R Graph Templates for Visualizing Living Income

Molly Leavens on behalf of the Sustainable Food Lab

4/7/2021

Contents

Instructions for use

- The Sustainable Food Lab created this document to support companies and research organizations who are visualizing income data and its relationship to established Living Income Benchmarks. This document supplements tools developed by KIT to generate these graphics in Stata and Excel.
- The advantages of data cleaning, management, and visualization in R programming language are countless and include: free open source use, reproducibility and documentation of data cleaning, seamless integration with GitHub (a software/platform to track, store, and collaborate on code scripts), and flexible functions.
- While the code presented in this document is intended to be as user friendly as possible, some basic knowledge of R is necessary. The user will need to understand how to navigate the basic R interface to run the script and export desired graphics.
- The example graphs below visualize a publicly-available data set based on previous work of KIT for the Living Income Community of Practice.
- The sample data set assigns different Living Income benchmarks to different household types. No changes to the code are necessary if your data only has one benchmark for all household types or genders.
- To run your own data through the code provided below, it is imperative that you format your data in exactly the same structure as the sample data set. See the KIT Excel format HERE to copy the exact column names for each key variable. If you alter any column names in your data file, you will need to update this code accordingly. Ensure that you format all numerical variables without any commas or currency notations.

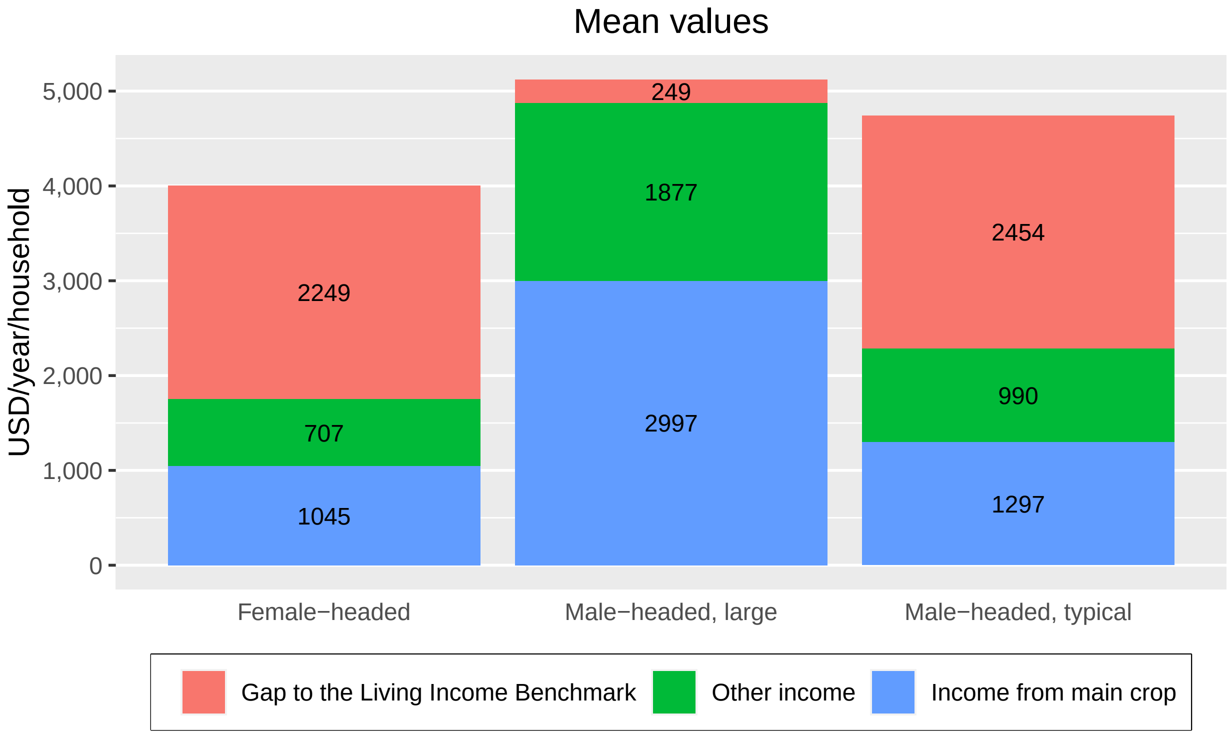
Setup

```
## Fist, load the R libraries that have the necessary functions for this analysis.

# If you have not previously used these libraries,
# you will need to instaLl them with the function install.packages("LIBRARY NAME")
library(tidyverse)
library(knitr)
library(scales)

## Import data set

# If you have troubles with the import,
# ensure the original data file is stored in your "working directory"
# (most often the same folder you have saved your working code script)

# If your original data file is saved as an Excel, export it as a CSV file.
# Ensure the top row is only column names.
# Replace 'KIT_Sample_Data' with the name of your data file.

DATA <- read_csv("KIT_Sample_Data.csv")
```

The gap of the mean income to the Living Income Benchmark - bar chart

```
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each household type, summarize the mean gap to the living income,
# the mean noncocoa income, and the mean cocoa income
  summarise(Gap = mean(benchmark - total_hh_income_2018),
            Noncocoa = mean(total_hh_income_2018 - total_cocoa_income_2018),
            cocoa = mean(total_cocoa_income_2018)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "stack", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Mean values",
       y = "USD/year/household",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                                "Noncocoa",
                                "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                                "Other income",
                                "Income from main crop")) +
# Format y-axis labels with a comma
  scale_y_continuous(labels = comma) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
        legend.position="bottom",
# Remove legend title
        legend.title = element_blank(),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
```

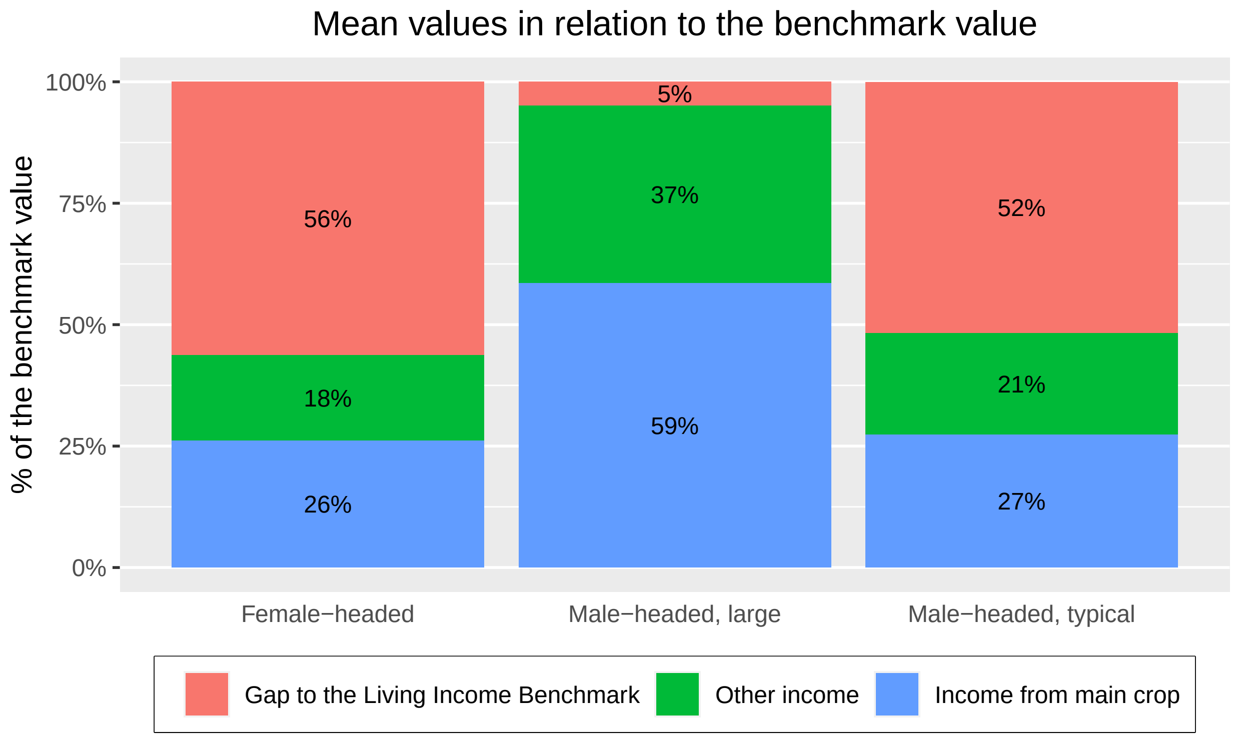
```
        plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
  geom_text(aes(label = round(Income)),
            position = position_stack(vjust = 0.5),
            size = 3)
```



Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

The gap of the relative mean income to the Living Income Benchmark - bar chart

```
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each household type, summarize the mean gap to the living income,
# the mean non-cocoa income, and the mean cocoa income.
# Calculate all as percentages of the Living income benchmark - the total of the income and gap
  summarise(Gap = mean((benchmark - total_hh_income_2018)/benchmark),
            Noncocoa = mean((total_hh_income_2018 - total_cocoa_income_2018)/benchmark),
            cocoa = mean(total_cocoa_income_2018/benchmark)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "stack", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Mean values in relation to the benchmark value",
       y = "% of the benchmark value",
       x = "",
# Manually UPDATE the numbers in this character string
  # to the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                               "Noncocoa",
                               "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                               "Other income",
                               "Income from main crop")) +
# Format y-axis labels with a comma
  scale_y_continuous(labels = percent) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
        legend.position="bottom",
# Remove legend title
        legend.title = element_blank(),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
  geom_text(aes(label = label_percent(accuracy = 1L)(Income)),
            position = position_stack(vjust = 0.5),
            size = 3)
```

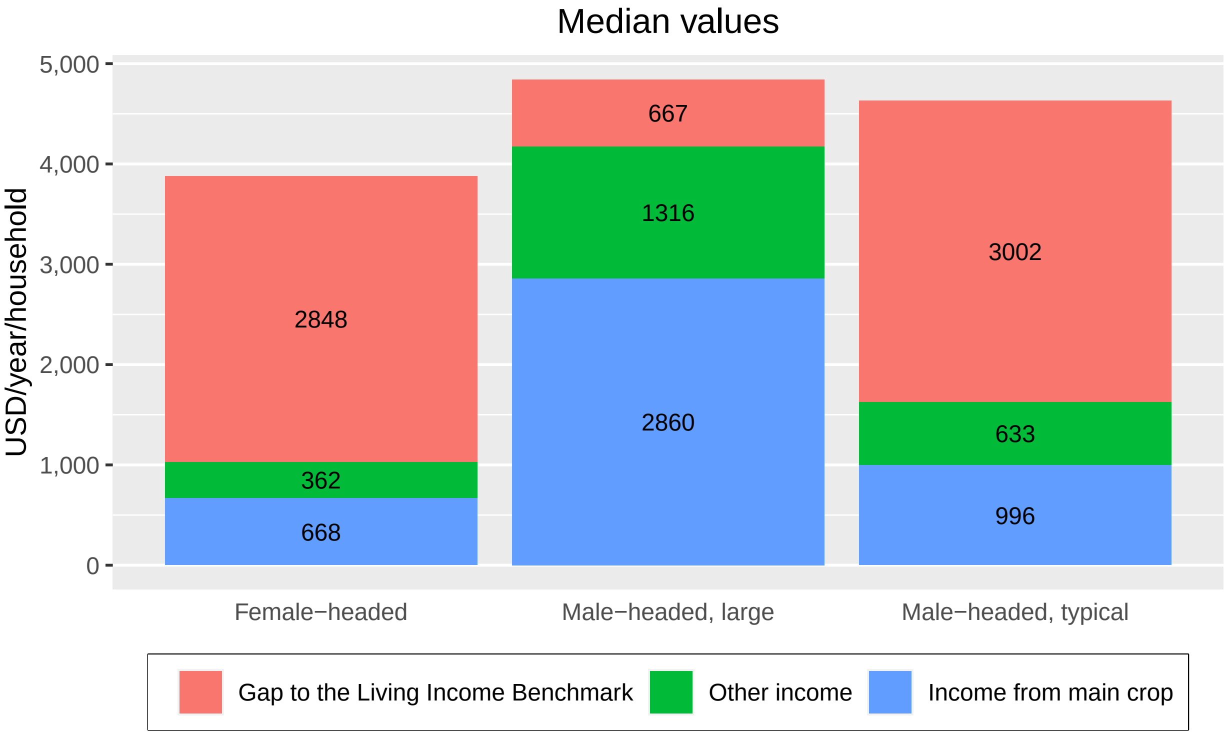


Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

The gap of the absolute median income to the Living Income Benchmark - bar chart

```r
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each gender, summarize the median gap to the living income,
# the median non-cocoa income, and the median cocoa income.
  summarise(Gap = median(benchmark - total_hh_income_2018),
            Noncocoa = median(total_hh_income_2018 - total_cocoa_income_2018),
            cocoa = median(total_cocoa_income_2018)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "stack", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Median values",
       y = "USD/year/household",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                                "Noncocoa",
                                "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                                "Other income",
                                "Income from main crop")) +
# Format y-axis labels with a comma
  scale_y_continuous(labels = comma) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
        legend.position="bottom",
# Remove legend title
        legend.title = element_blank(),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
```

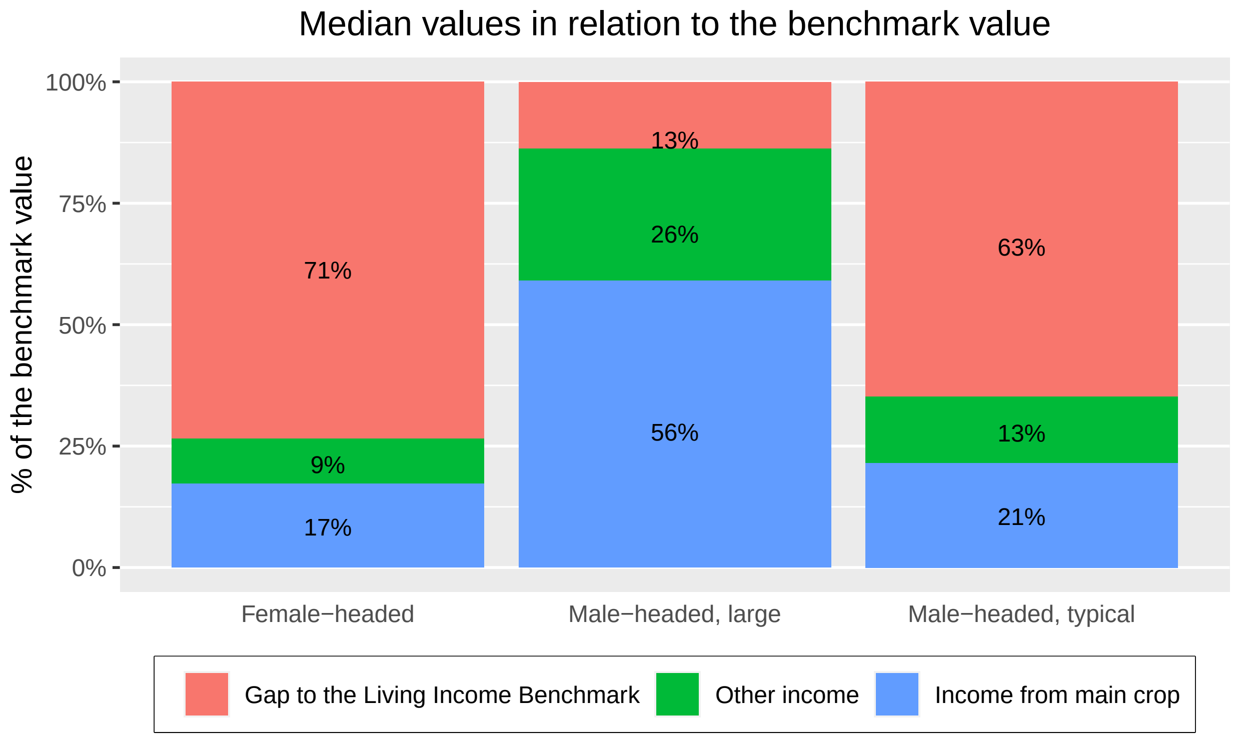
```
        plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
  geom_text(aes(label = round(Income)),
            position = position_stack(vjust = 0.5),
            size = 3)
```



Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

The gap of the relative median income to the Living Income Benchmark - bar chart

```r
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each gender, summarize the median gap to the living income,
# the median non-cocoa income, and the median cocoa income.
# Calculate all as percentages of the Living income benchmark - the total of the income and gap
  summarise(Gap = median((benchmark - total_hh_income_2018)/benchmark),
            Noncocoa = median((total_hh_income_2018 - total_cocoa_income_2018)/benchmark),
            cocoa = median(total_cocoa_income_2018/benchmark)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "fill", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Median values in relation to the benchmark value",
       y = "% of the benchmark value",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                                "Noncocoa",
                                "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                                "Other income",
                                "Income from main crop")) +
# Format y-axis labels with a comma
  scale_y_continuous(labels = percent) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
        legend.position="bottom",
# Remove legend title
        legend.title = element_blank(),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
  geom_text(aes(label = label_percent(accuracy = 1L)(Income)),
            position = position_stack(vjust = 0.5),
            size = 3)
```



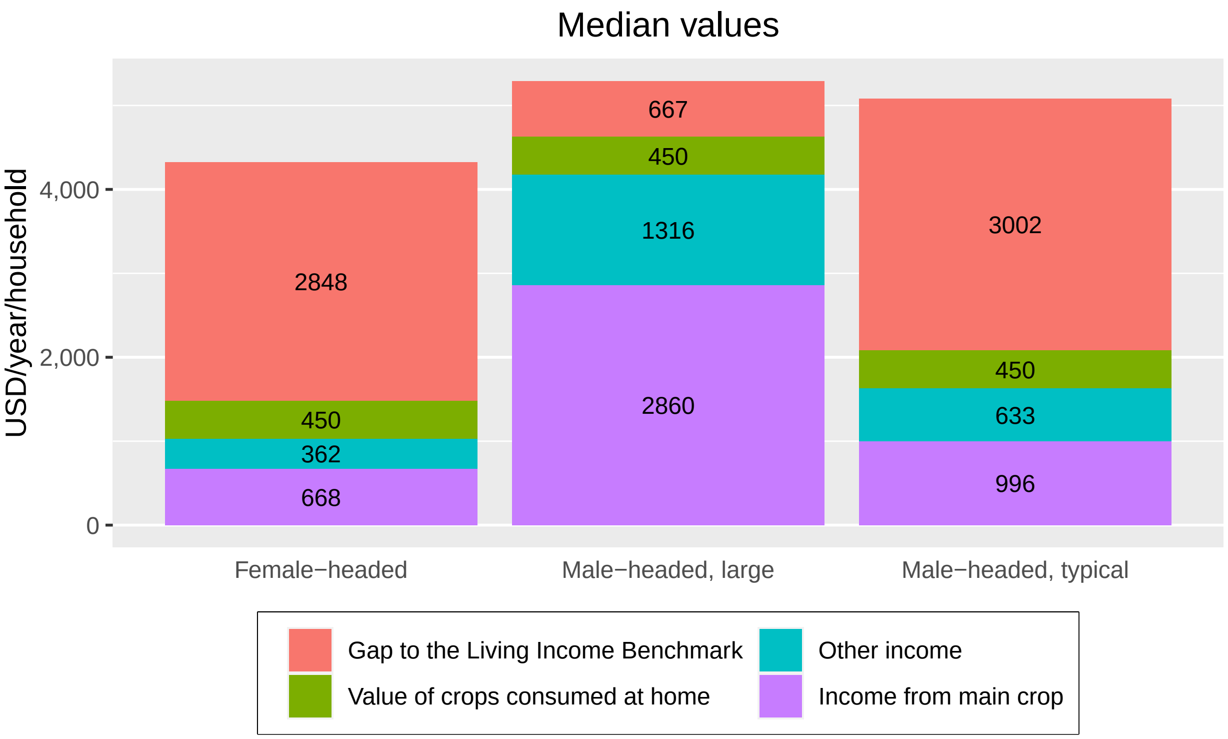
Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

Bar charts including the intrinsic value of food produced and consumed at home

The gap of the absolute median income to the Living Income Benchmark - bar chart

```r
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by gender
  group_by(grouping) %>%
# For each gender, summarize the median gap to the living income,
# the median non-cocoa income, and the median cocoa income.
  summarise(Gap = median(benchmark - total_hh_income_2018),
            Noncocoa = median(total_hh_income_2018 - total_cocoa_income_2018),
            Food = median(food_value),
            cocoa = median(total_cocoa_income_2018)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Food", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "stack", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Median values",
       y = "USD/year/household",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                               "Food",
                               "Noncocoa",
                               "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                               "Value of crops consumed at home",
                                "Other income",
                                "Income from main crop")) +
# Wrap legend onto 2 lines to fit everything neatly
  guides(fill = guide_legend(nrow = 2)) +
# Format y-axis labels with a comma
  scale_y_continuous(labels = comma) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
```

```
    plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
    legend.position="bottom",
# Remove legend title
    legend.title = element_blank(),
# Put a box around the legend
    legend.box.background = element_rect(),
# Move caption to desired location
    plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
 geom_text(aes(label = round(Income)),
      position = position_stack(vjust = 0.5),
      size = 3)
```

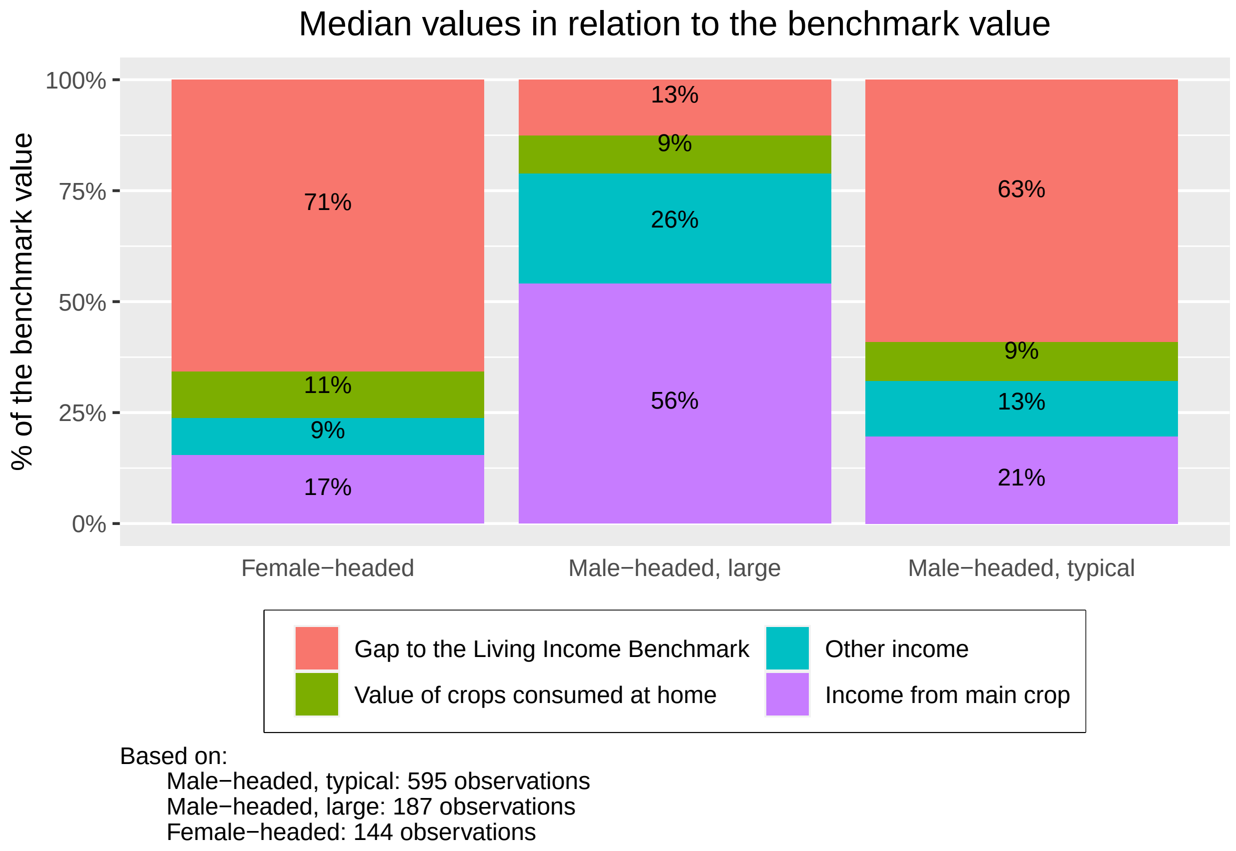


Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

The gap of the relative median income to the Living Income Benchmark - bar chart

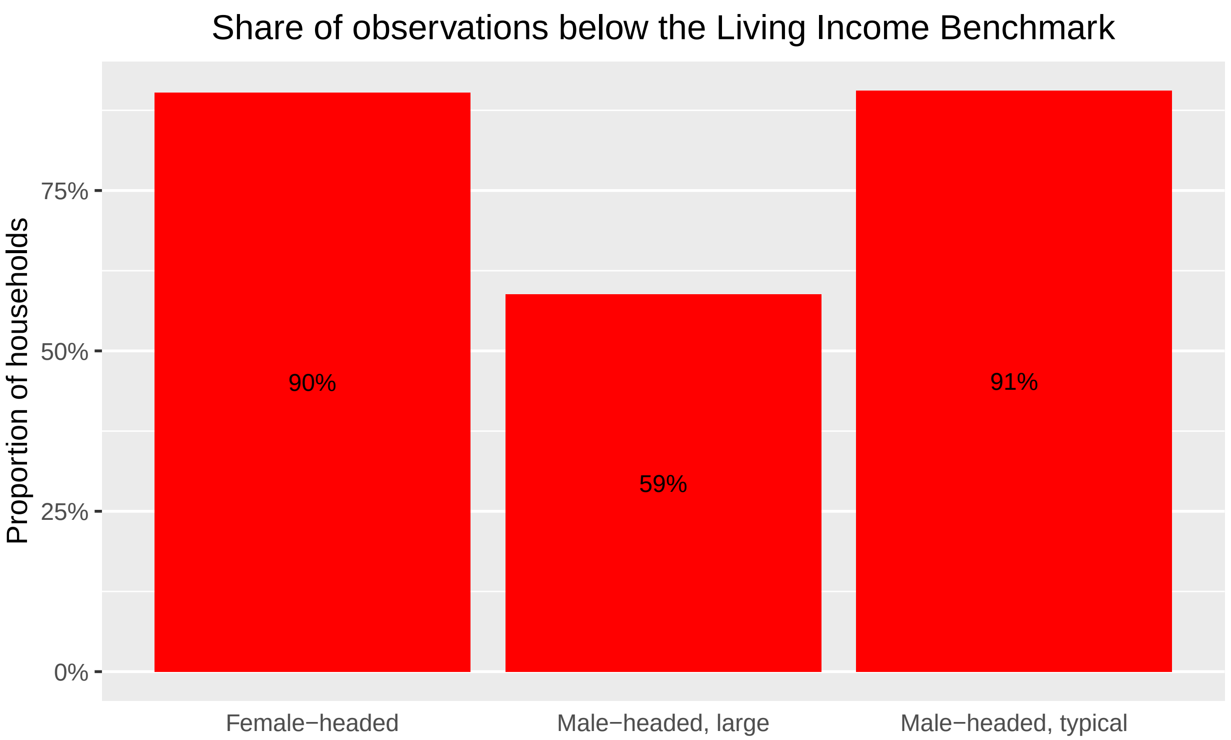
```
## The first section of this code summarizes and formats the data to be graph-ready
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each household type, summarize the median gap to the living income,
# the median non-cocoa income, and the median cocoa income.
# Calculate all as percentages of the Living income benchmark - the total of the income and gap
  summarise(Gap = median((benchmark - total_hh_income_2018)/benchmark),
            Food = median(food_value/benchmark),
            Noncocoa = median((total_hh_income_2018 - total_cocoa_income_2018)/benchmark),
            cocoa = median(total_cocoa_income_2018/benchmark)) %>%
# Gather each income components into one column so the data is in 'long' format
  gather(key = "Component", value = "Income", Gap:cocoa) %>%
# Re-level the income factors for the order you want them stacked on the graph
  mutate(Component = factor(Component,
                            levels = c("Gap", "Food", "Noncocoa", "cocoa"))) %>%
# Generate ggplot graph for income by gender and income component
  ggplot(aes(y = Income, x = grouping, fill = Component)) +
# Assign graph as stacked bar chart
  geom_bar(position = "fill", stat = "identity") +
# Label the graph title, axis, and caption
  labs(title = "Median values in relation to the benchmark value",
     y = "% of the benchmark value",
     x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
     caption = "Based on:
     Male-headed, typical: 595 observations
     Male-headed, large: 187 observations
     Female-headed: 144 observations") +
# Label the legend
  scale_fill_discrete(breaks=c("Gap",
                               "Food",
                               "Noncocoa",
                               "cocoa"),
                      labels=c("Gap to the Living Income Benchmark",
                               "Value of crops consumed at home",
                               "Other income",
                               "Income from main crop")) +
# Wrap legend onto 2 lines to fit everything neatly
  guides(fill = guide_legend(nrow = 2)) +
# Format y-axis labels with a comma and assign limits 0-100%
  scale_y_continuous(labels = percent, limits = c(0,1)) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move the legend to the bottom of the graph
        legend.position="bottom",
```

```r
# Remove legend title
        legend.title = element_blank(),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Add incomes to prospective graph components
  geom_text(aes(label = label_percent(accuracy = 1L)(Income)),
            position = position_stack(vjust = 0.5),
            size = 3)
```



Share of those below the Living Income benchmark - bar chart

```r
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each household type, calculate the percentage above the living income benchmark
  summarise(Below = sum(below_benchmark)/n()) %>%
# Generate ggplot graph for percentage by gender
  ggplot(aes(x = grouping, y = Below)) +
# Assign graph as bar graph and color bars red for aesthetics
  geom_col(fill= "red") +
# Label the graph title, axis, and caption
  labs(title = "Share of observations below the Living Income Benchmark",
       y = "Proportion of households",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Format y-axis labels with a percent
  scale_y_continuous(labels = percent) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Add percents to each graph bar
  geom_text(aes(label = label_percent(accuracy = 1L)(Below)),
            position = position_stack(vjust = 0.5),
            size = 3)
```


Share of observations below the Living Income Benchmark

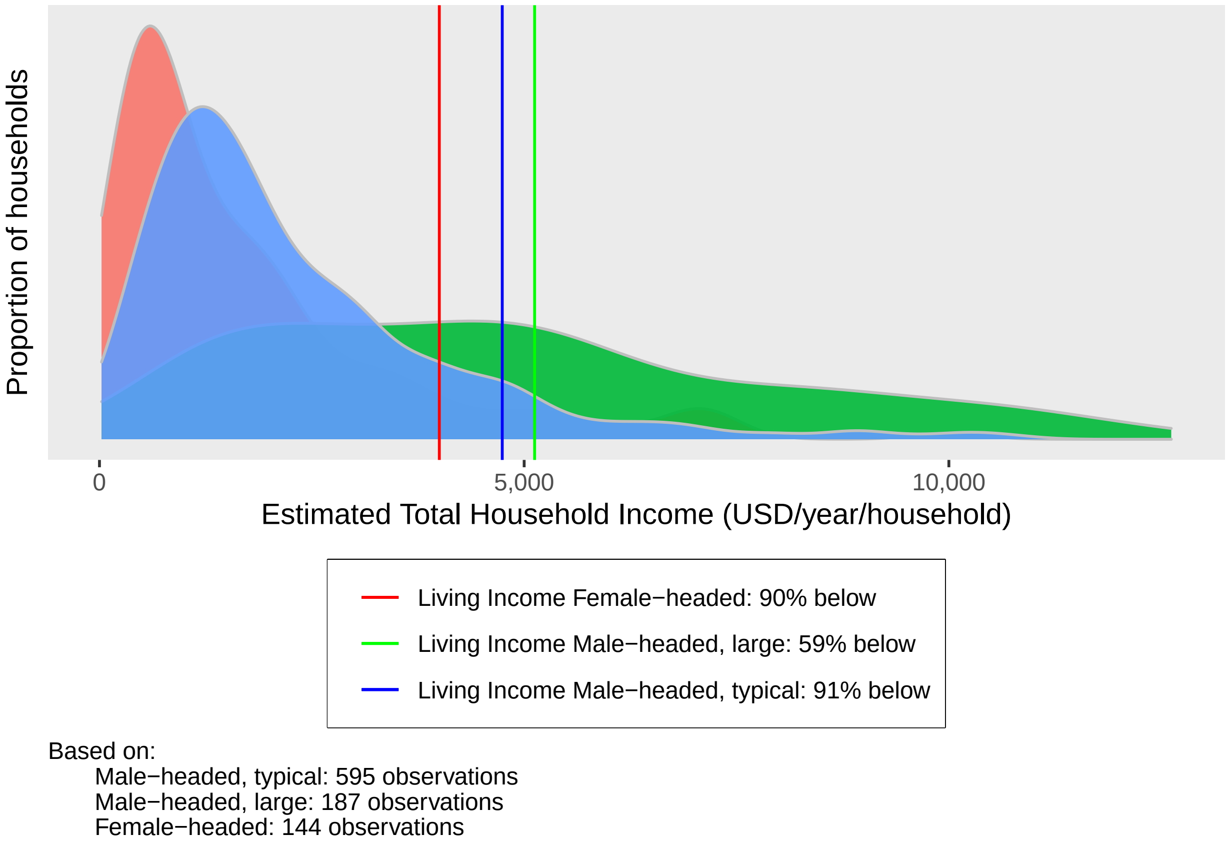
Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

Share of those below the Living Income Benchmark - distributional plot

- The KIT Guidance Document displays this graph style as a smooth histogram with binned income ranges. However, for a variety of reasons, we felt that the data would be better visualized with a density graph, so that is what the below code outputs.

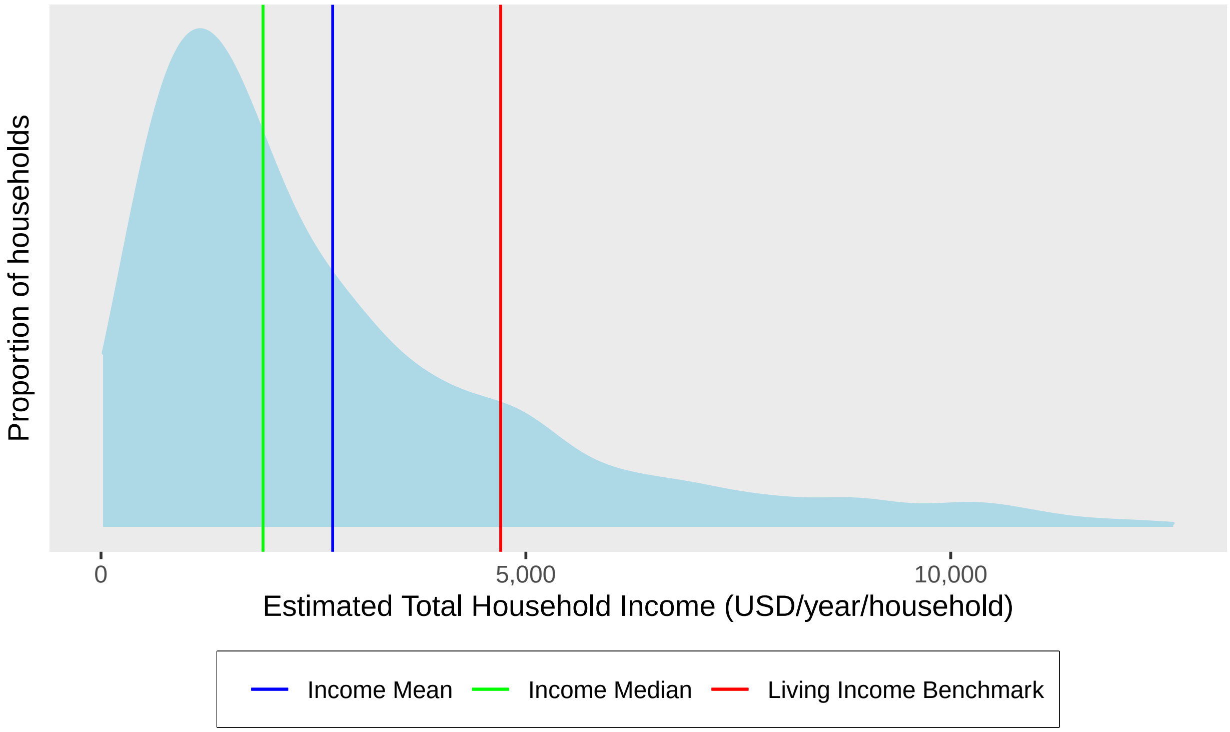
By household type

```
DATA %>%
# Set x-axis to total household income
  ggplot(aes(x = total_hh_income_2018,
# Color by household type
             fill = grouping)) +
# Add density line. You can change the fill and line colors.
  geom_density(alpha = 0.9, color = "grey") +

### Add vertical lines for Living Income benchmark(s)
# UPDATE these benchmarks and percent above manually
# Add and/or subtract lines based on number of household types
# and if you have a different benchmark for each.
  geom_vline(aes(xintercept = 4001,
                 color = "Living Income Female-headed: 90% below"),
             key_glyph = "path") +
  geom_vline(aes(xintercept = 4742,
                 color = "Living Income Male-headed, typical: 91% below"),
             key_glyph = "path") +
  geom_vline(aes(xintercept = 5123,
                 color = "Living Income Male-headed, large: 59% below"),
             key_glyph = "path") +
  scale_color_manual(name = "Legend",
                     values = c("Living Income Female-headed: 90% below"= "red",
                                "Living Income Male-headed, typical: 91% below"= "blue",
                                "Living Income Male-headed, large: 59% below"="green")) +
# Add graph and axis labels
  labs(x = "Estimated Total Household Income (USD/year/household)",
       y = "Proportion of households",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Format x-axis labels as numbers with commas
# If your data has extreme income outliers,
    # you may need to filter them or add x-axis limits so they do not warp the graph
  scale_x_continuous(labels = comma) +
# Remove y-axis labels and ticks
  theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(),
# Remove background grid (the grid is the default)
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
# Move the legend to the bottom of the graph
        legend.position="bottom",
# Remove legend title
        legend.title = element_blank(),
# Remove gray background for legend symbols
        legend.key=element_rect(fill=NA),
# Put a box around the legend
        legend.box.background = element_rect(),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Wrap legend onto 2 lines to fit everything neatly
  guides(fill = FALSE) +
# Wrap legend onto 3 lines to fit everything neatly
  guides(color = guide_legend(nrow = 3))
```



With mean and median

```r
DATA %>%
  # Set x-axis to key variable
  ggplot(aes(x = total_hh_income_2018)) +
  # Add density line. You can change the fill and line colors.
  geom_density(color = "#add8e6", fill = "#add8e6") +
  # Add vertical line for living income
  geom_vline(aes(xintercept = mean(benchmark),
                 color = "Living Income Benchmark"), key_glyph = "path") +
  # Add vertical line for mean income
  geom_vline(aes(xintercept = mean(DATA$total_hh_income_2018),
                 color = "Income Mean"), key_glyph = "path") +
  # Add vertical line for median income
  geom_vline(aes(xintercept = median(DATA$total_hh_income_2018),
                 color = "Income Median"), key_glyph = "path") +
  # Add graph and axis labels
  labs(x = "Estimated Total Household Income (USD/year/household)",
       y = "Proportion of households",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
  # Format x-axis labels as numbers with commas
  # If your data has extreme income outliers,
    # you may need to filter them or add x-axis limits so they do not warp the graph
  scale_x_continuous(labels = comma) +
  # Remove y-axis labels and ticks
  theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(),
  # Remove background grid (the grid is the default)
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  # Move the legend to the bottom of the graph
        legend.position="bottom",
  # Remove legend title
        legend.title = element_blank(),
  # Remove gray background for legend symbols
        legend.key=element_rect(fill=NA),
  # Put a box around the legend
        legend.box.background = element_rect(),
  # Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
  # Manually add legend
  scale_color_manual(values = c("Living Income Benchmark"= "red",
                                "Income Mean"= "blue",
                                "Income Median"="green"))
```

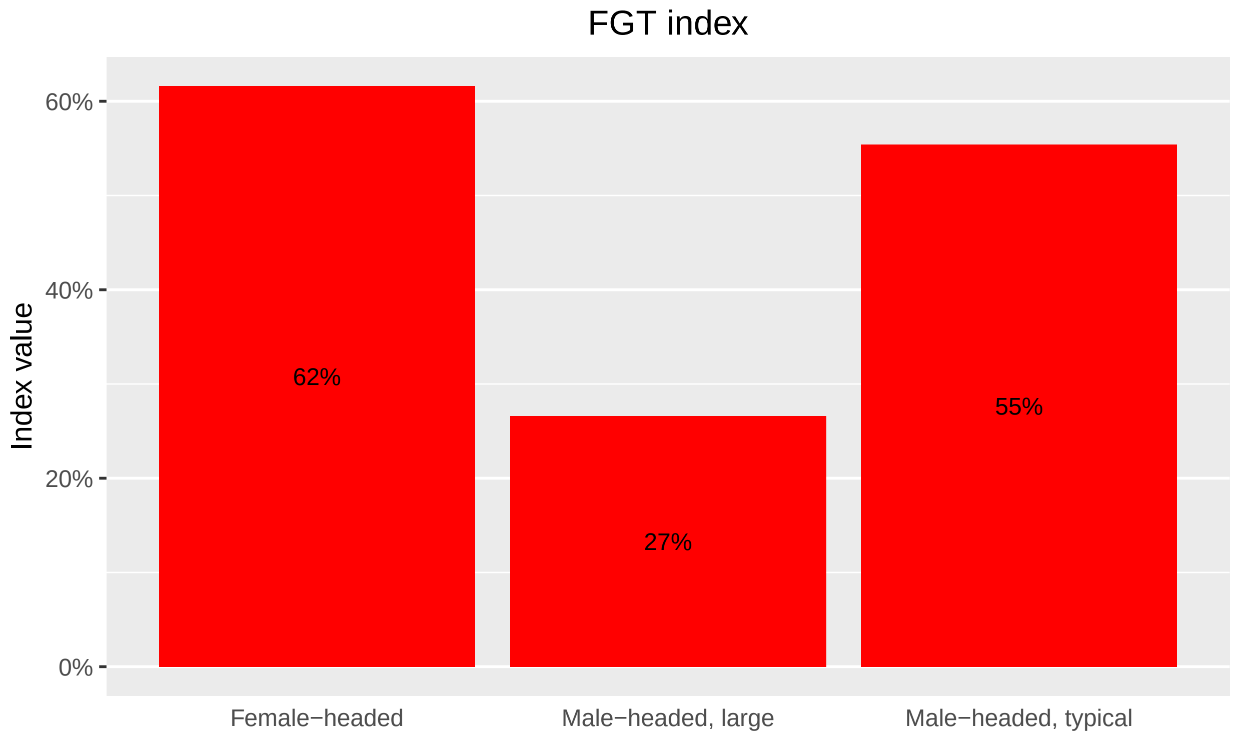


Based on:

Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations

Foster–Greer–Thorbecke (FGT) index

```r
DATA %>%
# Group by household type
  group_by(grouping) %>%
# For each gender, calculate the average Foster-Greer-Thorbecke (FGT) index
  summarise(FGT = mean(fgt_gap)) %>%
# Generate ggplot graph for percentage by gender
  ggplot(aes(x = grouping, y = FGT)) +
# Assign graph as bar graph and color bars red for aesthetics
  geom_col(fill= "red") +
# Label the graph title, axis, and caption
  labs(title = "FGT index",
       y = "Index value",
       x = "",
# Manually UPDATE the numbers in this character string to
  # the number of observations for each household type
       caption = "Based on:
       Male-headed, typical: 595 observations
       Male-headed, large: 187 observations
       Female-headed: 144 observations") +
# Format y-axis labels with a percent
  scale_y_continuous(labels = percent) +
# Remove x-axis grid lines and tick marks
  theme(panel.grid.major.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
# Center plot title
        plot.title = element_text(hjust = 0.5),
# Move caption to desired location
        plot.caption = element_text(hjust = 0)) +
# Add percents to each graph bar
  geom_text(aes(label = label_percent(accuracy = 1L)(FGT)),
            position = position_stack(vjust = 0.5),
            size = 3)
```



Based on:
Male−headed, typical: 595 observations
Male−headed, large: 187 observations
Female−headed: 144 observations