Final DT
merge_1 <- merge(data[, .(session_code, participant_code, participant_label, touch_capabil
merge_2 <- merge(merge_1, viewport, by = c("participant_code", "tweet"), all = TRUE)
merge_3 <- merge(merge_2, flow_collapsed, by = c("participant_code", "tweet"), all = TRUE)
merge_4 <- merge(merge_3, reactions, by = c("participant_code", "tweet"), all = TRUE)
tmp     <- merge(merge_4, rowheight, by = c("participant_code", "tweet"), all = TRUE)

# Reorder columns (and rows)
new_order <- c(3, 1, 4, 8, 5, 6, 7)
remaining_cols <- setdiff(1:ncol(tmp), new_order)
dice <- tmp[, c(new_order, remaining_cols), with = FALSE]
setorder(dice, session_code, participant_code, displayed_sequence)

# Re-re-name
setnames(x = dice,
         new = 'doc_id',
         old = 'tweet')

rm(list = c("tmp", "merge_1", "merge_2", "merge_3", "merge_4", "data", "display", "likes",
```

Manipulations

```
setnames(old = "PROLIFIC_PID",
         new = "participant_label",
         x = qualtrics)

qualtrics <- qualtrics[c(-1, -2)]
qualtrics <- qualtrics[Finished == "True"]
```

```
input[, sponsored := as.logical(sponsored)]
```

```
dice[, log_dwell_time := log(seconds_in_viewport)]
dice[, log_dwell_pixel := log_dwell_time / height]
```

```
# female
qualtrics[, female := FALSE]
qualtrics[gender == "Female", female := TRUE]

# age
qualtrics[, age := as.numeric(age)]
```

```
dice[, is_desktop := ifelse(test = device_type == "Desktop", yes = 1, no = 0)]
dice[, device_type := as.factor(device_type)]
```

```
dice[, appropriate := FALSE]
dice[condition == "appropriate", appropriate := TRUE]

input[, appropriate := FALSE]
input[condition == "appropriate", appropriate := TRUE]
```

```
qualtrics[,
          brand_attitude := mean(c(as.numeric(brand_att_1), as.numeric(brand_att_2), as.nu
          by = participant_label]
```

```
qualtrics[, klm_uncued_recall := ifelse(test = str_detect(string = str_to_lower(uncued_rec
                                                          pattern = "klm"),
                                        yes = TRUE,
                                        no = FALSE)]

qualtrics[, klm_cued_recall := ifelse(test = str_detect(string = cued_recall, pattern = "K
                                      yes  = TRUE,
                                      no   = FALSE)]
```

```
times <- page_times[session_code %in% dice[, unique(session_code)] & participant_code %in%

setorderv(x = times, cols = c("session_code", "participant_id_in_session", "page_index"))

times[,
      time_spent_on_page := epoch_time_completed - shift(epoch_time_completed, n = 1, fill
      by = c("session_code", "participant_id_in_session")]
```

Merge Data

Merge Qualtrics and DICE to Output

```
dice_plus <- data.table::merge.data.table(x = dice,
                                          y = times[page_name == "C_Feed",
                                                    .(participant_code, time_spent_on_page
                                          by = "participant_code")
```

```
output <- data.table::merge.data.table(x = dice_plus,
                                       y = qualtrics,
                                       by = "participant_label")[participant_label %in% qu
```

Merge Output and Input

```
long <- data.table::merge.data.table(x = output,
                             y = input,
                             by = c("doc_id", "condition"))

setorder(long, participant_code, displayed_sequence)

long <- long[complete.cases(long[, .SD, .SDcols = 34:56])]
```

Short

```
long[,
     relative_dwell_time := seconds_in_viewport / time_spent_on_page,
     by = participant_label]

# To do: How comes, the following code also returns values smaller than 1? Refreshs? Page
# long[, sum(relative_dwell_time, na.rm = TRUE), by = participant_label]
```

```
tmp <- data.table::merge.data.table(x = dice_plus[,
                                                  .(condition = unique(condition),
                                                    time_spent_on_page = unique(time_spe
                                                by = c("participant_label", "session_code")
```

```
                                        y = qualtrics,
                                        by = "participant_label")

# short <- short[complete.cases(short[, .SD, .SDcols = 21:ncol(short)])]
```

```
short <- data.table::merge.data.table(x = tmp,
                                        y = long[displayed_sequence == 5,
                                                 .(participant_label,
                                                   relative_dwell_time,
                                                   seconds_in_viewport,
                                                   log_dwell_time,
                                                   log_dwell_pixel,
                                                   height,
                                                   liked,
                                                   hasReply,
                                                   is_desktop,
                                                   device_type)],
                                        by = "participant_label")[!duplicated(participant_la
```

```
short[, is_flood_aware := str_detect(string = flood_awareness, pattern = "yes|Yes")]
```

Write Data

```
data.table::fwrite(x = short, file = "../data/processed/brand-safety-short.csv")
data.table::fwrite(x = long, file = "../data/processed/brand-safety-long.csv")
```

Session Info

```
t2 <- Sys.time()
```

The analyses presented in this document required 4.64 seconds, after loading and installing the required packages. *Rendering* the document (i.e., presenting the results in a PDF) required slightly more time (up to one minute). Below, we print the `sessionInfo()` to document the hardware and software used to render this document.

```
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: x86_64-apple-darwin20
Running under: macOS Sonoma 14.4.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Berlin
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] jsonlite_1.8.9    stringr_1.5.1     knitr_1.48        data.table_1.16.0
[5] magrittr_2.0.3

loaded via a namespace (and not attached):
 [1] digest_0.6.37     fastmap_1.2.0     xfun_0.47         groundhog_3.2.1
 [5] glue_1.8.0        parallel_4.4.1    htmltools_0.5.8.1 rmarkdown_2.28
 [9] lifecycle_1.0.4   cli_3.6.3         vctrs_0.6.5       compiler_4.4.1
[13] rstudioapi_0.16.0 tools_4.4.1       evaluate_1.0.0    yaml_2.3.10
[17] rlang_1.1.4       stringi_1.8.4
```