Composite scores

2024-08-15

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(tidyr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## 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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

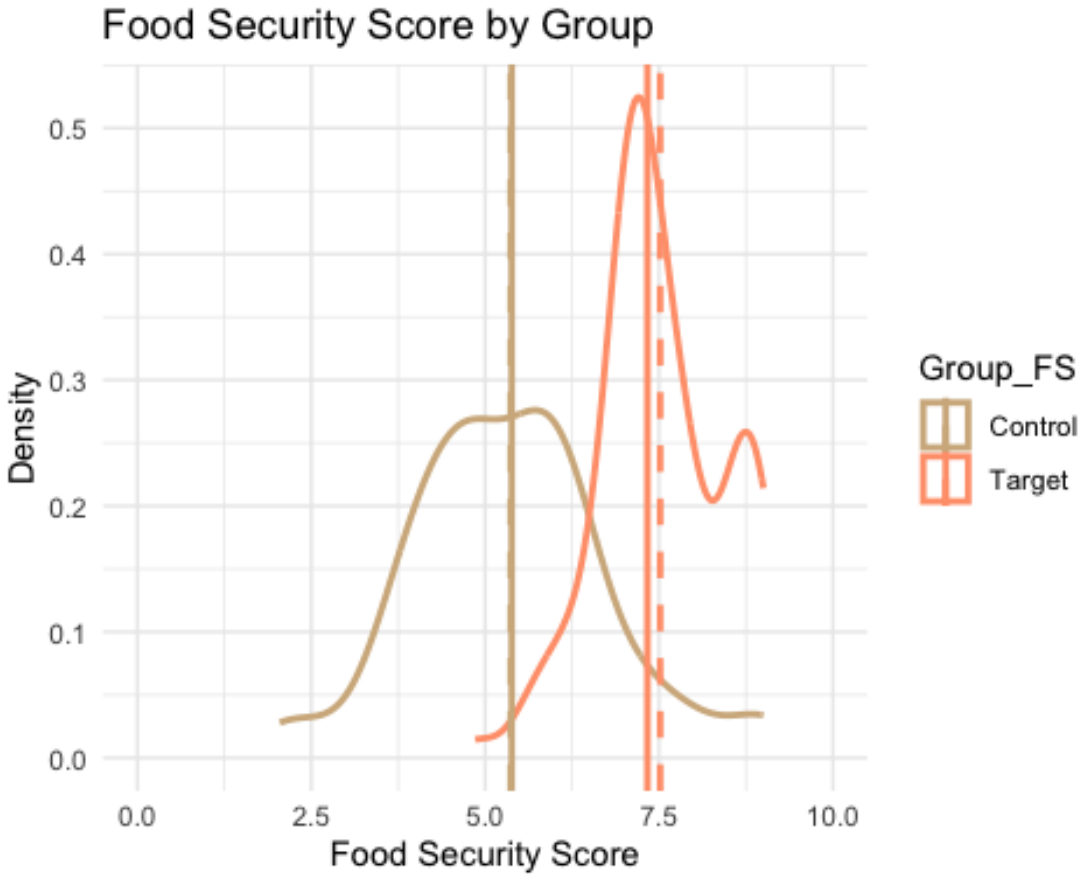
```
library(gt)

data1 <- read_excel("/Users/hamid/Documents/UC Berkeley/URAP/Garden
Questionnaire Dataset/GardenQues/Separate.xlsx")
```

```
## New names:
## • `` -> `...413`
## • `` -> `...414`
## • `` -> `...415`
## • `` -> `...416`
```

```
# Using binary columns to generate a score
month_columns <- data1[, c("16. January", "16. February", "16. March", "16.
April",
                           "16. May", "16. June", "16. July", "16. August",
                           "16. September", "16. October", "16. November",
"16. December")]

# Compute the Lean_Month_Count
data1$Lean_Month_Count <- rowSums(month_columns, na.rm = TRUE)

# Compute the Lean_Month_Score
```

```r
data1$Lean_Month_Score <- ifelse(data1$`No Lean Months` == 1, 12, 12 -
data1$Lean_Month_Count)

# Create the Food_Security_score
data1 <- data1 %>%
  mutate(
    Food_Security_score = (Lean_Month_Score / 3) +
      (`HFIAS Inverse Score` * 4 / 21) +
      ifelse(`30. Do you grow vegetables?` == "Yes", 1, 0)
  )

control_group <- data1 %>% filter(`Household # (Code)` < 200)
target_group <- data1 %>% filter(`Household # (Code)` >= 200)
# Define the control and target groups
control_group_fs <- data1 %>% filter(`Household # (Code)` < 200)
target_group_fs <- data1 %>% filter(`Household # (Code)` >= 200)

# Add group labels to the dataset
control_group_fs <- control_group_fs %>% mutate(Group_FS = "Control")
target_group_fs <- target_group_fs %>% mutate(Group_FS = "Target")

# Combine the datasets
combined_data_fs <- bind_rows(control_group_fs, target_group_fs)

# Plot the combined density plot with mean and median lines
ggplot(combined_data_fs, aes(x = Food_Security_score, color = Group_FS)) +
  geom_density(size = 1, trim=TRUE) +
  scale_color_manual(values = c("Control" = "#D2B48C", "Target" = "#FFA07A"))
+  # Consistent colors
  labs(title = "Food Security Score by Group",
       x = "Food Security Score",
       y = "Density") +
  theme_minimal() +
  geom_vline(aes(xintercept = mean(Food_Security_score[Group_FS ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(Food_Security_score[Group_FS ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "solid", size = 1) +
  geom_vline(aes(xintercept = mean(Food_Security_score[Group_FS == "Target"],
na.rm = TRUE),
                 color = "Target"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(Food_Security_score[Group_FS ==
"Target"], na.rm = TRUE),
                 color = "Target"), linetype = "solid", size = 1) +
  scale_x_continuous(limits = c(0, 10))

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
# Function to calculate quartiles and other statistics
calculate_quartiles <- function(data, variable) {
  data %>%
    summarise(
      Min = min(!!sym(variable), na.rm = TRUE),
      Lower_Quartile = quantile(!!sym(variable), 0.25, na.rm = TRUE),
      Median = median(!!sym(variable), na.rm = TRUE),
      Third_Quartile = quantile(!!sym(variable), 0.75, na.rm = TRUE),
      Max = max(!!sym(variable), na.rm = TRUE),
      Mean = mean(!!sym(variable), na.rm = TRUE),
      SD = sd(!!sym(variable), na.rm = TRUE)
    )
}

# Calculate statistics for total group
total_stats <- calculate_quartiles(data1, "Food_Security_score") %>%
  mutate(Group = "Total")

# Calculate statistics for control group
control_stats <- calculate_quartiles(control_group_fs, "Food_Security_score")
```

```
%>%
  mutate(Group = "Control")

# Calculate statistics for target group
target_stats <- calculate_quartiles(target_group_fs, "Food_Security_score")
%>%
  mutate(Group = "Target")

# Combine all statistics into one table
quartile_table <- bind_rows(total_stats, control_stats, target_stats)

# Reorder columns to have Group first
quartile_table <- quartile_table %>% select(Group, everything())

# Create a well-formatted table using gt
quartile_table_gt <- quartile_table %>%
  gt() %>%
  tab_header(
    title = "Food Security Score Summary Statistics",
    subtitle = "Descriptive statistics for Total, Control, and Target groups"
  ) %>%
  cols_label(
    Group = "Group",
    Min = "Minimum",
    Lower_Quartile = "Lower Quartile",
    Median = "Median",
    Third_Quartile = "Upper Quartile",
    Max = "Maximum",
    Mean = "Mean",
    SD = "Standard Deviation"
  ) %>%
  fmt_number(
    columns = vars(Min, Lower_Quartile, Median, Third_Quartile, Max, Mean,
SD),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
      weight = px(1)
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
```

```
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
    heading.align = "center"
  )
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
```

```
# Print the formatted gt table
quartile_table_gt
```

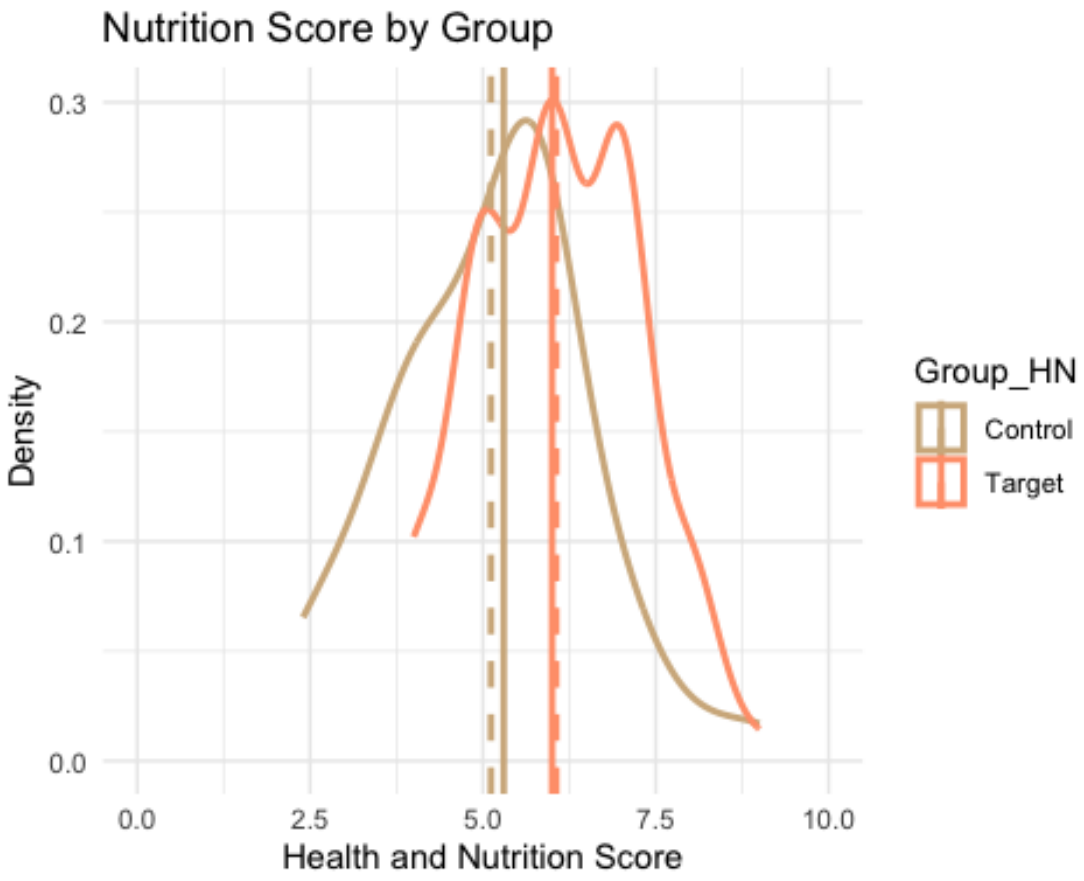
Table 1: Food Security Score Summary Statistics

Descriptive statistics for Total, Control, and Target groups

Group	Minimum	Lower Quartile	Median	Upper Quartile	Maximum	Mean	Standard Deviation
Total	2.05	5.93	7.10	7.76	9.00	6.80	1.49
Control	2.05	4.45	5.38	6.13	9.00	5.37	1.43
Target	4.86	6.95	7.33	8.07	9.00	7.52	0.89

```
#health and nutrition score generation
data1 <- data1 %>%
  mutate(
    Health_and_nutrition_score = (`MDDW Score` * 4 / 5) *
      ifelse(`15. How many meals a day does the family usually eat?` == "2
Meals", 1,
             ifelse(`15. How many meals a day does the family usually eat?`
%in% c("3 Meals", "4 Meals"), 1.25, NA))
  )

# Define the control and target groups
control_group_hn <- data1 %>% filter(`Household # (Code)` < 200)
target_group_hn <- data1 %>% filter(`Household # (Code)` >= 200)

# Add group labels to the dataset
control_group_hn <- control_group_hn %>% mutate(Group_HN = "Control")
target_group_hn <- target_group_hn %>% mutate(Group_HN = "Target")

# Combine the datasets
combined_data_hn <- bind_rows(control_group_hn, target_group_hn)

# Plot the combined density plot with mean and median lines
ggplot(combined_data_hn, aes(x = Health_and_nutrition_score, color =
```

```
Group_HN)) +
  geom_density(size = 1, trim=TRUE) +
  scale_color_manual(values = c("Control" = "#D2B48C", "Target" = "#FFA07A"))
+  # Consistent colors
  labs(title = "Nutrition Score by Group",
       x = "Health and Nutrition Score",
       y = "Density") +
  theme_minimal() +
  geom_vline(aes(xintercept = mean(Health_and_nutrition_score[Group_HN ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(Health_and_nutrition_score[Group_HN ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "solid", size = 1) +
  geom_vline(aes(xintercept = mean(Health_and_nutrition_score[Group_HN ==
"Target"], na.rm = TRUE),
                 color = "Target"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(Health_and_nutrition_score[Group_HN ==
"Target"], na.rm = TRUE),
                 color = "Target"), linetype = "solid", size = 1) +
  scale_x_continuous(limits = c(0, 10))
```



```r
# Calculate statistics for total group
total_stats_health_nutrition <- calculate_quartiles(data1,
"Health_and_nutrition_score") %>%
  mutate(Group = "Total")

# Calculate statistics for control group
control_stats_health_nutrition <- calculate_quartiles(control_group_hn,
"Health_and_nutrition_score") %>%
  mutate(Group = "Control")

# Calculate statistics for target group
target_stats_health_nutrition <- calculate_quartiles(target_group_hn,
"Health_and_nutrition_score") %>%
  mutate(Group = "Target")

# Combine all statistics into one table
quartile_table_health_nutrition <- bind_rows(total_stats_health_nutrition,
control_stats_health_nutrition, target_stats_health_nutrition)

# Reorder columns to have Group first
quartile_table_health_nutrition <- quartile_table_health_nutrition %>%
select(Group, everything())

# Create a well-formatted table using gt
quartile_table_health_nutrition_gt <- quartile_table_health_nutrition %>%
  gt() %>%
  tab_header(
    title = "Health and Nutrition Score Summary Statistics",
    subtitle = "Descriptive statistics for Total, Control, and Target groups"
  ) %>%
  cols_label(
    Group = "Group",
    Min = "Minimum",
    Lower_Quartile = "Lower Quartile",
    Median = "Median",
    Third_Quartile = "Upper Quartile",
    Max = "Maximum",
    Mean = "Mean",
    SD = "Standard Deviation"
  ) %>%
  fmt_number(
    columns = vars(Min, Lower_Quartile, Median, Third_Quartile, Max, Mean,
SD),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
```

```
      weight = px(1)
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
    heading.align = "center"
  )
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
```

```
# Print the formatted gt table
quartile_table_health_nutrition_gt
```

Table 2: Health and Nutrition Score Summary Statistics

Descriptive statistics for Total, Control, and Target groups

Group	Minimum	Lower Quartile	Median	Upper Quartile	Maximum	Mean	Standard Deviation
Total	2.40	5.00	6.00	7.00	9.00	5.75	1.32
Control	2.40	4.00	5.30	6.00	9.00	5.12	1.41
Target	4.00	5.00	6.00	7.00	9.00	6.06	1.16

```
# Calculate the resilience_score
data1 <- data1 %>%
  mutate(
    # Part 1: Water sources (multiplied by 2 and divided by 3)
    water_sources_score = (rowSums(select(., `11. River/Creek/Stream`, `11.
Rainwater Harvest`, `11. Spring Water`,
                                          `11. Well Water`, `11.
Borehole/Groundwater`, `11. Tap/Piped`), na.rm = TRUE)/2),

    # Part 2: Household assets (divided by 2)
    assets_score = rowSums(select(., `14. Improved Stove`, `14.
Refrigerator`, `14. Mobile phone`, `14. Smart mobile phone`,
                                  `14. Bicycle`, `14. Motorcycle`, `14.
Radio`, `14. Television`), na.rm = TRUE)*3 / 8,
```

```r
# Part 3: Illness score (8 if "No Illness", otherwise 8 minus sum of
illnesses, divided by 2)
illness_score = ifelse(`No Illness` == 1, 8,
                           (8 - rowSums(select(., `Malaria`, `Respiratory
diseases/cough`, `Diarrheal diseases`, `Tuberculosis`,
                                                `HIV-AIDS`, `Cholera`,
`Chronic (diabetes, heart disease, cancer, hypertension/blood pressure)`,
                                                `Ulcers`), na.rm =
TRUE)))*0.5,

    # Part 4: Income sources (divided by 3)
    income_sources_score = rowSums(select(.,
                                          `65. Sales of vegetables from my
garden`, `65. Farm sales - food`,
                                          `65. Farm sales - commercial`, `65.
Farm sales - animals`,
                                          `65. Small business`, `65. Day
labor wages`), na.rm = TRUE),
    # Part 5: Vegetable practices (divided by 2)
    vegetable_practice_score = rowSums(select(., `Nursery bed preparation`,
`Transplanting to raised beds`, `Use of compost`,
                                              `Use of natural/botanical
pesticide`, `Mulching`, `Companion/intercropping`,
                                              `Seed removal/saving`,
`Rainwater harvesting`), na.rm = TRUE) / 2,

    # Combine all parts to create the resilience score
    resilience_score = water_sources_score + assets_score + illness_score +
income_sources_score + vegetable_practice_score
  )



# Define the control and target groups
control_group_resilience <- data1 %>% filter(`Household # (Code)` < 200)
target_group_resilience <- data1 %>% filter(`Household # (Code)` >= 200)

# Add group labels to the dataset
control_group_resilience <- control_group_resilience %>%
mutate(Group_Resilience = "Control")
target_group_resilience <- target_group_resilience %>%
mutate(Group_Resilience = "Target")

# Combine the datasets
combined_data_resilience <- bind_rows(control_group_resilience,
target_group_resilience)

# Plot the combined density plot with mean and median lines
```

```r
ggplot(combined_data_resilience, aes(x = resilience_score, color =
Group_Resilience)) +
  geom_density(size = 1, trim=TRUE) +
  scale_color_manual(values = c("Control" = "#D2B48C", "Target" = "#FFA07A"))
+  # Consistent colors
  labs(title = "Resilience Score by Group",
       x = "Resilience Score",
       y = "Density") +
  theme_minimal() +
  geom_vline(aes(xintercept = mean(resilience_score[Group_Resilience ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(resilience_score[Group_Resilience ==
"Control"], na.rm = TRUE),
                 color = "Control"), linetype = "solid", size = 1) +
  geom_vline(aes(xintercept = mean(resilience_score[Group_Resilience ==
"Target"], na.rm = TRUE),
                 color = "Target"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(resilience_score[Group_Resilience ==
"Target"], na.rm = TRUE),
                 color = "Target"), linetype = "solid", size = 1) +
  scale_x_continuous(limits = c(0, 20))
```

```r
# Calculate statistics for total group
total_stats_resilience <- calculate_quartiles(data1, "resilience_score") %>%
  mutate(Group = "Total")

# Calculate statistics for control group
control_stats_resilience <- calculate_quartiles(control_group_resilience,
"resilience_score") %>%
  mutate(Group = "Control")

# Calculate statistics for target group
target_stats_resilience <- calculate_quartiles(target_group_resilience,
"resilience_score") %>%
  mutate(Group = "Target")

# Combine all statistics into one table
quartile_table_resilience <- bind_rows(total_stats_resilience,
control_stats_resilience, target_stats_resilience)

# Reorder columns to have Group first
quartile_table_resilience <- quartile_table_resilience %>% select(Group,
everything())

# Create a well-formatted table using gt
quartile_table_resilience_gt <- quartile_table_resilience %>%
  gt() %>%
  tab_header(
    title = "Resilience Score Summary Statistics",
    subtitle = "Descriptive statistics for Total, Control, and Target groups"
  ) %>%
  cols_label(
    Group = "Group",
    Min = "Minimum",
    Lower_Quartile = "Lower Quartile",
    Median = "Median",
    Third_Quartile = "Upper Quartile",
    Max = "Maximum",
    Mean = "Mean",
    SD = "Standard Deviation"
  ) %>%
  fmt_number(
    columns = vars(Min, Lower_Quartile, Median, Third_Quartile, Max, Mean,
SD),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
      weight = px(1)
```

```
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
    heading.align = "center"
  )
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
```

```
# Print the formatted gt table
quartile_table_resilience_gt
```

Table 3: Resilience Score Summary Statistics

Descriptive statistics for Total, Control, and Target groups

Group	Minimum	Lower Quartile	Median	Upper Quartile	Maximum	Mean	Standard Deviation
Total	4.25	7.38	8.56	10.25	13.12	8.75	1.99
Control	4.88	6.28	7.62	8.34	11.00	7.38	1.38
Target	4.25	8.25	9.62	10.78	13.12	9.43	1.90

```
# Calculate the womens_empowerment_score
data1 <- data1 %>%
  mutate(
    # Part 1: House ownership
    house_ownership_score = case_when(
      `3. Do you own this or any other house either alone or jointly with
someone else?` == "Alone" ~ 2,
      `3. Do you own this or any other house either alone or jointly with
someone else?` == "Jointly" ~ 1,
      TRUE ~ 0
    ),

    # Part 2: Farmland ownership
    farmland_ownership_score = case_when(
      `4. Do you own farmland either alone or jointly with someone else?` ==
```

```
"Alone" ~ 2,
      `4. Do you own farmland either alone or jointly with someone else?` ==
"Jointly" ~ 1,
      TRUE ~ 0
    ),

    # Part 3: Title deed for any land owned
    title_deed_score = ifelse(`6. Do you have a title deed for any land you
own?` == "Yes", 1, 0),

    # Part 4: Name on the title deed
    name_on_deed_score = ifelse(`7. Is your name on the title deed?` ==
"Yes", 1, 0),

    # Part 5: Decision on how to spend the garden income
    decision_on_income_score = case_when(
      `62. Who decides how the garden money is used?` == "I decide on my own
how to spend the garden income" ~ 2,
      `62. Who decides how the garden money is used?` == "I discuss how to
spend it with my husband/partner" ~ 1,
      TRUE ~ 0
    ),

    # Part 6: Confidence to take on new things
    confidence_score = case_when(
      `81. “I believe in myself and am confident to take on new things”` ==
"Yes, definitely" ~ 2,
      `81. “I believe in myself and am confident to take on new things”` ==
"Yes, somewhat" ~ 1,
      TRUE ~ 0
    ),

    # Part 7: Future goals and dreams
    goals_and_dreams_score = case_when(
      `82. “I have a full idea of my future goals and dreams”` == "Yes,
definitely" ~ 2,
      `82. “I have a full idea of my future goals and dreams”` == "Yes,
somewhat" ~ 1,
      TRUE ~ 0
    ),

    # Part 8: Feeling proud of oneself
    proud_of_self_score = case_when(
      `83. “On the whole, I feel proud of myself”` == "Yes, definitely" ~ 2,
      `83. “On the whole, I feel proud of myself”` == "Yes, somewhat" ~ 1,
      TRUE ~ 0
    ),

    # Sum all parts to create the womens_empowerment_score
```

```
    womens_empowerment_score = (coalesce(house_ownership_score, 0) +
      coalesce(farmland_ownership_score, 0) +
      coalesce(title_deed_score, 0) +
      coalesce(name_on_deed_score, 0) +
      coalesce(decision_on_income_score, 0) +
      coalesce(confidence_score, 0) +
      coalesce(goals_and_dreams_score, 0) +
      coalesce(proud_of_self_score, 0))*5/7
  )


# Define the control and target groups
control_group_empowerment <- data1 %>% filter(`Household # (Code)` < 200)
target_group_empowerment <- data1 %>% filter(`Household # (Code)` >= 200)

# Add group labels to the dataset
control_group_empowerment <- control_group_empowerment %>%
mutate(Group_Empowerment = "Control")
target_group_empowerment <- target_group_empowerment %>%
mutate(Group_Empowerment = "Target")

# Combine the datasets
combined_data_empowerment <- bind_rows(control_group_empowerment,
target_group_empowerment)

# Plot the combined density plot with mean and median lines
ggplot(combined_data_empowerment, aes(x = womens_empowerment_score, color =
Group_Empowerment)) +
  geom_density(size = 1, trim=TRUE) +
  scale_color_manual(values = c("Control" = "#D2B48C", "Target" = "#FFA07A"))
+  # Consistent colors
  labs(title = "Women's Empowerment Score by Group",
       x = "Women's Empowerment Score",
       y = "Density") +
  theme_minimal() +
  geom_vline(aes(xintercept = mean(womens_empowerment_score[Group_Empowerment
== "Control"], na.rm = TRUE),
                 color = "Control"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept =
median(womens_empowerment_score[Group_Empowerment == "Control"], na.rm =
TRUE),
                 color = "Control"), linetype = "solid", size = 1) +
  geom_vline(aes(xintercept = mean(womens_empowerment_score[Group_Empowerment
== "Target"], na.rm = TRUE),
                 color = "Target"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept =
median(womens_empowerment_score[Group_Empowerment == "Target"], na.rm =
TRUE),
```

```
                color = "Target"), linetype = "solid", size = 1) +
  scale_x_continuous(limits = c(0, 10))
```


```
# Calculate statistics for total group
total_stats_women_empowerment <- calculate_quartiles(data1,
"womens_empowerment_score") %>%
  mutate(Group = "Total")

# Calculate statistics for control group
control_stats_women_empowerment <-
calculate_quartiles(control_group_empowerment, "womens_empowerment_score")
%>%
  mutate(Group = "Control")

# Calculate statistics for target group
target_stats_women_empowerment <-
calculate_quartiles(target_group_empowerment, "womens_empowerment_score") %>%
  mutate(Group = "Target")

# Combine all statistics into one table
quartile_table_women_empowerment <- bind_rows(total_stats_women_empowerment,
control_stats_women_empowerment, target_stats_women_empowerment)
```

```r
# Reorder columns to have Group first
quartile_table_women_empowerment <- quartile_table_women_empowerment %>%
select(Group, everything())

# Create a well-formatted table using gt
quartile_table_women_empowerment_gt <- quartile_table_women_empowerment %>%
  gt() %>%
  tab_header(
    title = "Women's Empowerment Score Summary Statistics",
    subtitle = "Descriptive statistics for Total, Control, and Target groups"
  ) %>%
  cols_label(
    Group = "Group",
    Min = "Minimum",
    Lower_Quartile = "Lower Quartile",
    Median = "Median",
    Third_Quartile = "Upper Quartile",
    Max = "Maximum",
    Mean = "Mean",
    SD = "Standard Deviation"
  ) %>%
  fmt_number(
    columns = vars(Min, Lower_Quartile, Median, Third_Quartile, Max, Mean,
SD),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
      weight = px(1)
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
```

```
    heading.align = "center"
  )

## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.

# Print the formatted gt table
quartile_table_women_empowerment_gt
```

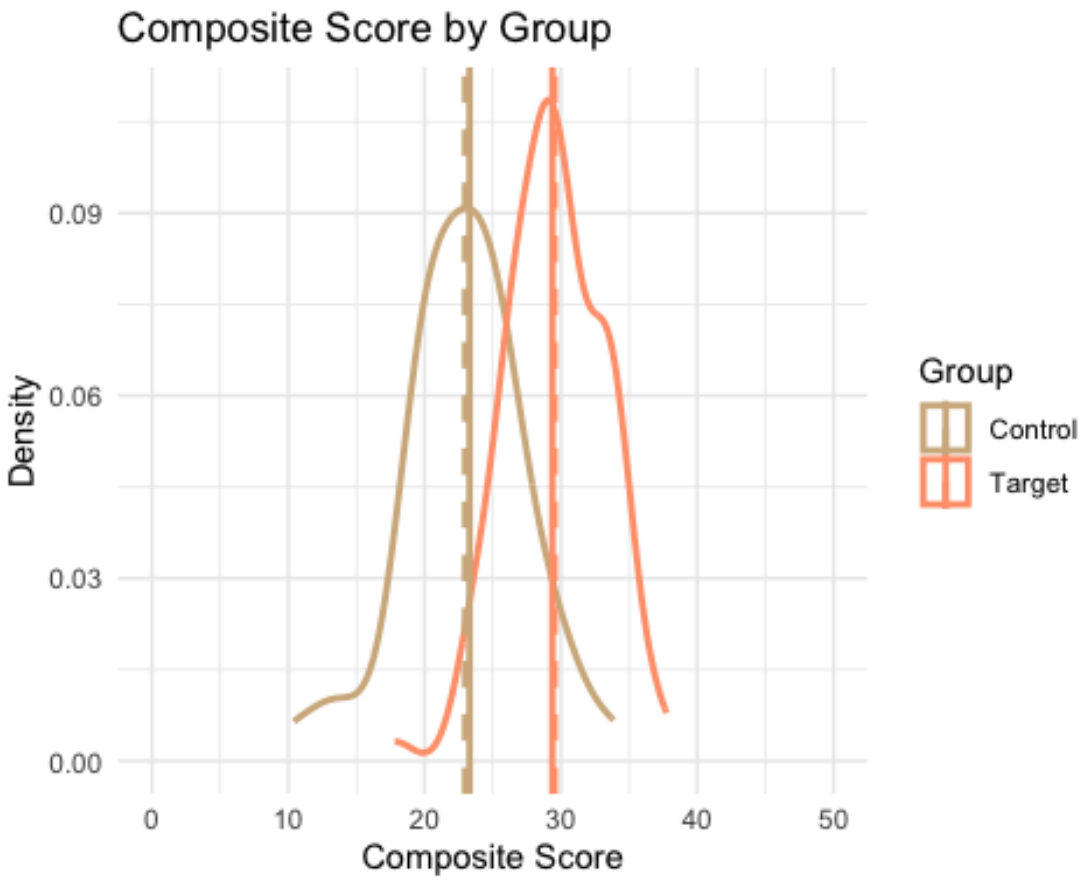
Table 4: Women's Empowerment Score Summary Statistics

Descriptive statistics for Total, Control, and Target groups

Group	Minimum	Lower Quartile	Median	Upper Quartile	Maximum	Mean	Standard Deviation
Total	0.00	5.00	6.43	7.14	10.00	6.05	1.62
Control	0.00	4.29	5.36	6.43	7.86	5.11	1.83
Target	2.14	5.71	6.43	7.14	10.00	6.51	1.27

```
#combine scores to make a composite score
data1$composite_score = data1$Food_Security_score +
data1$Health_and_nutrition_score + data1$resilience_score +
data1$womens_empowerment_score

# Define the control and target groups
control_group_compositescore <- data1 %>% filter(`Household # (Code)` < 200)
target_group_compositescore <- data1 %>% filter(`Household # (Code)` >= 200)

# Add group labels to the dataset
control_group_compositescore <- control_group_compositescore %>% mutate(Group
= "Control")
target_group_compositescore <- target_group_compositescore %>% mutate(Group =
"Target")

# Combine the datasets
combined_data_compositescore <- bind_rows(control_group_compositescore,
target_group_compositescore)

# Plot the combined density plot with mean and median lines
ggplot(combined_data_compositescore, aes(x = composite_score, color = Group))
+
  geom_density(size = 1, trim=TRUE) +
  scale_color_manual(values = c("Control" = "#D2B48C", "Target" = "#FFA07A"))
+  # Consistent colors
  labs(title = "Composite Score by Group",
       x = "Composite Score",
       y = "Density") +
  theme_minimal() +
  geom_vline(aes(xintercept = mean(composite_score[Group == "Control"], na.rm
= TRUE),
                 color = "Control"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(composite_score[Group == "Control"],
na.rm = TRUE),
```

```
                color = "Control"), linetype = "solid", size = 1) +
  geom_vline(aes(xintercept = mean(composite_score[Group == "Target"], na.rm
= TRUE),
                color = "Target"), linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(composite_score[Group == "Target"],
na.rm = TRUE),
                color = "Target"), linetype = "solid", size = 1) +
  scale_x_continuous(limits = c(0, 50))
```




```
# Calculate statistics for total group
total_stats_composite <- calculate_quartiles(data1, "composite_score") %>%
  mutate(Group = "Total")

# Calculate statistics for control group
control_stats_compositescore <-
calculate_quartiles(control_group_compositescore, "composite_score") %>%
  mutate(Group = "Control")

# Calculate statistics for target group
target_stats_compositescore <-
calculate_quartiles(target_group_compositescore, "composite_score") %>%
  mutate(Group = "Target")
```

```r
# Combine all statistics into one table
quartile_table_composite <- bind_rows(total_stats_composite,
control_stats_compositescore, target_stats_compositescore)

# Reorder columns to have Group first
quartile_table_composite <- quartile_table_composite %>% select(Group,
everything())

# Create a well-formatted table using gt
quartile_table_composite_gt <- quartile_table_composite %>%
  gt() %>%
  tab_header(
    title = "Composite Score Summary Statistics",
    subtitle = "Descriptive statistics for Total, Control, and Target groups"
  ) %>%
  cols_label(
    Group = "Group",
    Min = "Minimum",
    Lower_Quartile = "Lower Quartile",
    Median = "Median",
    Third_Quartile = "Upper Quartile",
    Max = "Maximum",
    Mean = "Mean",
    SD = "Standard Deviation"
  ) %>%
  fmt_number(
    columns = vars(Min, Lower_Quartile, Median, Third_Quartile, Max, Mean,
SD),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
      weight = px(1)
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
```

```
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
    heading.align = "center"
  )
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
```

```
# Print the formatted gt table
quartile_table_composite_gt
```

Table 5: Composite Score Summary Statistics

Descriptive statistics for Total, Control, and Target groups

Group	Minimum	Lower Quartile	Median	Upper Quartile	Maximum	Mean	Standard Deviation
Total	10.42	24.33	27.80	30.56	37.75	27.34	4.94
Control	10.42	20.23	23.30	26.00	33.86	22.97	4.43
Target	17.81	26.87	29.37	32.35	37.75	29.52	3.53

```
# Define the function to calculate the statistics for each score
calculate_statistics <- function(target_group, control_group, score) {
  target_values <- target_group[[score]]
  control_values <- control_group[[score]]

  # Remove NAs
  target_values <- target_values[!is.na(target_values)]
  control_values <- control_values[!is.na(control_values)]

  # Calculate means and standard deviations
  target_mean <- mean(target_values)
  target_sd <- sd(target_values)
  control_mean <- mean(control_values)
  control_sd <- sd(control_values)

  # Perform t-test
  t_test <- t.test(target_values, control_values, var.equal = TRUE)

  # Mean difference, t-statistic, and p-value
  mean_diff <- target_mean - control_mean
  t_stat <- t_test$statistic
  p_value <- t_test$p.value

  return(data.frame(
    Score = score,
    Target_Mean = target_mean,
    Target_SD = target_sd,
    Control_Mean = control_mean,
    Control_SD = control_sd,
```

```
    Mean_Difference = mean_diff,
    T_Statistic = t_stat,
    P_Value = p_value
  ))
}

# Define the control and target groups
control_group <- data1 %>% filter(`Household # (Code)` < 200)
target_group <- data1 %>% filter(`Household # (Code)` >= 200)

# List of scores to compare
scores <- c("Food_Security_score", "Health_and_nutrition_score",
"resilience_score", "womens_empowerment_score", "composite_score")

# Initialize an empty data frame to store the results
results_table <- data.frame()

# Loop through each score and calculate the statistics
for (score in scores) {
  stats <- calculate_statistics(target_group, control_group, score)
  results_table <- bind_rows(results_table, stats)
}

# Create a well-formatted table using gt
results_table_gt <- results_table %>%
  gt() %>%
  tab_header(
    title = "Comparison of Scores between Target and Control Groups",
    subtitle = "Mean, Standard Deviation, and T-Test Results"
  ) %>%
  cols_label(
    Score = "Score",
    Target_Mean = "Target Mean",
    Target_SD = "Target SD",
    Control_Mean = "Control Mean",
    Control_SD = "Control SD",
    Mean_Difference = "Mean Difference",
    T_Statistic = "T-Statistic",
    P_Value = "P-Value"
  ) %>%
  fmt_number(
    columns = vars(Target_Mean, Target_SD, Control_Mean, Control_SD,
Mean_Difference, T_Statistic, P_Value),
    decimals = 2
  ) %>%
  tab_style(
    style = cell_borders(
      sides = c("top", "bottom"),
      color = "black",
```

```
      weight = px(1)
    ),
    locations = cells_title(groups = c("title", "subtitle"))
  ) %>%
  tab_style(
    style = cell_borders(
      sides = "all",
      color = "gray",
      weight = px(1)
    ),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_column_labels()
  ) %>%
  tab_options(
    table.border.top.color = "black",
    table.border.bottom.color = "black",
    table.font.size = 12,
    heading.align = "center"
  ) %>%
  tab_style(
    style = cell_text(color = "red"),
    locations = cells_body(columns = vars(P_Value), rows = P_Value < 0.05)
  )
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
## Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## • Please use `columns = c(...)` instead.
```

```
# Print the formatted gt table
results_table_gt
```

Table 6: Comparison of Scores between Target and Control Groups

Mean, Standard Deviation, and T-Test Results

Score	Target Mean	Target SD	Control Mean	Control SD	Mean Difference	T-Statistic	P-Value
Food_Security_score	7.52	0.89	5.37	1.43	2.15	11.29	0.00
Health_and_nutrition_score	6.06	1.16	5.12	1.41	0.94	4.36	0.00
resilience_score	9.43	1.90	7.38	1.38	2.06	6.82	0.00
womens_empowerment_score	6.51	1.27	5.11	1.83	1.40	5.46	0.00
composite_score	29.52	3.53	22.97	4.43	6.55	9.82	0.00