drm-data-prepare

Jiayi Guo

2025-06-21

Set Up

```
library(readxl)
library(dplyr)
```

```

Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(purrr)
library(stringr)
library(writexl)
library(tidyr)
library(stringr)
```

List all documents

```
# file_paths <- list.files(path = "drm_qualtrics_original_download", pattern =
"\\.xlsx$", full.names = TRUE)

file_paths <- list.files(
  path = "drm_qualtrics_original_download",
  pattern = "\\.xlsx$",
  full.names = TRUE
```

```
) %>%
  discard(~ grepl("/~\\$", .x) || grepl("^~\\$", basename(.x)))
```

Read Files

```
# Step 2: Function to read and standardize SONA column
# will skip the second row (row with questions)
read_and_standardize <- function(file) {
  df <- read_excel(file)

  sona_col <- names(df)[str_detect(names(df), "SONA$")]
  if (length(sona_col) != 1) {
    stop(paste("⚠ Found", length(sona_col), "SONA columns in", basename(file)))
  }

  df <- df %>% rename(SONA = all_of(sona_col))

  # Remove second row (i.e., row index 2)
  df <- df[-1, ]

  return(df)
}
```

```
data_list <- map(file_paths, read_and_standardize)
```

Clearning Duplication

1. Check which files have duplicate SONAs

```
check_duplicates <- function(df, name) {
  df %>%
    count(SONA) %>%
    filter(n > 1) %>%
    mutate(file = name)
}

# Run for all files
duplicates_summary <- map2_dfr(data_list, basename(file_paths),
check_duplicates)

# View summary of SONA duplicates across files
duplicates_summary
```

```
# A tibble: 14 × 3
   SONA      n file
```

```
   <chr> <int> <chr>
 1 71665     2 DRM_Day 1_Packet A1_Cornell_250204_June 21, 2025_19.05.xlsx
 2 88723     2 DRM_Day 1_Packet A1_Cornell_250204_June 21, 2025_19.05.xlsx
 3 70645     2 DRM_Day 4_Packet A3_Cornell_250308_June 21, 2025_19.09.xlsx
 4 88153     2 DRM_Day 4_Packet A3_Cornell_250308_June 21, 2025_19.09.xlsx
 5 90436     2 DRM_Day 4_Packet A3_Cornell_250308_June 21, 2025_19.09.xlsx
 6 90934     2 DRM_Day 4_Packet A3_Cornell_250308_June 21, 2025_19.09.xlsx
 7 91489     2 DRM_Day 4_Packet A3_Cornell_250308_June 21, 2025_19.09.xlsx
 8 70258     2 DRM_Day 5_Packet C2_Cornell_250204_June 21, 2025_19.11.xlsx
 9 86614     2 DRM_Day 5_Packet C2_Cornell_250204_June 21, 2025_19.11.xlsx
10 86776     2 DRM_Day 5_Packet C2_Cornell_250204_June 21, 2025_19.11.xlsx
11 88153     2 DRM_Day 5_Packet C2_Cornell_250204_June 21, 2025_19.11.xlsx
12 92218     2 DRM_Day 5_Packet C2_Cornell_250204_June 21, 2025_19.11.xlsx
13 92173     2 DRM_Day 7_Packet C3_Cornell_250204_June 21, 2025_19.12.xlsx
14 92428     2 DRM_Day 7_Packet C3_Cornell_250204_June 21, 2025_19.12.xlsx
```

2. Only keep the row of entry of each SONA with the most complete entry (fewer NA values)

```
# Create a named list: file -> vector of duplicated SONA IDs
dup_sona_list <- split(duplicates_summary$SONA, duplicates_summary$file)

# Create a new cleaned list
data_list_cleaned <- map2(data_list, basename(file_paths), function(df, fname)
{
  if (fname %in% names(dup_sona_list)) {
    dup_ids <- dup_sona_list[[fname]]

    df_deduped <- df %>%
      rowwise() %>%
      mutate(non_missing_count = sum(!is.na(c_across(everything())))) %>%
      group_by(SONA) %>%
      # For duplicated SONAs, keep the row with most info; for others, keep all
      mutate(dup_flag = SONA %in% dup_ids) %>%
        filter(!dup_flag | (dup_flag & rank(-non_missing_count, ties.method =
"first") == 1)) %>%
      ungroup() %>%
      select(-non_missing_count, -dup_flag)

    return(df_deduped)
  } else {
    return(df)  # no duplication, return as-is
  }
})
```

Inner Join to combine all files by 1ASONA ID

```r
# Step 4: Join one-by-one using full_join
combined_data <- data_list_cleaned[[1]]
for (i in 2:length(data_list_cleaned)) {
  message("Joining file ", i, ": ", file_paths[i])
  combined_data <- full_join(combined_data, data_list_cleaned[[i]], by = "SONA")
}
```

```
Joining file 2: drm_qualtrics_original_download/DRM_Day 3_Packet
C1_Cornell_250204_June 21, 2025_19.08.xlsx
```

```
Joining file 3: drm_qualtrics_original_download/DRM_Day 4_Packet
A3_Cornell_250308_June 21, 2025_19.09.xlsx
```

```
Joining file 4: drm_qualtrics_original_download/DRM_Day 5_Packet
C2_Cornell_250204_June 21, 2025_19.11.xlsx
```

```
Joining file 5: drm_qualtrics_original_download/DRM_Day 6_Packet
A4_Cornell_250308_June 21, 2025_19.10.xlsx
```

```
Joining file 6: drm_qualtrics_original_download/DRM_Day 7_Packet
C3_Cornell_250204_June 21, 2025_19.12.xlsx
```

```
Joining file 7: drm_qualtrics_original_download/DRM_Day 8_Packet
D_Cornell_250204_June 21, 2025_19.12.xlsx
```

Save Combined Data

```r
write_xlsx(combined_data, path = "drm_qualtrics_combined.xlsx")
```

remove quantitative columns

```r
#qualitative data columns
qual_columns <- c("SONA",
                  "C1MNAM",
                  "C1MBEG",
                  "C1MEND",
                  "C1MDO",
                  "C1MDM",
                  "C1MCO",
                  "C1ADO",
                  "C1ADM",
```

```
                "C1ACO",
                "C1AFEEL",
                "C1AELSE",
                "C1ENAM",
                "C1EBEG",
                "C1EBEG",
                "C1EEND",
                "C1EDC",
                "C1EDO",
                "C1EDM",
                "C1ECO",
                "C1EFEEL",
                "C1EELSE",
                "C2MNAM",
                "C2MBEG",
                "C2MEND",
                "C2MDO",
                "C2MDM",
                "C2MCO",
                "C2ADO",
                "C2ADM",
                "C2ACO",
                "C2AFEEL",
                "C2AELSE",
                "C2ENAM",
                "C2EBEG",
                "C2EBEG",
                "C2EEND",
                "C2EDC",
                "C2EDO",
                "C2EDM",
                "C2ECO",
                "C2EFEEL",
                "C2EELSE",
                "C3MNAM",
                "C3MBEG",
                "C3MEND",
                "C3MDO",
                "C3MDM",
                "C3MCO",
                "C3ADO",
                "C3ADM",
                "C3ACO",
                "C3AFEEL",
                "C3AELSE",
                "C3ENAM",
                "C3EBEG",
                "C3EBEG",
```

```
                  "C3EEND",
                  "C3EDC",
                  "C3EDO",
                  "C3EDM",
                  "C3ECO",
                  "C3EFEEL",
                  "C3EELSE",
                  "DNAM",
                  "DBEG",
                  "DEND",
                  "DDO",
                  "DDM",
                  "DCO",
                  "DFEEL",
                  "DELSE",
                  "NH-TRANS",
                  "NH-GROUND",
                  "NH-DRAIN",
                  "FACTS",
                  "FEED"
                  )

# make a spreadsheet for only qualitative data
qualitative_data <- combined_data |>
  select(any_of(qual_columns))
```

Save qualitative data by participant

```
write_xlsx(qualitative_data, path = "drm_qualtrics_qualitative.xlsx")
```

Break off by episodes

```
# episode prefixes
episodes <- c("C1M", "C1A", "C1E", "C2M", "C2A", "C2E", "C3M", "C3A", "C3E",
"D")

# Select only episode-related columns plus SONA
episode_data <- qualitative_data |>
  select(SONA, matches(paste0("^(", paste(episodes, collapse = "|"), ")")))

long_data <- episode_data |>
  pivot_longer(
    cols = -SONA,
    names_to = "temp",
    values_to = "value"
  )
```

```
long_data <- long_data |>
  mutate(
    episode = str_extract(temp, paste(episodes, collapse = "|")),
    variable = str_remove(temp, paste(episodes, collapse = "|"))
  )

episode_long <- long_data |>
  select(-temp) %>%
  pivot_wider(
    names_from = variable,
    values_from = value
  )

episode_long_cleaned <- episode_long |>
  filter(
    rowSums(!is.na(across(-c(SONA, episode)))) > 0
  )
```

Save qualitative data by episode

```
write_xlsx(qualitative_data, path = "drm_qualitative_episodes.xlsx")
```