

Pre-Processing

Case Study 2

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",
         "quanteda", "quanteda.textstats")

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

rm(pkgs)

t1 <- Sys.time()
```

Data

```
input <- data.table::fread(file = "../stimuli/9gag.csv")
dice <- data.table::fread(file = "../data/raw/DICE-processed-2024-10-15.csv")
qualtrics <- data.table::fread(file = "../data/raw/DICE_Meme_Feed_October+15,+2024_18.35.c
page_times <- data.table::fread(file = "../data/raw/PageTimes-2024-10-15.csv")
```

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

Manipulations

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

```
qualtrics <- qualtrics[c(-1, -2)]
```

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

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

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

```
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")
```

Merge Output and Input

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

setorder(data, participant_code, displayed_sequence)
```

Processing

```
data[, uncued_whoop := ifelse(test = str_detect(string = str_to_lower(uncued_recall),
                                                pattern = "whoop"),
                             yes = TRUE,
                             no  = FALSE)]

data[, uncued_samsung := ifelse(test = str_detect(string = str_to_lower(uncued_recall),
                                                pattern = "samsung"),
                             yes = TRUE,
                             no  = FALSE)]

data[, uncued_bose := ifelse(test = str_detect(string = str_to_lower(uncued_recall),
                                                pattern = "bose|boise"),
                             yes = TRUE,
                             no  = FALSE)]
```

```

data[, uncued_nintendo := ifelse(test = str_detect(string = str_to_lower(uncued_recall),
                                                pattern = "nintendo|switch"),
                                yes = TRUE,
                                no  = FALSE)]

data[, uncued_apple := ifelse(test = str_detect(string = str_to_lower(uncued_recall),
                                                pattern = "apple"),
                                yes = TRUE,
                                no  = FALSE)]

data[, cued_whoop := ifelse(test = str_detect(string = str_to_lower(cued_recall),
                                                pattern = "whoop"),
                                yes = TRUE,
                                no  = FALSE)]

data[, cued_samsung := ifelse(test = str_detect(string = str_to_lower(cued_recall),
                                                pattern = "samsung"),
                                yes = TRUE,
                                no  = FALSE)]

data[, cued_bose := ifelse(test = str_detect(string = str_to_lower(cued_recall),
                                                pattern = "bose"),
                                yes = TRUE,
                                no  = FALSE)]

data[, cued_nintendo := ifelse(test = str_detect(string = str_to_lower(cued_recall),
                                                pattern = "nintendo"),
                                yes = TRUE,
                                no  = FALSE)]

data[, cued_apple := ifelse(test = str_detect(string = str_to_lower(cued_recall),
                                                pattern = "apple"),
                                yes = TRUE,
                                no  = FALSE)]

data[sponsored == TRUE, recalled_brand_uncued := FALSE]
data[sponsored == TRUE, recalled_brand_cued := FALSE]

data[sponsored == TRUE & username == "Apple" & uncued_apple == TRUE,
     recalled_brand_uncued := TRUE]

```

```

data[sponsored == TRUE & username == "WHOOOP" & uncued_whoop == TRUE,
     recalled_brand_uncued := TRUE]
data[sponsored == TRUE & username == "Bose" & uncued_bose == TRUE,
     recalled_brand_uncued := TRUE]
data[sponsored == TRUE & username == "Nintendo of America" & uncued_nintendo == TRUE,
     recalled_brand_uncued := TRUE]
data[sponsored == TRUE & username == "Samsung US" & uncued_samsung == TRUE,
     recalled_brand_uncued := TRUE]

data[sponsored == TRUE & username == "Apple" & cued_apple == TRUE,
     recalled_brand_cued := TRUE]
data[sponsored == TRUE & username == "WHOOOP" & cued_whoop == TRUE,
     recalled_brand_cued := TRUE]
data[sponsored == TRUE & username == "Bose" & cued_bose == TRUE,
     recalled_brand_cued := TRUE]
data[sponsored == TRUE & username == "Nintendo of America" & cued_nintendo == TRUE,
     recalled_brand_cued := TRUE]
data[sponsored == TRUE & username == "Samsung US" & cued_samsung == TRUE,
     recalled_brand_cued := TRUE]

data[username != "9GAG Memeland", brand := username]
data[username == "Nintendo of America", brand := "Nintendo"]
data[username == "Samsung US", brand := "Samsung"]
data[username == "WHOOOP", brand := "Whoop"]
data[, brand := as.factor(brand)]

data[is.na(liked), liked := FALSE]
data[is.na(hasReply), hasReply := FALSE]

# text length
data[, text_length := nchar(text)]

# gunning fog readability index (as used in Robertson et al. 2022)
corpus <- corpus(data$text)
readability_stats <- quantda.textstats::textstat_readability(data$text, measure = "FOG")
data[, text_complexity := readability_stats$FOG]

# whether participant has liked/replied to any post
data[, has_liked_any := any(liked), by = participant_label]
data[, has_replied_any := any(hasReply), by = participant_label]

```

```
# whether participant has liked/replied to any sponsored post
data[sponsored == 1, has_liked_sponsored := any(liked), by = participant_label]
data[sponsored == 1, has_replied_sponsored := any(hasReply), by = participant_label]

# Device Types
data[, is_desktop := ifelse(test = device_type == "Desktop", yes = 1, no = 0)]
data[, device_type := as.factor(device_type)]

# Demographics
data[, female := ifelse(test = gender == "Female", yes = TRUE, no = FALSE)]
data[, age := as.numeric(age)]

data[, type := as.factor(ifelse(test = sponsored, yes = "Sponsored Posts", no = "Organic P
```

Write Data

```
data.table::fwrite(x = data, file = "../data/processed/meme-feed-data.csv")
```

Session Info

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

The analyses presented in this document required 3.18 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] quanteda.textstats_0.97.2 quanteda_4.1.0
[3] jsonlite_1.8.9 stringr_1.5.1
[5] knitr_1.48 data.table_1.16.0
[7] magrittr_2.0.3

loaded via a namespace (and not attached):

[1] vctrs_0.6.5 cli_3.6.3 rlang_1.1.4 xfun_0.47
[5] stringi_1.8.4 glue_1.8.0 htmltools_0.5.8.1 rmarkdown_2.28
[9] grid_4.4.1 evaluate_1.0.0 nsyllable_1.0.1 fastmap_1.2.0
[13] yaml_2.3.10 lifecycle_1.0.4 compiler_4.4.1 Rcpp_1.0.13
[17] fastmatch_1.1-4 rstudioapi_0.16.0 lattice_0.22-6 digest_0.6.37
[21] groundhog_3.2.1 parallel_4.4.1 stopwords_2.3 Matrix_1.7-0
[25] tools_4.4.1