AE 04: Adjusting scales for World Bank indicators

Suggested answers

APPLICATION EXERCISE ANSWERS

MODIFIED
February 3, 2025

Important

These are suggested answers. This document should be used as reference only, it's not designed to be an exhaustive key.

```
library(tidyverse)
library(viridis)
library(scales)

options(scipen = 999)   # avoid printing in scientific notation
theme_set(theme_minimal()) # different default theme
```

Data: World economic measures

The World Bank publishes a rich and detailed set of socioeconomic indicators spanning several decades and dozens of topics. Here we focus on a few key indicators for the year 2021.

- `gdp_per_cap` - GDP per capita (current USD)
- `pop` - Total population
- `life_exp` - Life expectancy at birth, total (years)
- `female_labor_pct` - Labor force, female (% of total labor force)
- `income_level` - Classification of economies based on national income levels

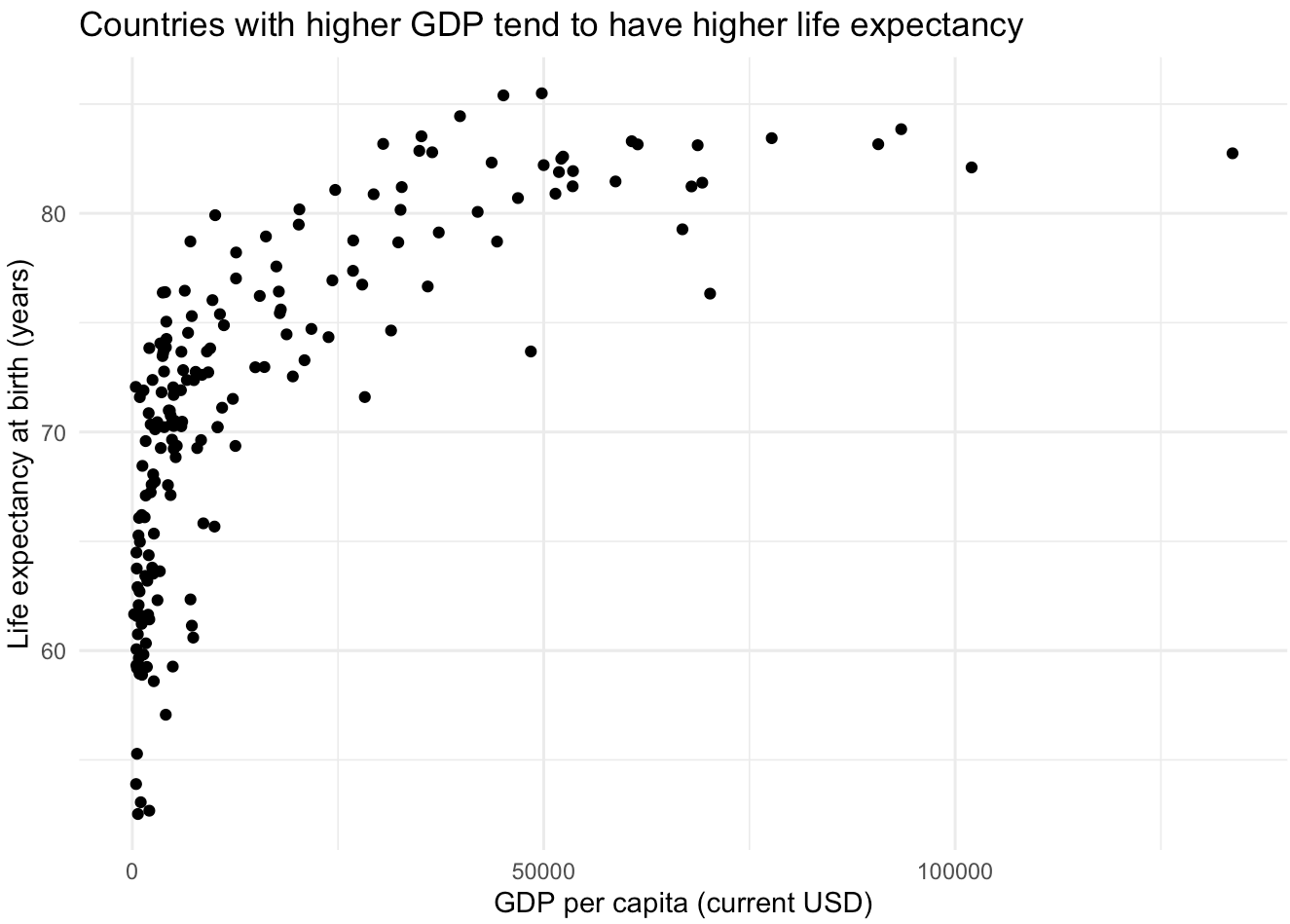
The data is stored in `wb-indicators.rds`. To import the data, use the `read_rds()` function.

```
world_bank <- read_rds("data/wb-indicators.rds")
```

Part 1: Transforming axes

Is there a relationship between a country's per capita GDP and life expectancy? Let's explore this relationship using a scatterplot.

```
ggplot(data = world_bank, mapping = aes(x = gdp_per_cap, y = life_exp)) +
  geom_point() +
  labs(
    title = "Countries with higher GDP tend to have higher life expectancy",
    x = "GDP per capita (current USD)",
    y = "Life expectancy at birth (years)"
  )
```



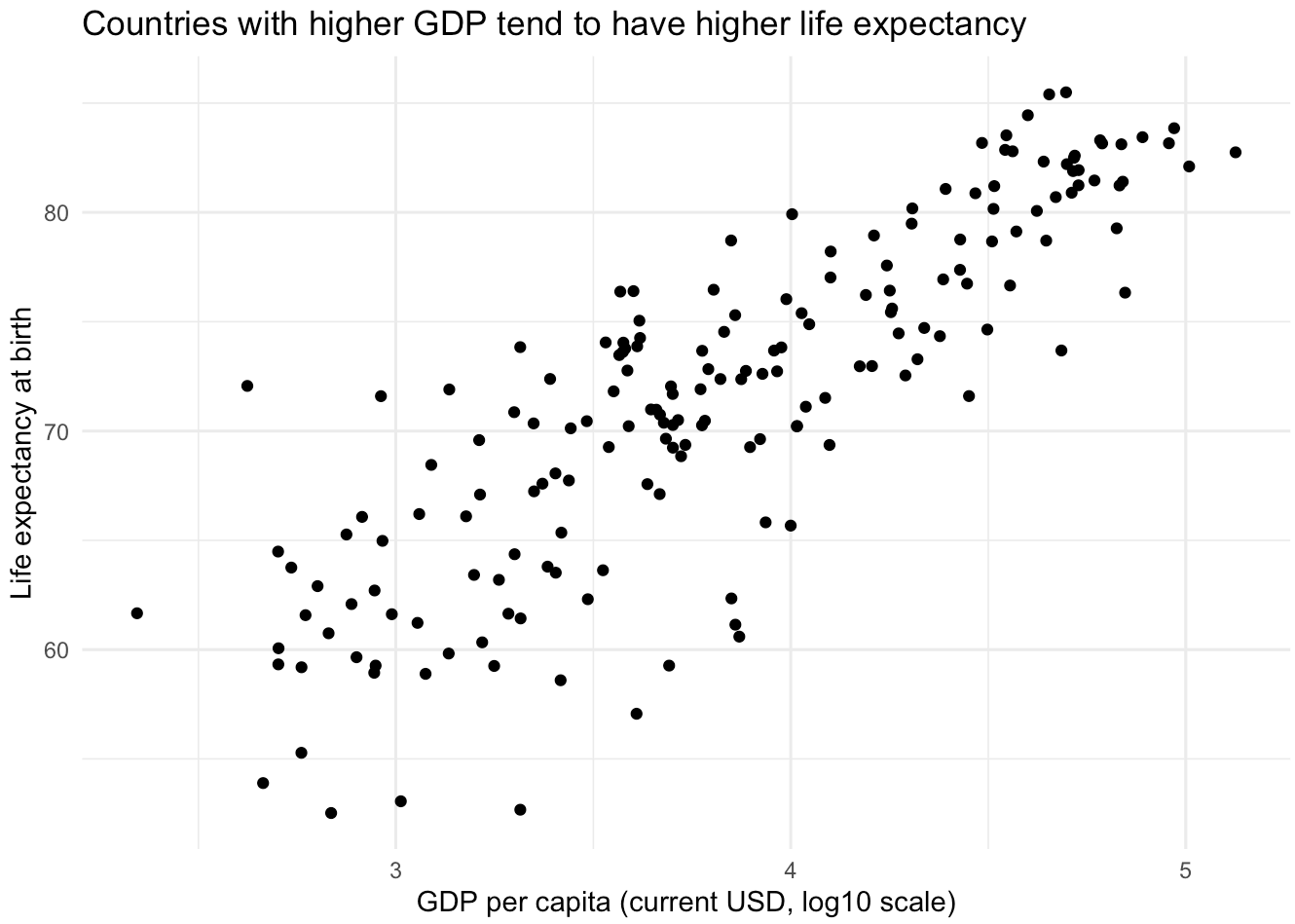
Seems like there is an association, but the relationship is not linear. Let's try a log transformation on the x-axis to see if that helps.

Your turn: Log-transform the x-axis by mutating the original column prior to graphing.

> **Note**
>
> By default, `log()` computes **natural logarithms** (base-e). To compute base-10 logarithms, use `log10()`.

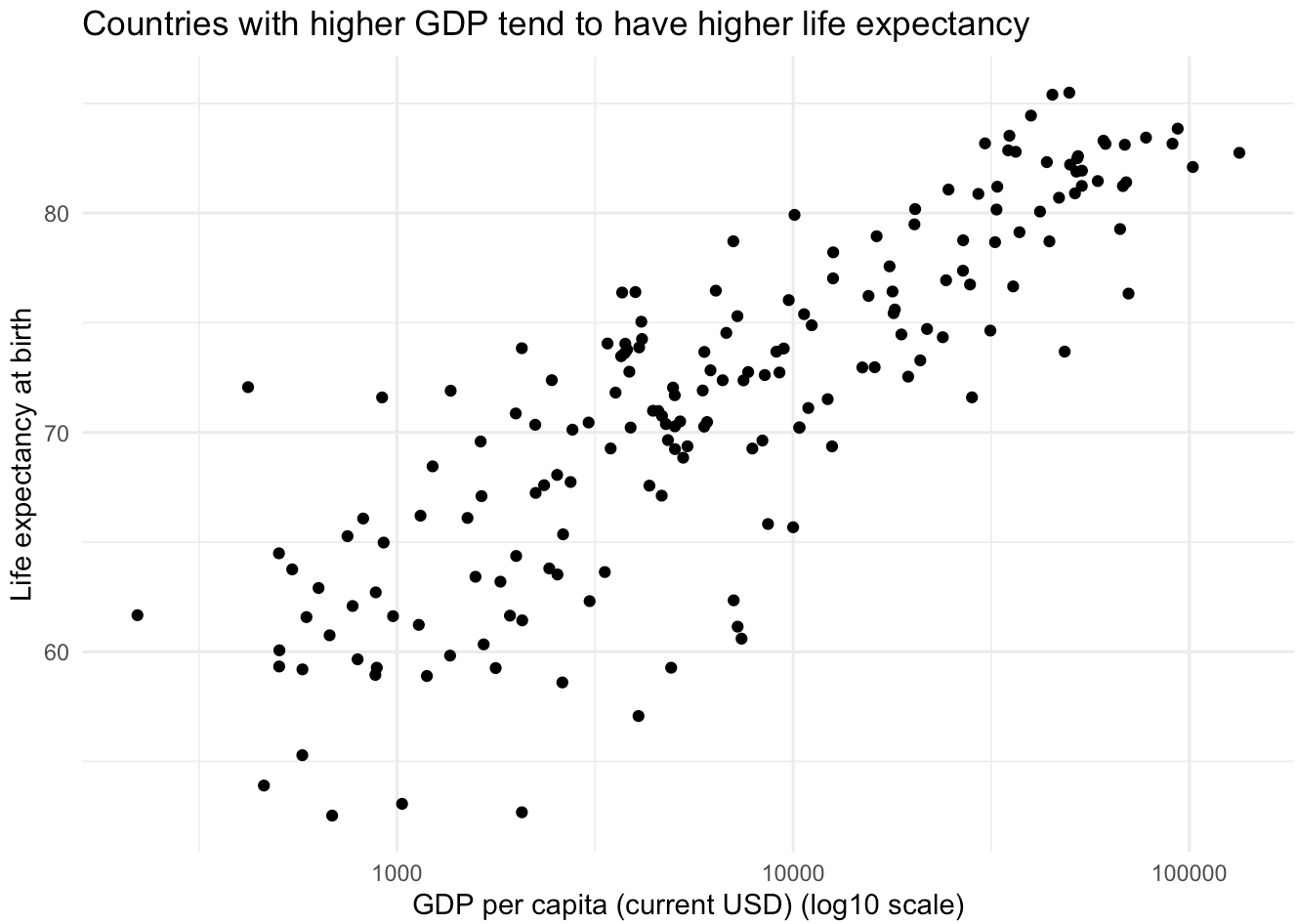
```
world_bank |>
  mutate(gdp_per_cap = log10(gdp_per_cap)) |>
  ggplot(mapping = aes(x = gdp_per_cap, y = life_exp)) +
  geom_point() +
```

```
  labs(
    title = "Countries with higher GDP tend to have higher life expectancy",
    x = "GDP per capita (current USD, log10 scale)",
    y = "Life expectancy at birth"
  )
```



Your turn: Now log-transform the x-axis by using the original per capita GDP measure and an appropriate `scale_x_*()` function.

```
ggplot(data = world_bank, mapping = aes(x = gdp_per_cap, y = life_exp)) +
  geom_point() +
  scale_x_log10() +
  labs(
    title = "Countries with higher GDP tend to have higher life expectancy",
    x = "GDP per capita (current USD) (log10 scale)",
    y = "Life expectancy at birth"
  )
```



Your turn: Which is more interpretable, and why?

Log-transforming the column first results in a plot with non-sensical labels on the x-axis. People do not intuitively understand log scales – we have to exponentiate out of them to understand what they represent. This is occasionally the case with base-10 logarithmic transformations, but often we employ **natural logarithms** (or base-e transformations) to transform variables.

It's better to use the `scale_*_log10()` function to transform the axis directly. This way, the labels are directly human-readable.

Part 2: Customize scales

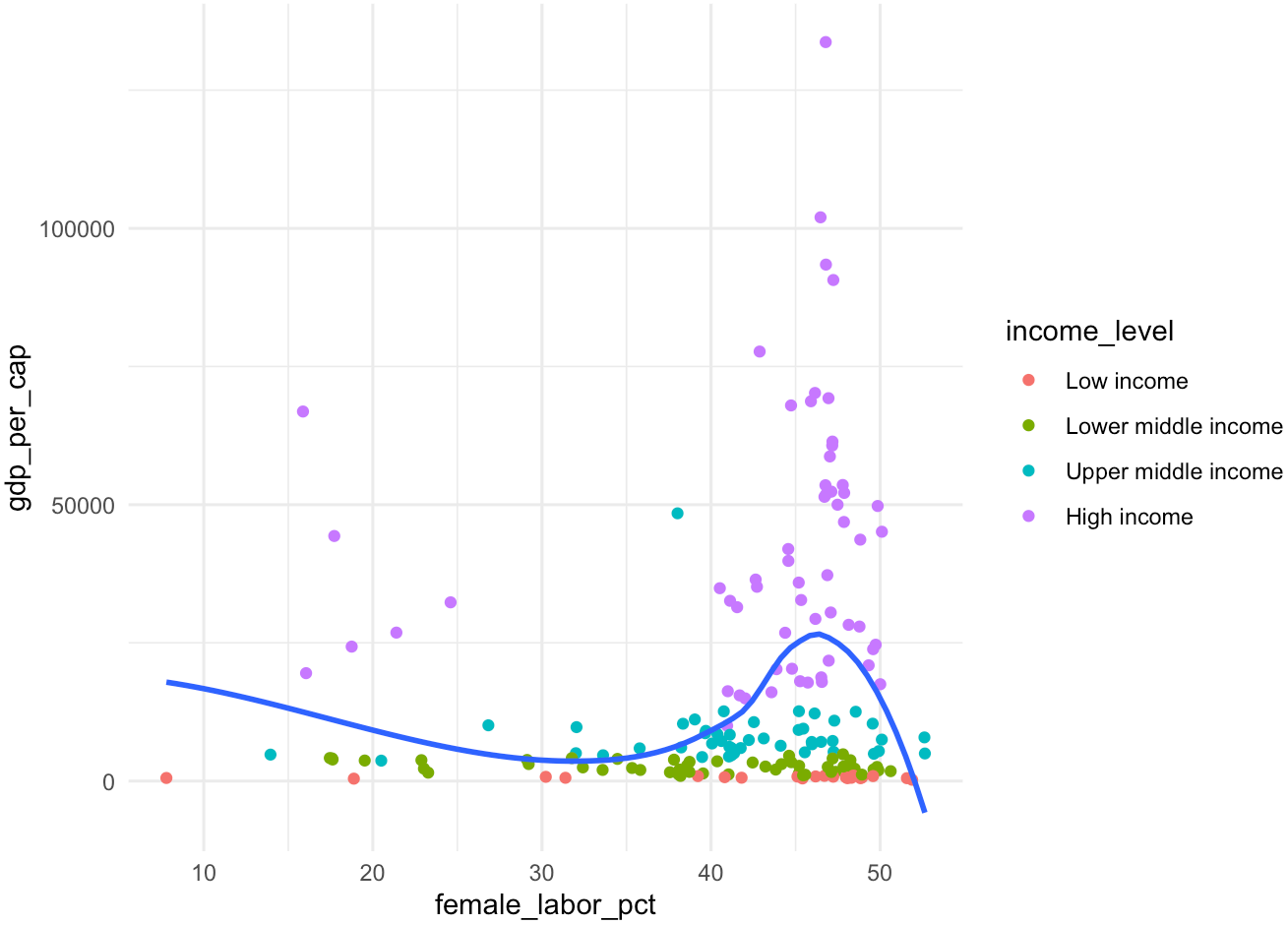
Let's consider the relationship between female labor participation and per capita GDP. We'll use the `income_level` variable to color the points and provide context on the overall wealth of the countries.[1]

Step 1: Base plot

First, let's generate a color-coded scatterplot with a single smoothing line.

```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level)) +
```

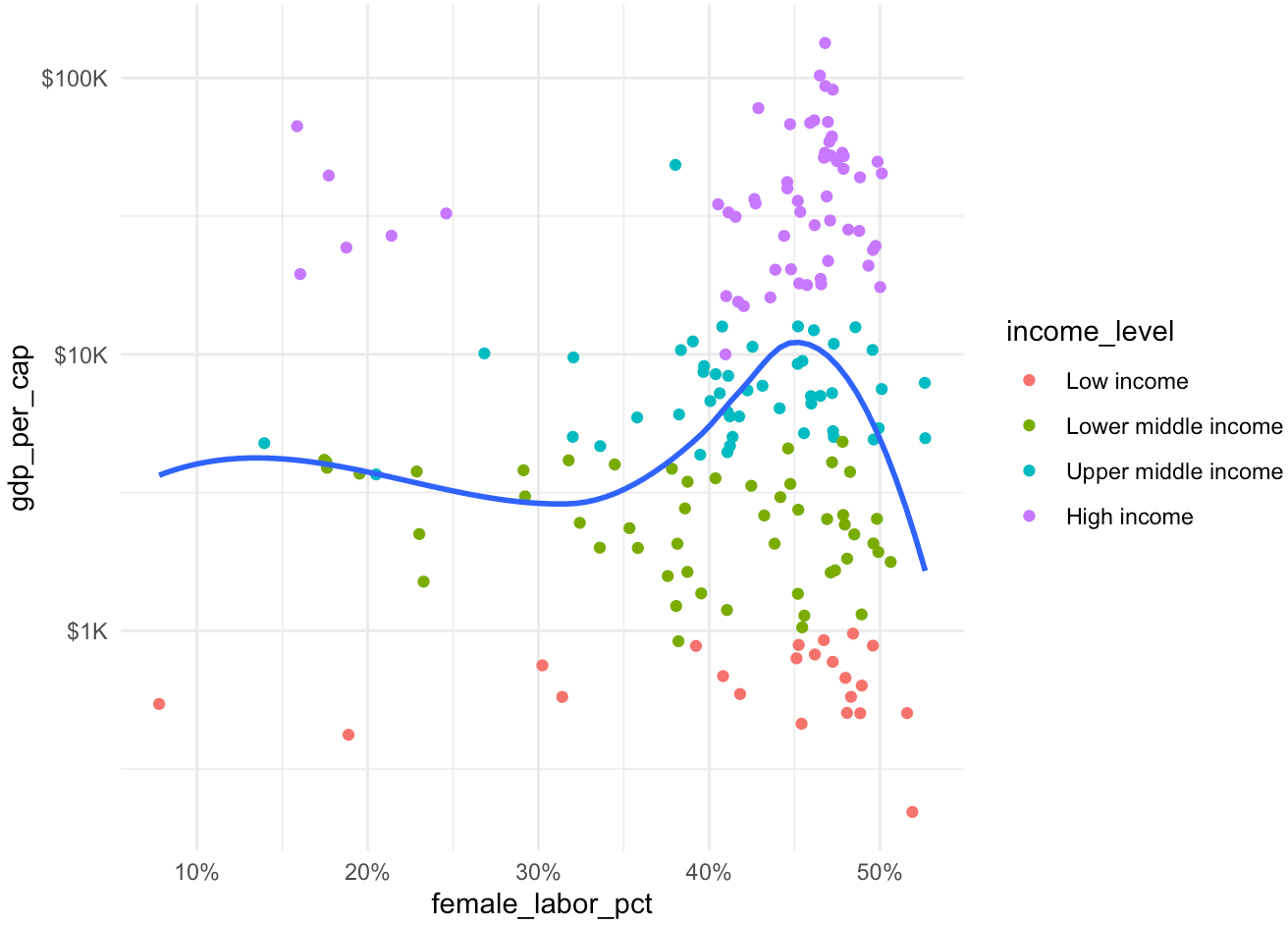
```
  geom_smooth(se = FALSE)
```



Step 2: Your turn

Now, let's modify the scales to make the chart more readable. Log-transform the y-axis and format the labels so they are explicitly identified as percentages and currency.

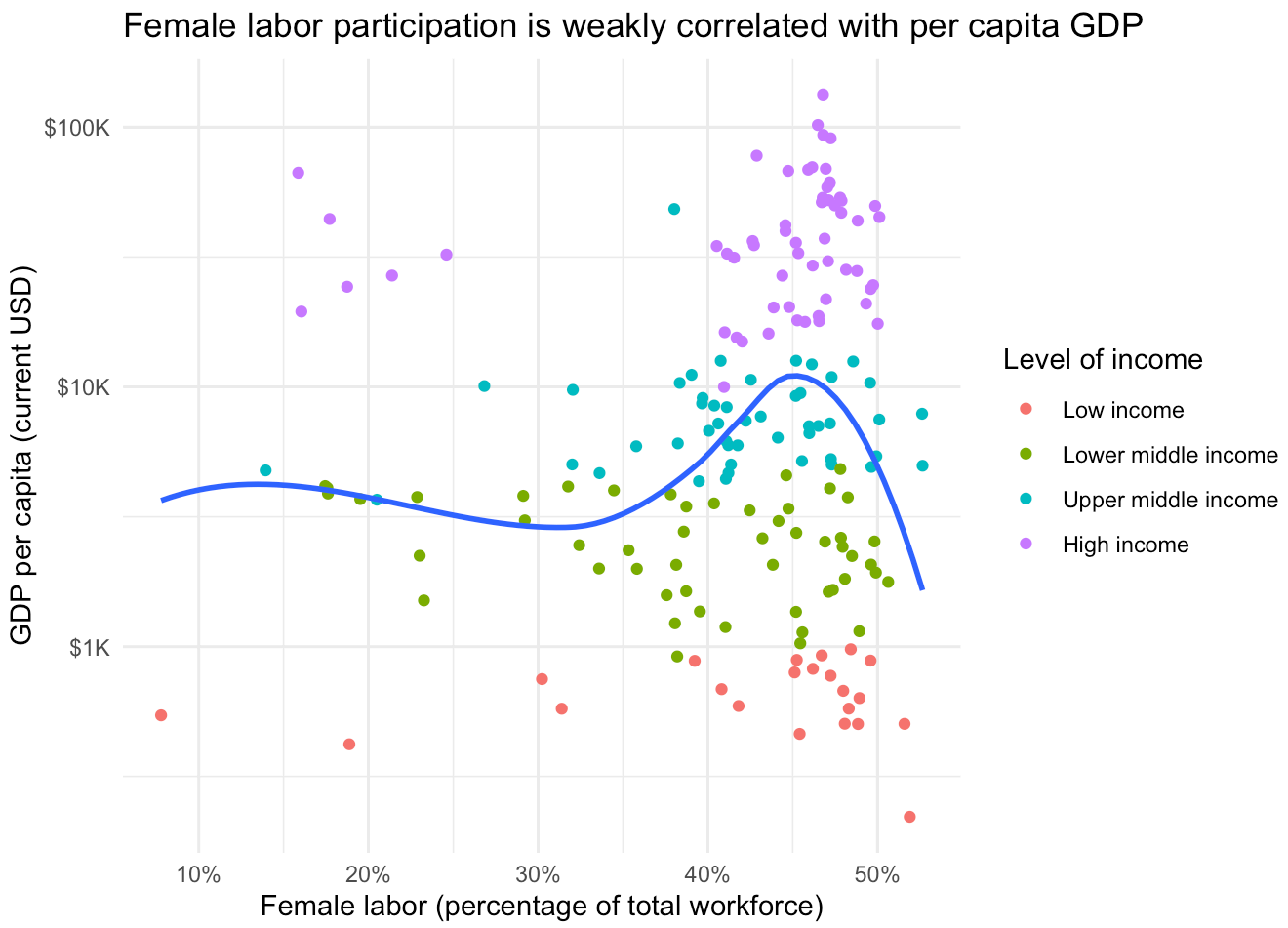
```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level)) +
  geom_smooth(se = FALSE) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_log10(labels = label_currency(scale_cut = cut_short_scale()))
```



Step 3: Your turn

Add human-readable labels for the title, axes, and legend.

```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level)) +
  geom_smooth(se = FALSE) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_log10(labels = label_currency(scale_cut = cut_short_scale())) +
  labs(
    title = "Female labor participation is weakly correlated with per capita GDP",
    x = "Female labor (percentage of total workforce)",
    y = "GDP per capita (current USD)",
    color = "Level of income"
  )
```



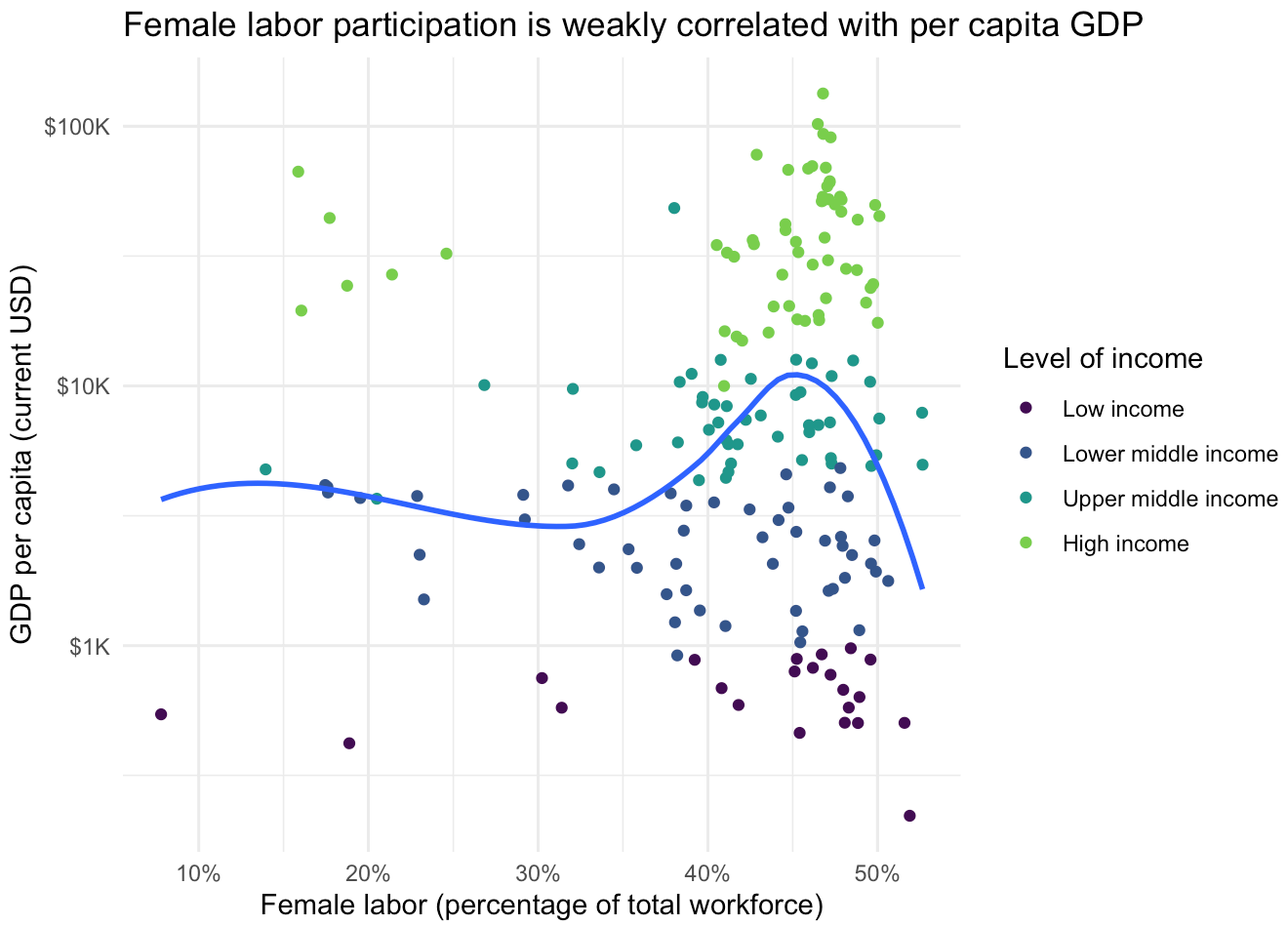
Step 4: Your turn

Use the {viridis} color palette for `income_level`.

> **Tip**
>
> The bright yellow at the end of the palette is hard on the eyes. You can condense the hue at which the color map ends using the `end` argument to the appropriate `scale_color_*()` function.

```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level)) +
  geom_smooth(se = FALSE) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_log10(labels = label_currency(scale_cut = cut_short_scale())) +
  scale_color_viridis_d(end = 0.8) +
  labs(
    title = "Female labor participation is weakly correlated with per capita GDP",
    x = "Female labor (percentage of total workforce)",
    y = "GDP per capita (current USD)",
```

```
    color = "Level of income"
  )
```

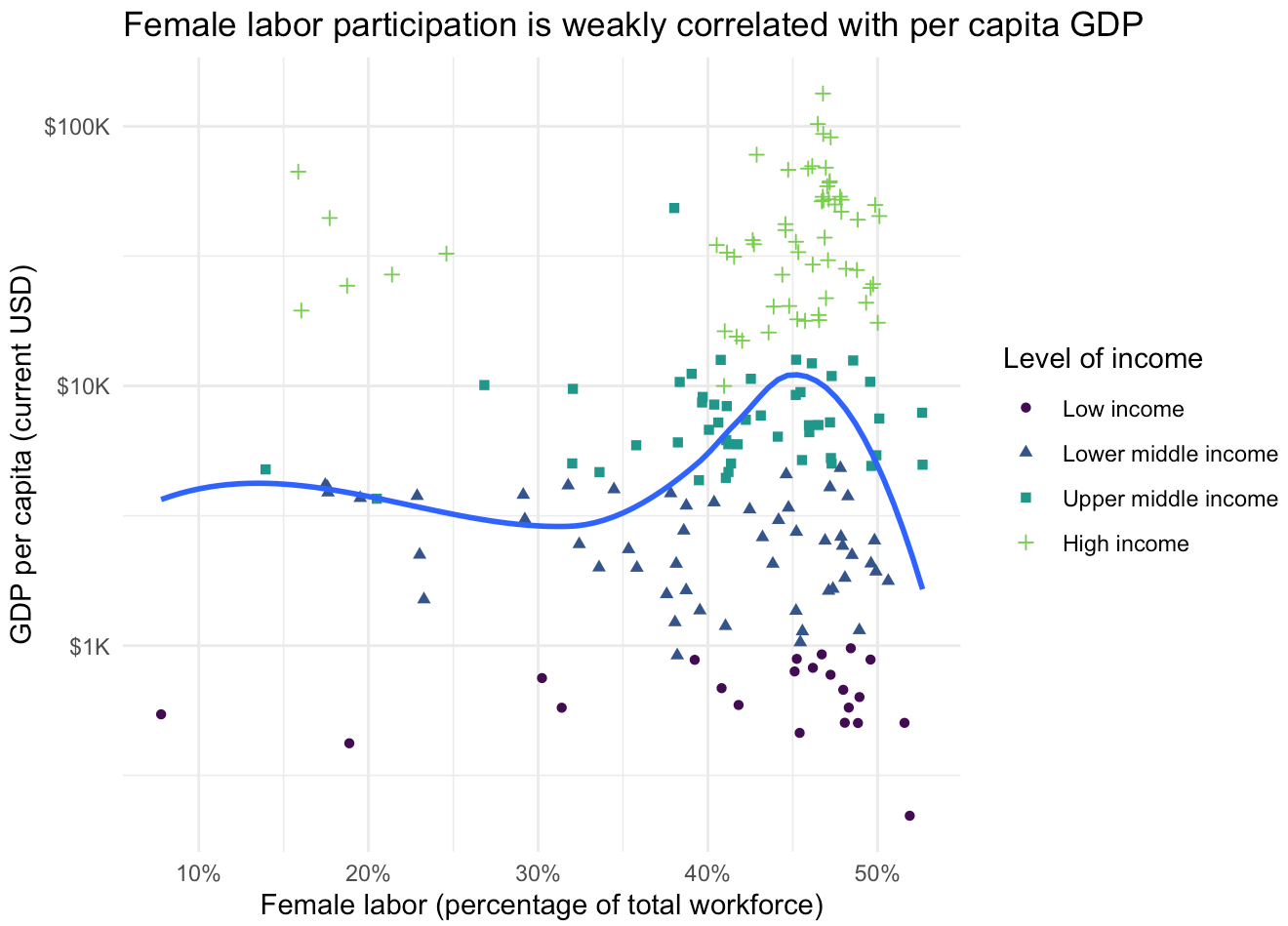


Step 5: Your turn

Double-encode the `income_level` variable by using both color and shape to represent the same variable. Condense the guides so you use a single legend.

```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level, shape = income_level)) +
  geom_smooth(se = FALSE) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_log10(labels = label_currency(scale_cut = cut_short_scale())) +
  scale_color_viridis_d(end = 0.8) +
  labs(
    title = "Female labor participation is weakly correlated with per capita GDP",
    x = "Female labor (percentage of total workforce)",
    y = "GDP per capita (current USD)",
    color = "Level of income",
```

```
    shape = "Level of income"
  )
```

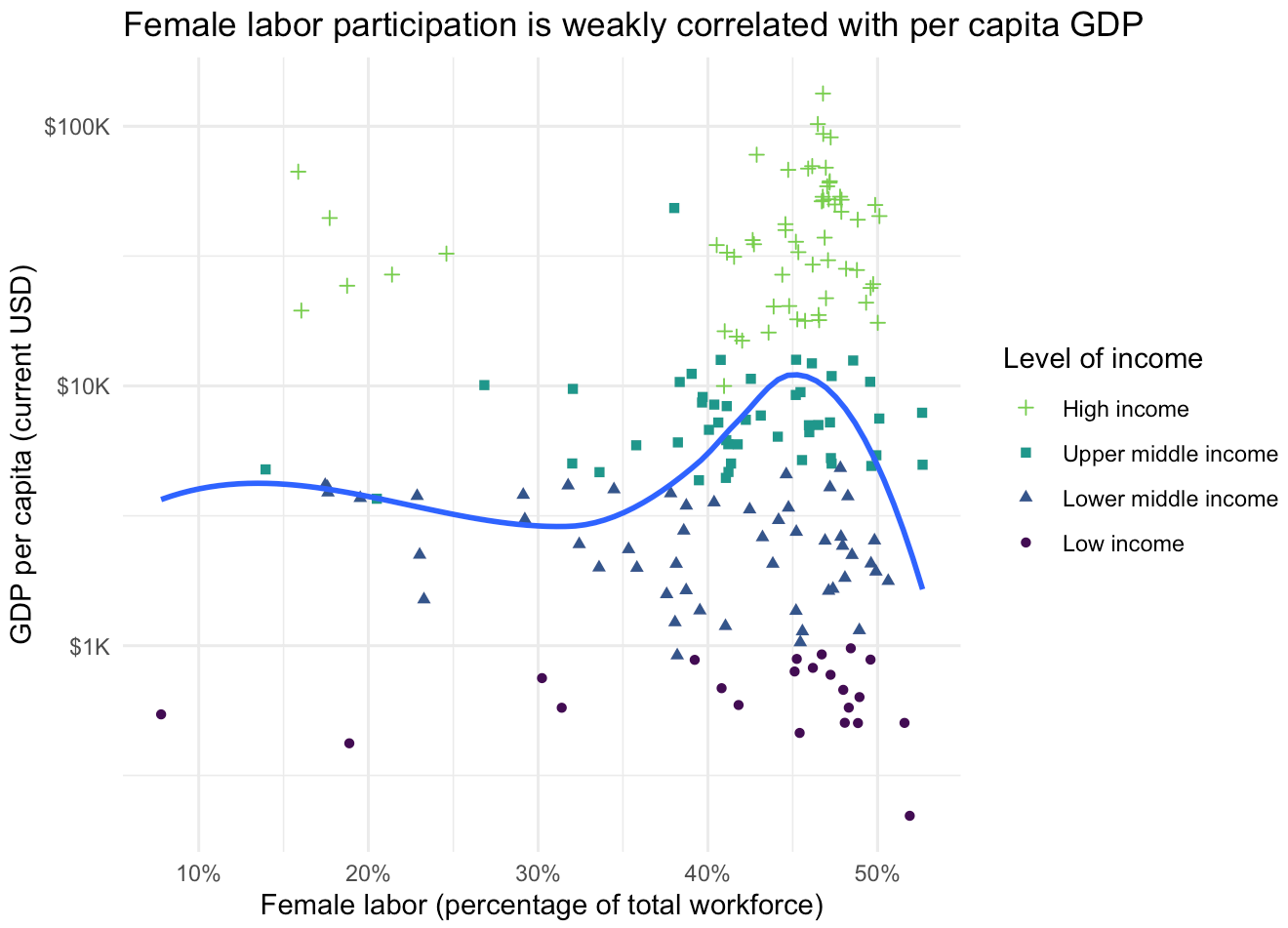


Step 6: Your turn

It's annoying that the order of the values in the legend are opposite from how the income levels are ordered in the chart. Reverse the order of the values in the legend so they correspond to the ordering on the y-axis.

```
ggplot(data = world_bank, mapping = aes(x = female_labor_pct, y = gdp_per_cap)) +
  geom_point(mapping = aes(color = income_level, shape = income_level)) +
  geom_smooth(se = FALSE) +
  scale_x_continuous(labels = label_percent(scale = 1)) +
  scale_y_log10(labels = label_currency(scale_cut = cut_short_scale())) +
  scale_color_viridis_d(end = 0.8, guide = guide_legend(reverse = TRUE)) +
  scale_shape_discrete(guide = guide_legend(reverse = TRUE)) +
  labs(
    title = "Female labor participation is weakly correlated with per capita GDP",
    x = "Female labor (percentage of total workforce)",
    y = "GDP per capita (current USD)",
    color = "Level of income",
```

```
    shape = "Level of income"
  )
```



Session information

Footnotes

1. Note that the income level is based on the GNI per capita, which is strongly correlated with GDP per capita, but not exactly the same. ↩