Knowledge of Food Safety

Reducing Food Insecurity Among College Students

Katherine Chan

2025-10-30

Food insecurity, characterized by the lack of access to nutritious and quality food, affects millions worldwide and domestically in the United States. This issue strongly impacts college students, especially regarding their knowledge of cooking and cooking safety techniques. A cooking intervention was conducted among Binghamton University students who are at least 18 years and older. Participants completed pre and post-intervention surveys assessing their cooking safety knowledge, cooking efficacy, and food insecurity experiences.

Table of contents


```
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("ggplot2")
library("stats")
library("tidyr")
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.1     v readr     2.1.5
v lubridate 1.9.4     v stringr   1.5.2
v purrr     1.1.0     v tibble    3.3.0

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```r
library("ggpubr")
## source: https://r4ds.hadley.nz/spreadsheets.html
library(readxl)
# Read your Excel file
alldata <- read_excel("10.24.2025.data.team4.clean.xlsx")

# Check what columns actually contain "EFFSAFE" (print them)
grep("EFFSAFE", names(alldata), value = TRUE)
```

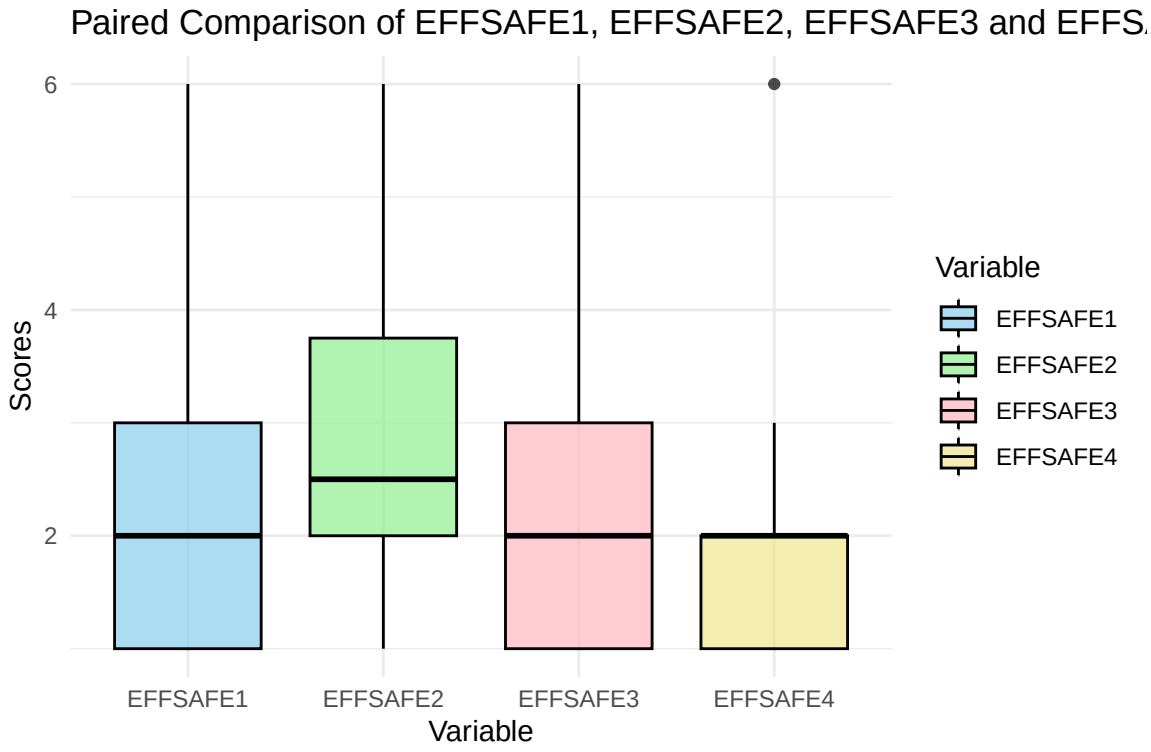
```
[1] "EFFSAFE1" "EFFSAFE2" "EFFSAFE3" "EFFSAFE4"
```

```r
# Automatically grab all EFFSAFE columns and rename them
effsafe_cols <- grep("EFFSAFE", names(alldata), value = TRUE)

selectdata <- alldata %>%
  select(all_of(effsafe_cols))
selectdata <- selectdata %>%
  mutate(
   EFFSAFE1 = case_when(
      EFFSAFE1 == 1 ~ 6,   # strongly disagree = 1
      EFFSAFE1 == 2 ~ 5,   # disagree = 2
      EFFSAFE1 == 3 ~ 4,   # slightly disagree = 3
      EFFSAFE1 == 4 ~ 3,   # slightly agree = 4
      EFFSAFE1 == 5 ~ 2,   # agree = 5
      EFFSAFE1 == 6 ~ 1,   # strongly agree = 6
      EFFSAFE1 == -50 ~ NA, # I don't know = NA
      EFFSAFE1 == -99 ~ NA  # I refuse to answer = NA
    ),
    EFFSAFE2 = case_when(
      EFFSAFE2 == 1 ~ 6,   # strongly disagree = 1
      EFFSAFE2 == 2 ~ 5,   # disagree = 2
      EFFSAFE2 == 3 ~ 4,   # slightly disagree = 3
      EFFSAFE2 == 4 ~ 3,   # slightly agree = 4
      EFFSAFE2 == 5 ~ 2,   # agree = 5
      EFFSAFE2 == 6 ~ 1,   # strongly agree = 6
      EFFSAFE2 == -50 ~ NA, # I don't know = NA
```

```
      EFFSAFE2 == -99 ~ NA  # I refuse to answer = NA
    ),
    EFFSAFE3 = case_when(
      EFFSAFE3 == 1 ~ 6,   # strongly disagree = 1
      EFFSAFE3 == 2 ~ 5,   # disagree = 2
      EFFSAFE3 == 3 ~ 4,   # slightly disagree = 3
      EFFSAFE3 == 4 ~ 3,   # slightly agree = 4
      EFFSAFE3 == 5 ~ 2,   # agree = 5
      EFFSAFE3 == 6 ~ 1,   # strongly agree = 6
      EFFSAFE3 == -50 ~ NA, # I don't know = NA
      EFFSAFE3 == -99 ~ NA  # I refuse to answer = NA
    ),
    EFFSAFE4 = case_when(
      EFFSAFE4 == 1 ~ 6,   # strongly disagree = 1
      EFFSAFE4 == 2 ~ 5,   # disagree = 2
      EFFSAFE4 == 3 ~ 4,   # slightly disagree = 3
      EFFSAFE4 == 4 ~ 3,   # slightly agree = 4
      EFFSAFE4 == 5 ~ 2,   # agree = 5
      EFFSAFE4 == 6 ~ 1,   # strongly agree = 6
      EFFSAFE4 == -50 ~ NA, # I don't know = NA
      EFFSAFE4 == -99 ~ NA  # I refuse to answer = NA
    )
  )

# Rename columns sequentially as EFFSAFE1, EFFSAFE2, ...
names(selectdata) <- paste0("EFFSAFE", seq_along(effsafe_cols))

# Convert to long format
longdata <- selectdata %>%
  pivot_longer(
    cols = everything(),
    names_to = "Variable",
    values_to = "Score"
  )

# Reshape the data to long format for ggplot
longdata <- selectdata %>%
  pivot_longer(
    cols = c(EFFSAFE1, EFFSAFE2, EFFSAFE3, EFFSAFE4),
    names_to = "Variable",
    values_to = "Score"
  )

# Create  boxplot
ggplot(longdata, aes(x = Variable, y = Score, fill = Variable)) +
  geom_boxplot(alpha = 0.7, color = "black") +
  labs(
```

```
    title = "Paired Comparison of EFFSAFE1, EFFSAFE2, EFFSAFE3 and EFFSAFE4",
    x = "Variable",
    y = "Scores"
  ) +
  scale_fill_manual(values = c("skyblue", "lightgreen", "lightpink", "khaki")) +
  theme_minimal()
```

```
Warning: Removed 41 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```
## Shapiro-Wilk normality test --> this will give p-values
## source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/shapiro.test.html
## note to self: shapiro-wilk norm. test should use the raw variables (not composite scores, e


# Convert all columns to numeric safely and remove rows with missing values
selectdata <- selectdata %>%
  mutate(across(everything(), ~ as.numeric(as.character(.)))) %>%
  drop_na()

# --- Shapiro-Wilk normality tests ---
# (Each requires at least 3 non-missing values)
shapiro.test(selectdata$EFFSAFE1)
```

```
    Shapiro-Wilk normality test

data:  selectdata$EFFSAFE1
W = 0.84066, p-value = 4.394e-06
```

```
shapiro.test(selectdata$EFFSAFE2)
```

```
    Shapiro-Wilk normality test

data:  selectdata$EFFSAFE2
W = 0.89583, p-value = 0.0002043
```

```
shapiro.test(selectdata$EFFSAFE3)
```

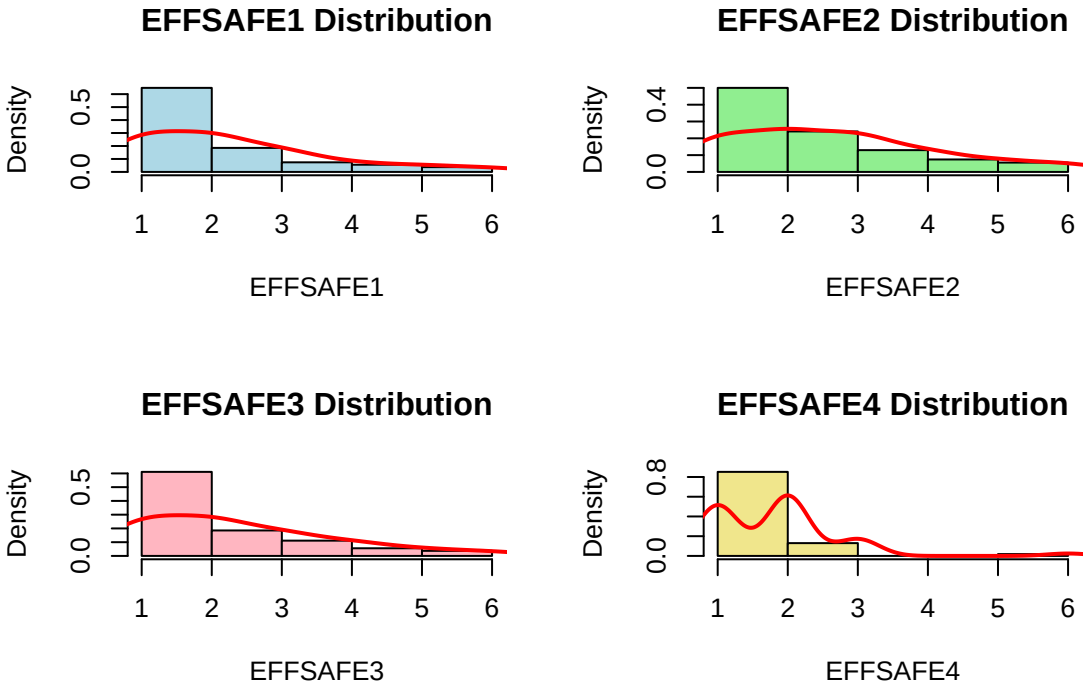
```
    Shapiro-Wilk normality test

data:  selectdata$EFFSAFE3
W = 0.8619, p-value = 1.748e-05
```

```
shapiro.test(selectdata$EFFSAFE4)
```

```
    Shapiro-Wilk normality test

data:  selectdata$EFFSAFE4
W = 0.72865, p-value = 1.165e-08
```

```
# --- Histograms and density plots ---
# Create 4 plots in a 2x2 grid
par(mfrow = c(2, 2))

hist(selectdata$EFFSAFE1,
     main = "EFFSAFE1 Distribution",
     xlab = "EFFSAFE1",
     col = "lightblue",
     freq = FALSE)
lines(density(selectdata$EFFSAFE1, na.rm = TRUE), col = "red", lwd = 2)

hist(selectdata$EFFSAFE2,
     main = "EFFSAFE2 Distribution",
     xlab = "EFFSAFE2",
     col = "lightgreen",
     freq = FALSE)
```

```r
lines(density(selectdata$EFFSAFE2, na.rm = TRUE), col = "red", lwd = 2)

hist(selectdata$EFFSAFE3,
     main = "EFFSAFE3 Distribution",
     xlab = "EFFSAFE3",
     col = "lightpink",
     freq = FALSE)
lines(density(selectdata$EFFSAFE3, na.rm = TRUE), col = "red", lwd = 2)

hist(selectdata$EFFSAFE4,
     main = "EFFSAFE4 Distribution",
     xlab = "EFFSAFE4",
     col = "khaki",
     freq = FALSE)
lines(density(selectdata$EFFSAFE4, na.rm = TRUE), col = "red", lwd = 2)
```




```r
## what this tells us:
## since both p-values are much smaller than 0.05, we reject the null hypothesis of normality
## that means EFFSAFE1, EFFSAFE2, EFFSAFE3, and EFFSAFE4 are NOT normally distributed
## --> paired samples t-test (which assumes normality), you should use the Wilcoxon signed-rank
```

1 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

2 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).