```
# ============================================================
# Banner table from Excel survey data (text banners preferred)
# - Rows for each item:
#     Unweighted n (whole numbers)
#     Weighted n (1 decimal)
#     % (unweighted) (computed as percent 0–100, rounded to 1 decimal)
#     % (weighted)   (computed as percent 0–100, rounded to 1 decimal)
# - Header row bold
# - Blank row between the two items
# - Blank column between column categories (Total | Platform | Gender | Age | Income)
# - "Total: Total" renamed to "Total"
# - Numeric cells centered
# Output: banner_table.xlsx in your working directory
# From "Banner Tables: Why and How" by James R. Lewis and Jeff Sauro
# Copyright MeasuringU 2026
# ============================================================

#Set working directory
setwd("C:/Users/Jim/Documents/MeasuringU/Benchmarks/2024/Social Media")

# ---- 0) Packages ----
# install.packages(c("openxlsx", "dplyr", "tidyr"))

library(openxlsx)
library(dplyr)
library(tidyr)

# ---- 1) Read Excel ----
file_path  <- "C:/Users/Jim/Documents/MeasuringU/Benchmarks/2024/Social Media/Social Media 2024 Rake Weights
Exercise.xlsx"
sheet_name <- "SocialMedia"

df <- read.xlsx(file_path, sheet = sheet_name)

# ---- 2) Clean / types / text banner variables ----
df2 <- df %>%
```

```
  mutate(
    Weights     = as.numeric(Weights),
    BrandTop2   = as.numeric(BrandTop2),
    PoliticBot2 = as.numeric(PoliticBot2),

    Platform = factor(Platform),   # alphabetical by default
    Gender   = factor(Gender),
    AgeGroup = factor(AgeGroup),

    Income = factor(
      Income,
      levels = c("$0-$24k", "$25k-$49k", "$50k-$99k",
                 "$100k-$149k", "$150k-$199k", "$200k+")
    )
  )

# ---- 3) Helper: base + unweighted & weighted proportion of 1 for a 0/1 variable ----
summ01 <- function(data, y_col) {
  y <- data[[y_col]]
  w <- data$Weights

  # (You said no missing; this still works fine.)
  unw_n <- length(y)
  unw_prop <- mean(y == 1)

  wtd_n <- sum(w)
  wtd_prop <- sum(w * (y == 1)) / wtd_n

  tibble(Unw_n = unw_n, Wtd_n = wtd_n, Unw_prop = unw_prop, Wtd_prop = wtd_prop)
}

# ---- 4) Build one block (Total or a banner variable) ----
make_block <- function(data, banner_col = NULL, banner_name = "Total") {

  calc_two_items <- function(d) {
    bind_cols(
```

```
      summ01(d, "BrandTop2") %>%
        rename(
          BrandTop2_Unw_n     = Unw_n,
          BrandTop2_Wtd_n     = Wtd_n,
          BrandTop2_Unw_prop  = Unw_prop,
          BrandTop2_Wtd_prop  = Wtd_prop
        ),
      summ01(d, "PoliticBot2") %>%
        rename(
          PoliticBot2_Unw_n     = Unw_n,
          PoliticBot2_Wtd_n     = Wtd_n,
          PoliticBot2_Unw_prop  = Unw_prop,
          PoliticBot2_Wtd_prop  = Wtd_prop
        )
    )
  }

  if (is.null(banner_col)) {
    bind_cols(
      tibble(Banner = banner_name, Level = "Total"),
      calc_two_items(data)
    )
  } else {
    data %>%
      group_by(.data[[banner_col]]) %>%
      group_modify(~ calc_two_items(.x)) %>%
      ungroup() %>%
      transmute(
        Banner = banner_name,
        Level  = as.character(.data[[banner_col]]),
        across(where(is.numeric), ~ .x)
      )
  }
}

# ---- 5) Compute all blocks and combine ----
```

```
long <- bind_rows(
  make_block(df2, NULL,        "Total"),
  make_block(df2, "Platform",  "Platform"),
  make_block(df2, "Gender",    "Gender"),
  make_block(df2, "AgeGroup",  "Age"),
  make_block(df2, "Income",    "Income")
) %>%
  mutate(
    # Rename "Total: Total" to just "Total"
    Column = if_else(Banner == "Total", "Total", paste(Banner, Level, sep = ": "))
  )

# ---- 6) Pivot into classic banner table shape ----
banner_table <- long %>%
  select(
    Column,
    BrandTop2_Unw_n, BrandTop2_Wtd_n, BrandTop2_Unw_prop, BrandTop2_Wtd_prop,
    PoliticBot2_Unw_n, PoliticBot2_Wtd_n, PoliticBot2_Unw_prop, PoliticBot2_Wtd_prop
  ) %>%
  pivot_longer(-Column, names_to = "Metric", values_to = "Value") %>%
  pivot_wider(names_from = Column, values_from = Value) %>%
  mutate(
    Metric = recode(Metric,
      BrandTop2_Unw_n     = "Brand attitude (Top-2 Box): Unweighted n",
      BrandTop2_Wtd_n     = "Brand attitude (Top-2 Box): Weighted n",
      BrandTop2_Unw_prop  = "Brand attitude (Top-2 Box): % (unweighted)",
      BrandTop2_Wtd_prop  = "Brand attitude (Top-2 Box): % (weighted)",

      PoliticBot2_Unw_n     = "Political discourse reluctance (Bottom-2 Box): Unweighted n",
      PoliticBot2_Wtd_n     = "Political discourse reluctance (Bottom-2 Box): Weighted n",
      PoliticBot2_Unw_prop  = "Political discourse reluctance (Bottom-2 Box): % (unweighted)",
      PoliticBot2_Wtd_prop  = "Political discourse reluctance (Bottom-2 Box): % (weighted)"
    )
  )

# ---- 7) Insert a blank row between the two items ----
```

```
# BrandTop2 has 4 rows; PoliticBot2 has 4 rows.
blank_row <- banner_table[1, , drop = FALSE]
blank_row[1, ] <- NA
blank_row$Metric <- ""

banner_table2 <- bind_rows(
  banner_table[1:4, ],
  blank_row,
  banner_table[5:nrow(banner_table), ]
)

# ---- 8) Insert blank columns between column categories ----
all_cols <- names(banner_table2)
metric_col <- "Metric"
data_cols  <- setdiff(all_cols, metric_col)

total_cols  <- intersect(data_cols, "Total")

plat_cols <- sort(grep("^Platform: ", data_cols, value = TRUE))

# Use factor level order (NOT alphabetical)
gender_cols <- paste0("Gender: ", levels(df2$Gender))
gender_cols <- gender_cols[gender_cols %in% data_cols]

age_cols <- paste0("Age: ", levels(df2$AgeGroup))
age_cols <- age_cols[age_cols %in% data_cols]

inc_cols <- paste0("Income: ", levels(df2$Income))
inc_cols <- inc_cols[inc_cols %in% data_cols]
final_order <- c(metric_col)

add_group <- function(cols, sep_name) {
  if (length(cols) > 0) {
    final_order <<- c(final_order, cols, sep_name)
  }
}
```

```
# Unique “blank-looking” names for separator columns
sep_names <- c(" ", "  ", "   ", "    ")
sep_i <- 1

add_group(total_cols,  sep_names[sep_i]); sep_i <- sep_i + 1
add_group(plat_cols,   sep_names[sep_i]); sep_i <- sep_i + 1
add_group(gender_cols, sep_names[sep_i]); sep_i <- sep_i + 1
add_group(age_cols,    sep_names[sep_i]); sep_i <- sep_i + 1
final_order <- c(final_order, inc_cols)

# Add separator columns as numeric NA so styling stays simple
for (s in sep_names) {
  if (!(s %in% names(banner_table2))) banner_table2[[s]] <- NA_real_
}

banner_table2 <- banner_table2[, final_order, drop = FALSE]

# ---- 9) Convert proportions to percentages first, then round ----
# Row-type flags
unw_n_rows <- grepl("Unweighted n$", banner_table2$Metric)
wtd_n_rows <- grepl("Weighted n$",   banner_table2$Metric)
pct_rows   <- grepl("% \\(weighted\\)$",   banner_table2$Metric) |
              grepl("% \\(unweighted\\)$", banner_table2$Metric)

# Numeric columns (everything except Metric)
num_cols <- which(sapply(banner_table2, is.numeric))

# Convert % rows from proportion (0–1) to percent (0–100), then round to 1 decimal
if (any(pct_rows) && length(num_cols) > 0) {
  banner_table2[pct_rows, num_cols] <- banner_table2[pct_rows, num_cols] * 100
  banner_table2[pct_rows, num_cols] <- round(banner_table2[pct_rows, num_cols], 1)
}

# Unweighted n as whole numbers
if (any(unw_n_rows) && length(num_cols) > 0) {
```

```
  banner_table2[unw_n_rows, num_cols] <- round(banner_table2[unw_n_rows, num_cols], 0)
}

# Weighted n as 1 decimal
if (any(wtd_n_rows) && length(num_cols) > 0) {
  banner_table2[wtd_n_rows, num_cols] <- round(banner_table2[wtd_n_rows, num_cols], 1)
}

# ---- 10) Write to Excel with formatting ----
out_file <- file.path(getwd(), "banner_table.xlsx")

wb <- createWorkbook()
addWorksheet(wb, "Banner table")
writeData(wb, "Banner table", banner_table2, startRow = 1, startCol = 1)

# Styles
header_style <- createStyle(textDecoration = "bold", halign = "center", valign = "center")
metric_style <- createStyle(halign = "left",   valign = "center")

# Numeric styles (centered)
unw_n_style <- createStyle(halign = "center", valign = "center", numFmt = "0")
wtd_n_style <- createStyle(halign = "center", valign = "center", numFmt = "0.0")
pct_style   <- createStyle(halign = "center", valign = "center", numFmt = "0.0\"%\"")

# Header bold
addStyle(
  wb, "Banner table", header_style,
  rows = 1, cols = 1:ncol(banner_table2),
  gridExpand = TRUE, stack = TRUE
)

# Metric column left aligned
addStyle(
  wb, "Banner table", metric_style,
  rows = 2:(nrow(banner_table2) + 1), cols = 1,
  gridExpand = TRUE, stack = TRUE
```

```
)

# Apply numeric styles by row type (Excel rows are +1 due to header)
excel_rows_unw_n <- which(unw_n_rows) + 1
excel_rows_wtd_n <- which(wtd_n_rows) + 1
excel_rows_pct   <- which(pct_rows) + 1

if (length(num_cols) > 0) {
  if (length(excel_rows_unw_n) > 0) {
    addStyle(
      wb, "Banner table", unw_n_style,
      rows = excel_rows_unw_n, cols = num_cols,
      gridExpand = TRUE, stack = TRUE
    )
  }
  if (length(excel_rows_wtd_n) > 0) {
    addStyle(
      wb, "Banner table", wtd_n_style,
      rows = excel_rows_wtd_n, cols = num_cols,
      gridExpand = TRUE, stack = TRUE
    )
  }
  if (length(excel_rows_pct) > 0) {
    addStyle(
      wb, "Banner table", pct_style,
      rows = excel_rows_pct, cols = num_cols,
      gridExpand = TRUE, stack = TRUE
    )
  }
}

# Column widths
setColWidths(wb, "Banner table", cols = 1, widths = 55)
setColWidths(wb, "Banner table", cols = 2:ncol(banner_table2), widths = 14)

saveWorkbook(wb, out_file, overwrite = TRUE)
```

```
out_file
```