Week 3

Pedro Henrique Brant

2025-07-25

This report explores potential exposure variables related to dietary patterns in the NHANES dataset. All of the variables are categorical.

We begin by loading the necessary libraries:

```
library(tidyverse)
library(here)
library(janitor)
library(patchwork)
library(rlang)
library(gt)
library(forcats)
library(labelled)
```

Next, we read in the cleaned and merged dataset, which combines the variables chosen by Professor Fregni and the ones chosen by the students.

```
df <- readRDS(here("Output", "merged_dataset_fregni_plus_students.rds")) %>%
  clean_names()
```

To make our dietary variables more readable and meaningful, we recode them from numeric codes to labeled factors. This includes whether the person is currently on a diet (`drqsdiet`) and several types of diets such as weight loss, low fat, low salt, etc.

```
recode_diet_variables <- function(df) {
  # Save labels
  saved_labels <- list(
    currently_on_diet      = var_label(df$drqsdiet),
    weight_loss_diet       = var_label(df$drqsdt1),
    low_fat_diet           = var_label(df$drqsdt2),
    low_salt_diet          = var_label(df$drqsdt3),
    low_sugar_diet         = var_label(df$drqsdt4),
    low_fiber_diet         = var_label(df$drqsdt5),
    high_fiber_diet        = var_label(df$drqsdt6),
    diabetic_diet          = var_label(df$drqsdt7),
    weight_gain_diet       = var_label(df$drqsdt8),
    low_carb_diet          = var_label(df$drqsdt9),
    high_protein_diet      = var_label(df$drqsdt10),
    gluten_free_diet       = var_label(df$drqsdt11),
    renal_kidney_diet      = var_label(df$drqsdt12),
    other_special_diet     = var_label(df$drqsdt91)
```

```
  )

  # Rename variables
  df <- df %>%
    rename(
      currently_on_diet     = drqsdiet,
      weight_loss_diet      = drqsdt1,
      low_fat_diet          = drqsdt2,
      low_salt_diet         = drqsdt3,
      low_sugar_diet        = drqsdt4,
      low_fiber_diet        = drqsdt5,
      high_fiber_diet       = drqsdt6,
      diabetic_diet         = drqsdt7,
      weight_gain_diet      = drqsdt8,
      low_carb_diet         = drqsdt9,
      high_protein_diet     = drqsdt10,
      gluten_free_diet      = drqsdt11,
      renal_kidney_diet     = drqsdt12,
      other_special_diet    = drqsdt91
    )

  # Recode values
  df <- df %>%
    mutate(
      currently_on_diet    = recode_factor(currently_on_diet, `1` = "Yes", `2` = "No", `9` = "Don't kno
                             fct_explicit_na(na_level = "Missing"),
      weight_loss_diet     = recode_factor(weight_loss_diet, `1` = "Weight loss/Low calorie diet"),
      low_fat_diet         = recode_factor(low_fat_diet, `2` = "Low fat/Low cholesterol diet"),
      low_salt_diet        = recode_factor(low_salt_diet, `3` = "Low salt/Low sodium diet"),
      low_sugar_diet       = recode_factor(low_sugar_diet, `4` = "Sugar free/Low sugar diet"),
      low_fiber_diet       = recode_factor(low_fiber_diet, `5` = "Low fiber diet"),
      high_fiber_diet      = recode_factor(high_fiber_diet, `6` = "High fiber diet"),
      diabetic_diet        = recode_factor(diabetic_diet, `7` = "Diabetic diet"),
      weight_gain_diet     = recode_factor(weight_gain_diet, `8` = "Weight gain/Muscle building diet"),
      low_carb_diet        = recode_factor(low_carb_diet, `9` = "Low carbohydrate diet"),
      high_protein_diet    = recode_factor(high_protein_diet, `10` = "High protein diet"),
      gluten_free_diet     = recode_factor(gluten_free_diet, `11` = "Gluten-free/Celiac diet"),
      renal_kidney_diet    = recode_factor(renal_kidney_diet, `12` = "Renal/Kidney diet"),
      other_special_diet   = recode_factor(other_special_diet, `91` = "Other special diet")
    )

  # Restore labels
  for (var in names(saved_labels)) {
    var_label(df[[var]]) <- saved_labels[[var]]
  }

  return(df)
}

df <- recode_diet_variables(df)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `currently_on_diet = `%>%`(...)`.
```

```
## Caused by warning:
## ! `fct_explicit_na()` was deprecated in forcats 1.0.0.
## i Please use `fct_na_value_to_level()` instead.
```

To evaluate internal consistency, we created a derived variable `any_specific_diet_flag` that indicates whether participants reported following at least one specific diet. We then cross-tabulated this with the general question on whether the participant was currently on any diet. This allows us to identify potential discrepancies—such as individuals who reported a specific dietary pattern but did not indicate they were currently on a diet. The table below summarizes all combinations of these two variables:

```
df <- df %>%
  mutate(
    # Create a new factor variable indicating if any specific diet is followed (excluding 'currently_on
    any_specific_diet_flag = factor(
      if_else(
        # Count non-NA values across all *_diet variables, excluding 'currently_on_diet'
        rowSums(across(
          ends_with("_diet") & !matches("^currently_on_diet$"),
          ~ !is.na(.)
        )) > 0,
        "Yes",  # If any non-NA diet is present
        "No"    # If all are NA
      ),
      levels = c("Yes", "No")  # Set factor levels
    )
  )

summary_tab <- df %>%
  count(currently_on_diet, any_specific_diet_flag, name = "Count") %>%
  mutate(
    Percent = round(100 * Count / sum(Count), 1)
  ) %>%
  complete(
    currently_on_diet,
    any_specific_diet_flag,
    fill = list(Count = 0, Percent = 0)
  ) %>%
  rename(
    `Currently on diet` = currently_on_diet,
    `Following any specific diet` = any_specific_diet_flag
  ) %>%
  gt() %>%
  tab_header(
    title = md("**Consistency Between Diet Flags**"),
    subtitle = "Are there any patients flagged to be on a specific diet that didn't show up as being cu
  ) %>%
  cols_align(align = "center", columns = everything()) %>%
  opt_row_striping()

summary_tab
```

To begin, we explore the general question about whether participants are currently following a special diet. The variable `currently_on_diet` captures this information using labeled categorical responses. The plot

Consistency Between Diet Flags

Are there any patients flagged to be on a specific diet that didn't show up as being currently on a diet?

Currently on diet	Following any specific diet	Count	Percent
Yes	Yes	905	9.8
Yes	No	0	0.0
No	Yes	0	0.0
No	No	6773	73.2
Don't know	Yes	0	0.0
Don't know	No	50	0.5
Missing	Yes	0	0.0
Missing	No	1526	16.5

How Participants Responded to 'Are You Currently on a Diet?'

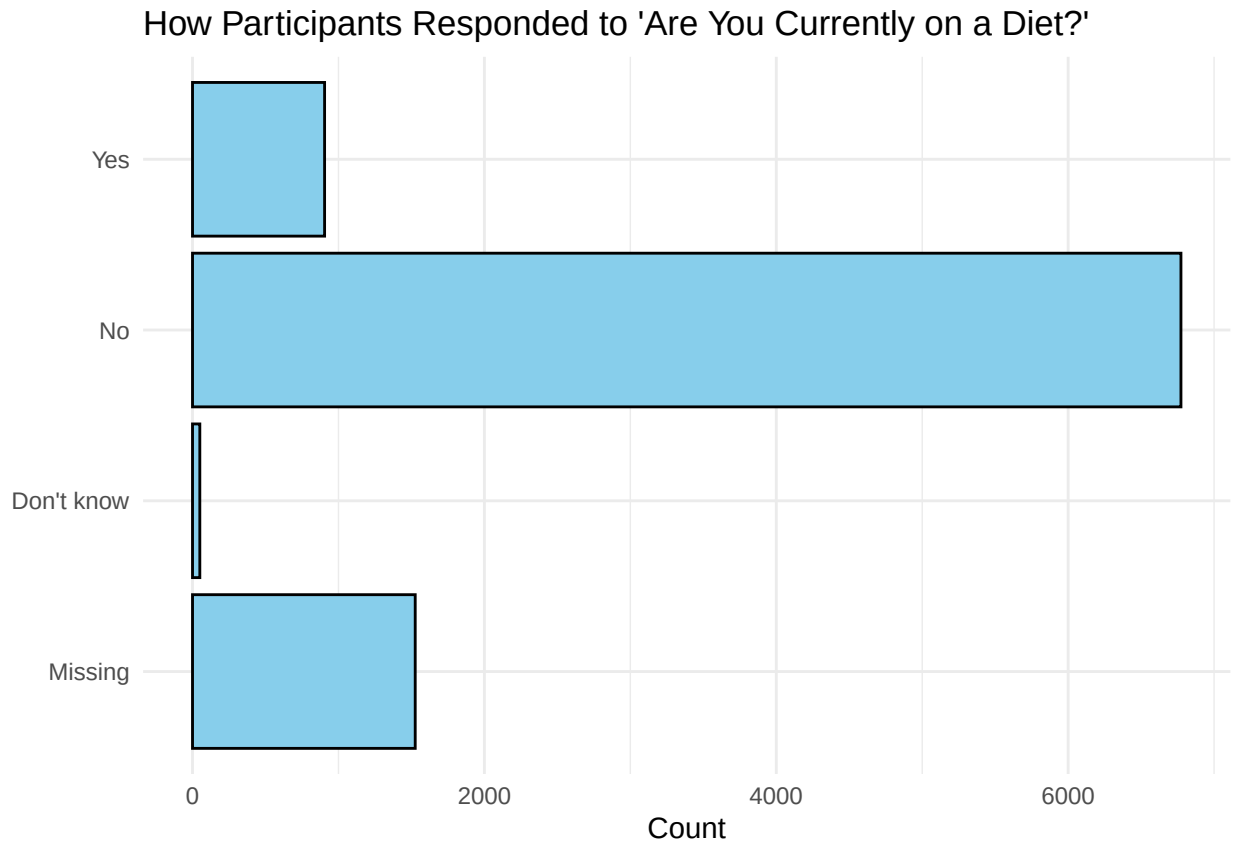
Response	Frequency	Percent (%)
Yes	905	9.8
No	6773	73.2
Don't know	50	0.5
Missing	1526	16.5

and table below display the distribution of responses, including those who responded "Yes", "No", "Don't know", or left the question unanswered.

```
# Frequency table and plot for the general diet question

diet_freq_tbl <- df %>%
  count(currently_on_diet, name = "n") %>%
  mutate(
    percent = round(100 * n / sum(n), 1)
  ) %>%
  gt() %>%
  cols_label(
    currently_on_diet = "Response",
    n = "Frequency",
    percent = "Percent (%)"
  ) %>%
  tab_header(
    title = "How Participants Responded to 'Are You Currently on a Diet?'"
  ) %>%
  fmt_number(columns = percent, decimals = 1) %>%
  cols_align(align = "center", columns = everything())

diet_freq_tbl
```

```
# Bar Plot
barplot_gen_variable <- df %>%
  count(currently_on_diet, name = "count") %>%
```

```
  mutate(
    percent = round(100 * count / sum(count), 2)
  ) %>%
  ggplot(aes(x = fct_rev(fct_infreq(currently_on_diet)), y = count)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(
    title = "How Participants Responded to 'Are You Currently on a Diet?'",
    x = NULL,
    y = "Count"
  ) +
  theme_minimal() +
  coord_flip()

barplot_gen_variable
```



To explore the distribution of special diets among respondents, we first calculate the total number of individuals currently following a diet, followed by a breakdown of the specific types of diets they report. We also summarize responses from those not currently on a diet, those who answered "Don't know," and those with missing data. The resulting table presents both absolute and relative frequencies, formatted for publication using the `gt` package. Specific diets are displayed as indented, italicized subcategories beneath the "Yes" group for clarity.

```
# 1) Totals
# Calculate the number of respondents currently on a special diet ("Yes")
n_yes   <- df %>% filter(currently_on_diet == "Yes") %>% nrow()
# Calculate the total number of respondents
```

```r
n_total <- nrow(df)

# 2) "Yes" summary row
# Create a row summarizing the total frequency and percentage of "Yes" responses
yes_row <- tibble(
  label       = "Yes",
  Frequency   = n_yes,
  `Percent (%)` = round(100 * n_yes / n_total, 1)
)

# 3) Specific diets under "Yes"
# Extract and count specific diet types among those who answered "Yes"
yes_diets <- df %>%
  filter(currently_on_diet == "Yes") %>%
  pivot_longer(
    cols      = ends_with("_diet") & !matches("^currently_on_diet$"),
    names_to  = "var", values_to = "diet"
  ) %>%
  filter(!is.na(diet)) %>%
  count(diet, name = "Frequency") %>%
  arrange(desc(Frequency)) %>%
  mutate(
    label         = paste0("• ", diet),  # Add bullet to distinguish subcategories
    `Percent (%)` = round(100 * Frequency / n_total, 1)  # Percent out of total
  ) %>%
  select(label, Frequency, `Percent (%)`)

# 4) Other categories
# Count and format the "No", "Don't know", and "Missing" responses
others <- df %>%
  filter(currently_on_diet != "Yes") %>%
  count(currently_on_diet, name = "Frequency") %>%
  mutate(
    label         = as.character(currently_on_diet),
    `Percent (%)` = round(100 * Frequency / n_total, 1)
  ) %>%
  select(label, Frequency, `Percent (%)`)

# 5) Combine
# Bind the summary row, specific diets, and other categories into one table
final_tbl <- bind_rows(yes_row, yes_diets, others)

# 6) Render with gt and style the bullets
# Create gt table with proper header, alignment, and styling
final_tbl %>%
  gt(rowname_col = "label") %>%
  tab_stubhead(label = "Following a diet?") %>%
  tab_header(title = "Absolute and relative frequencies of specific diets") %>%
  cols_align(align = "center", columns = c(Frequency, `Percent (%)`)) %>%
  # Indent and italicize only the bullet-labeled diet types
  tab_style(
    style = cell_text(style = "italic", indent = px(15), size = px(12)),
    locations = cells_stub(rows = startsWith(final_tbl$label, "• "))
```

Absolute and relative frequencies of specific diets

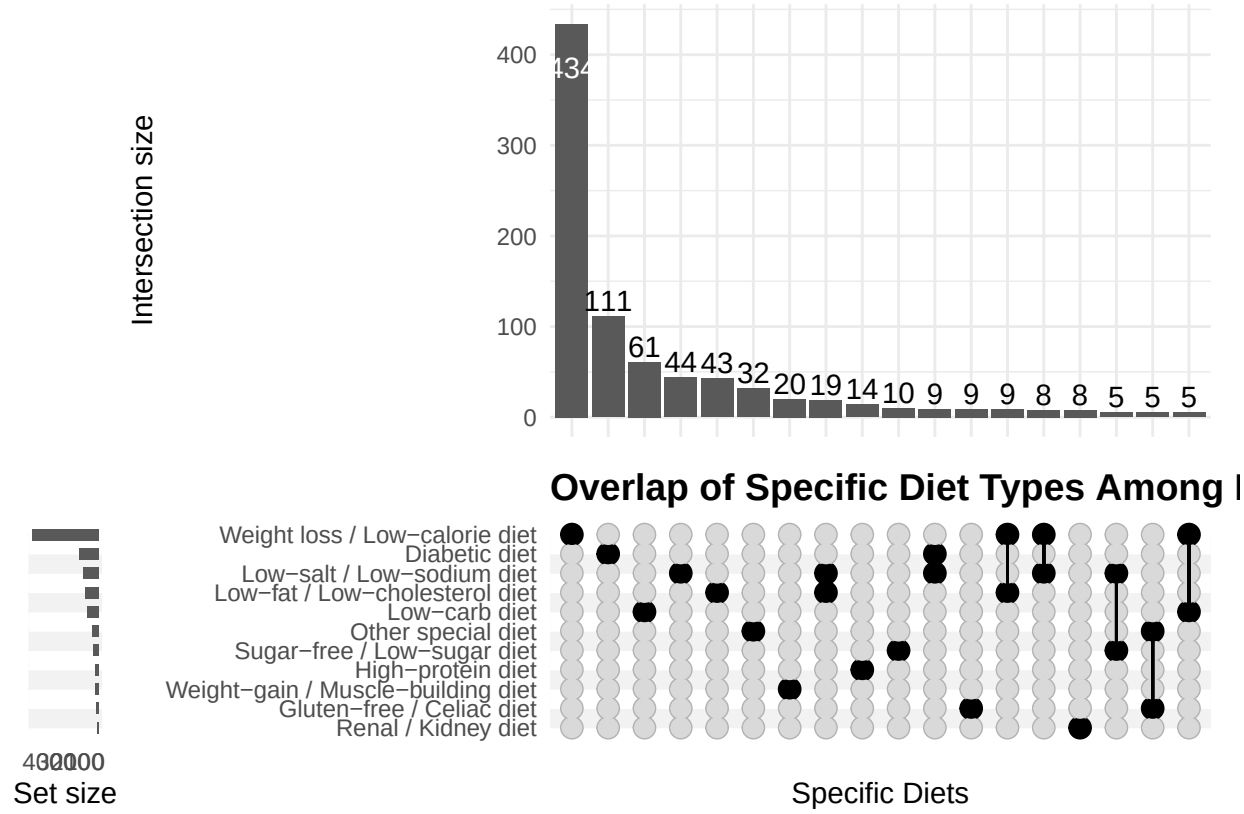
Following a diet?	Frequency	Percent (%)
Yes	905	9.8
• *Weight loss/Low calorie diet*	475	5.1
• *Diabetic diet*	139	1.5
• *Low salt/Low sodium diet*	109	1.2
• *Low fat/Low cholesterol diet*	95	1.0
• *Low carbohydrate diet*	85	0.9
• *Other special diet*	44	0.5
• *Sugar free/Low sugar diet*	40	0.4
• *High protein diet*	26	0.3
• *Weight gain/Muscle building diet*	23	0.2
• *Gluten-free/Celiac diet*	16	0.2
• *Renal/Kidney diet*	10	0.1
• *High fiber diet*	4	0.0
• *Low fiber diet*	2	0.0
No	6773	73.2
Don't know	50	0.5
Missing	1526	16.5

```
  ) %>%
  # Reduce font size in data cells of bullet-labeled rows
  tab_style(
    style = cell_text(size = px(12)),
    locations = cells_body(
      columns = c(Frequency, `Percent (%)`),
      rows = startsWith(final_tbl$label, "• ")
    )
  )
```

To visualize the overlap and combinations of different specific diets among participants who reported following a special diet, we used an UpSet plot. This type of plot offers a clear summary of how frequently participants reported one or more particular dietary patterns, as well as which combinations are most common.

```
library(ComplexUpset)

# 1. Prepare the data (as before)
upset_data <- df %>%
  filter(currently_on_diet == "Yes") %>%
  mutate(across(
    ends_with("_diet") & !matches("^currently_on_diet$"),
    ~ !is.na(.)
  ))

specific_diets <- upset_data %>%
  select(ends_with("_diet") & !matches("^currently_on_diet$")) %>%
  names()
```

```r
diet_labels <- c(
  weight_loss_diet      = "Weight loss / Low-calorie diet",
  low_fat_diet          = "Low-fat / Low-cholesterol diet",
  low_salt_diet         = "Low-salt / Low-sodium diet",
  low_sugar_diet        = "Sugar-free / Low-sugar diet",
  low_fiber_diet        = "Low-fiber diet",
  high_fiber_diet       = "High-fiber diet",
  diabetic_diet         = "Diabetic diet",
  weight_gain_diet      = "Weight-gain / Muscle-building diet",
  low_carb_diet         = "Low-carb diet",
  high_protein_diet     = "High-protein diet",
  gluten_free_diet      = "Gluten-free / Celiac diet",
  renal_kidney_diet     = "Renal / Kidney diet",
  other_special_diet    = "Other special diet"
)

# 2. Rename only the specific diet columns
upset_data_pub <- upset_data %>%
  rename_with(~ diet_labels[.x], .cols = all_of(specific_diets))

# 3. Now select ONLY the renamed columns (in order) for plotting
specific_diets_pub <- unname(diet_labels[specific_diets])  # vector of publication-ready names

# 4. UpSet plot using *only* the correct columns as sets
upset(
  upset_data_pub,
  specific_diets_pub,
  name = "Specific Diets",
  min_size = 5,
  width_ratio = 0.1
) +
  theme(
    axis.text.x = element_blank(),
    plot.title   = element_text(size = 14, face = "bold")
  ) +
  ggtitle("Overlap of Specific Diet Types Among Dieting Participants")
```



This plot shows the most common individual diets on the left, and the set intersections (i.e. combinations of diets) along the bottom, with bar heights indicating the number of participants in each intersection.