Data Visualization with ggplot2

Natalia Vasilenok

Math Camp
Department of Political Science
Stanford University

September 18, 2024

Why visualize?

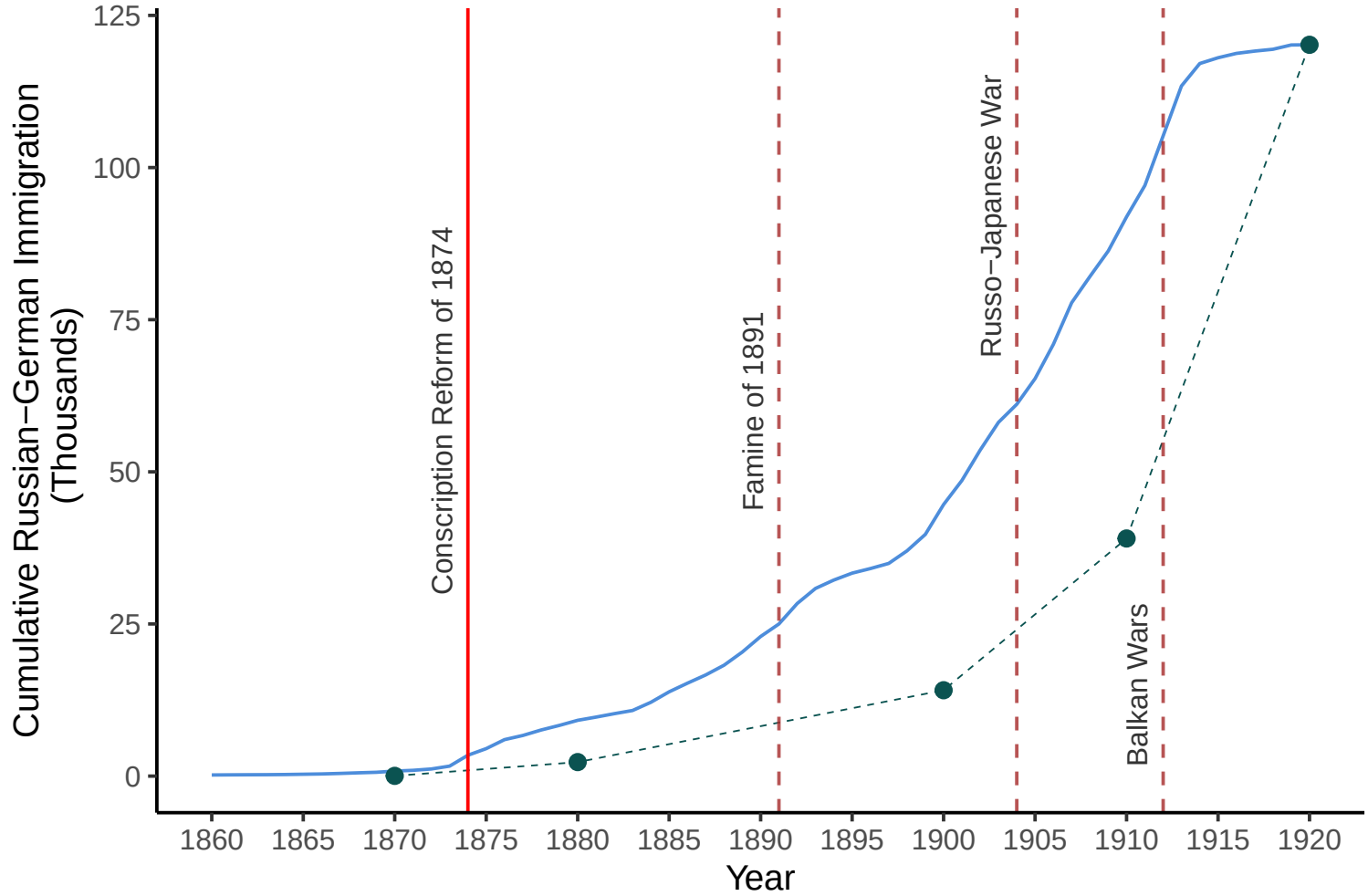
On top of all our struggles with collecting and processing data, why do we need to plot it?

Why visualize?

On top of all our struggles with collecting and processing data, why do we need to plot it?

- Representing data in a visual format can help us as researchers to make sense of it
- Clear and well-thought-out plots can help strengthen and deliver your argument

How visualize?



What do you like about this plot? What could have been done better?

Why ggplot2?

- Base R has the set of built-in plotting functions, such as `plot()` or `hist()`, which might be useful when you need to quickly look at your data
- If you need to come up with a highly customized figure, the `ggplot2` package is your choice
- With `ggplot2`, you build up a plot layer by layer, and each layer can come from a different dataset
  - `ggplot2` works with datasets and not with separate vectors as base R does; some people consider it a downside
- Plots made with `ggplot2` can be really pretty

What does a ggplot consist of?

- Data

What does a ggplot consist of?

- Data
- **Aes**thetic mappings that specify a relation between data and its visual properties, such as axes, colors or shapes
  - The most common aesthetics are `x`, `y`, `color`, and `group`

What does a ggplot consist of?

- Data
- **Aes**thetic mappings that specify a relation between data and its visual properties, such as axes, colors or shapes
  - The most common aesthetics are `x`, `y`, `color`, and `group`
- **Layers** as collections of **geom**etric elements and statistical transformations of your data
  - Layers display data on a plot

What does a ggplot consist of?

- Data
- **Aes**thetic mappings that specify a relation between data and its visual properties, such as axes, colors or shapes
  - The most common aesthetics are `x`, `y`, `color`, and `group`
- **Layers** as collections of **geom**etric elements and statistical transformations of your data
  - Layers display data on a plot
- `Scales` that **customize** aesthetic elements
  - Each aesthetic is associated with a separate scale

What does a ggplot consist of?

- Data
- **Aes**thetic mappings that specify a relation between data and its visual properties, such as axes, colors or shapes
  - The most common aesthetics are `x`, `y`, `color`, and `group`
- **Layers** as collections of **geom**etric elements and statistical transformations of your data
  - Layers display data on a plot
- `Scales` that **customize** aesthetic elements
  - Each aesthetic is associated with a separate scale
- A `facet` specifies how to break up and display **subsets** of data

What does a ggplot consist of?

- Data
- **Aes**thetic mappings that specify a relation between data and its visual properties, such as axes, colors or shapes
  - The most common aesthetics are `x`, `y`, `color`, and `group`
- **Layers** as collections of **geom**etric elements and statistical transformations of your data
  - Layers display data on a plot
- `Scales` that **customize** aesthetic elements
  - Each aesthetic is associated with a separate scale
- A `facet` specifies how to break up and display **subsets** of data
- A `theme` controls elements of the plot not associated with data, such as the font size, background color or the location of a legend
  - `ggplot2` has a large variety of pre-specified themes
  - You can also create your own theme

Base layer

```
ggplot2(data = df,
        aes(x = x, y = y))
```

Geoms

The `geom` function adds a layer to a plot and specifies its type

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot
- `geom_boxplot()` creates a boxplot

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot
- `geom_boxplot()` creates a boxplot
- `geom_bar(stat = "identity")` draws a bar plot

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot
- `geom_boxplot()` creates a boxplot
- `geom_bar(stat = "identity")` draws a bar plot
- `geom_histogram()` makes a histogram

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot
- `geom_boxplot()` creates a boxplot
- `geom_bar(stat = "identity")` draws a bar plot
- `geom_histogram()` makes a histogram
- `geom_polygon()` and `geom_sf()` can be used to draw a map

Geoms

The `geom` function adds a layer to a plot and specifies its type

- `geom_point()` produces scatterplots
- `geom_line()` makes a line plot
- `geom_boxplot()` creates a boxplot
- `geom_bar(stat = "identity")` draws a bar plot
- `geom_histogram()` makes a histogram
- `geom_polygon()` and `geom_sf()` can be used to draw a map

You can combine different geoms on the same plot; for example, you can add points to a line plot

Adding a geom

```
ggplot2(data = df, aes(x = x, y = y)) +
  geom_line()
```

Scales

Scales control the appearance of different **aes**thetics

Scales

Scales control the appearance of different **aes**thetics

- Position scales control the x and y axes
  - `scale_x_NAME()`, `scale_y_NAME()`
  - `NAME` refers to data type and most often is `continuous` or `discrete`
  - You can specify axes limits, breaks (the locations of the axis tick marks), or the labels of the tick marks

Scales

Scales control the appearance of different **aes**thetics

- Position scales control the x and y axes
  - `scale_x_NAME()`, `scale_y_NAME()`
  - `NAME` refers to data type and most often is `continuous` or `discrete`
  - You can specify axes limits, breaks (the locations of the axis tick marks), or the labels of the tick marks
- Color scales map values to colors and produce legends
  - `scale_color_NAME()`
  - `scale_fill_NAME()`
  - `NAME` is either `manual` or pre-specified palette sets, such as `brewer` for discrete values or `distiller` for continuous values

Controlling scales

```
ggplot2(data = df, aes(x = x, y = y)) +
  scale_x_continuous(limits = c(0, 1)) +
  geom_line()
```

Data for today

AJR (2001) The Colonial Origins of Comparative Development

- `logem4` the logarithm of the settler mortality rates per thousand
- `avexpr` the average value of an index of protection against expropriation between 1985 and 1995
- `f_brit` the dummy variable that takes on a value of one if a country was a British colony
- `muslim80` the fraction of the country population that is Muslim

Link to replication files

Data for today

AJR (2001) The Colonial Origins of Comparative Development

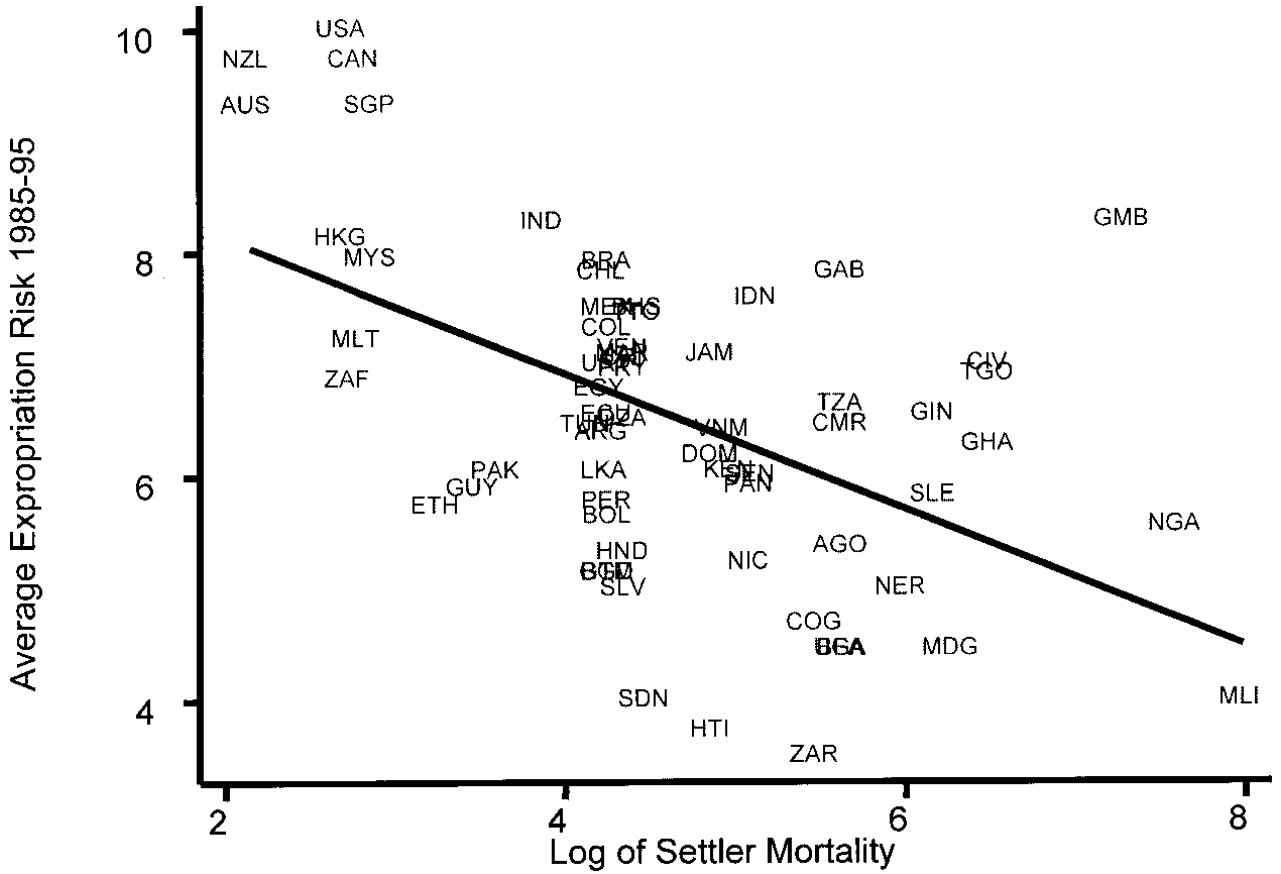


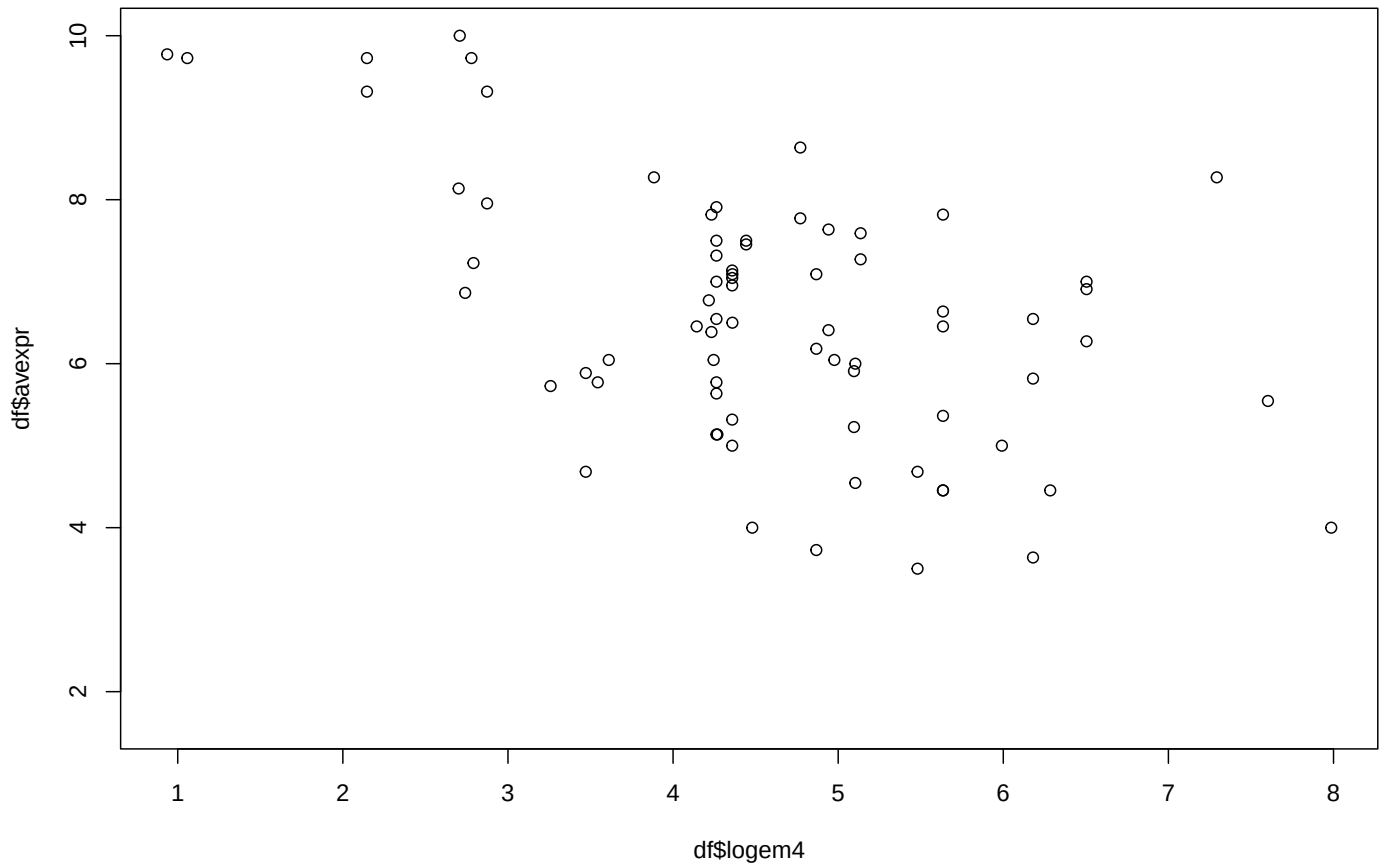
Figure 3: First-Stage Relationship between Settler Mortality and Expropriation Risk

Agenda

1. Scatterplots
2. Histograms
3. Intro to maps

Scatterplot in base R

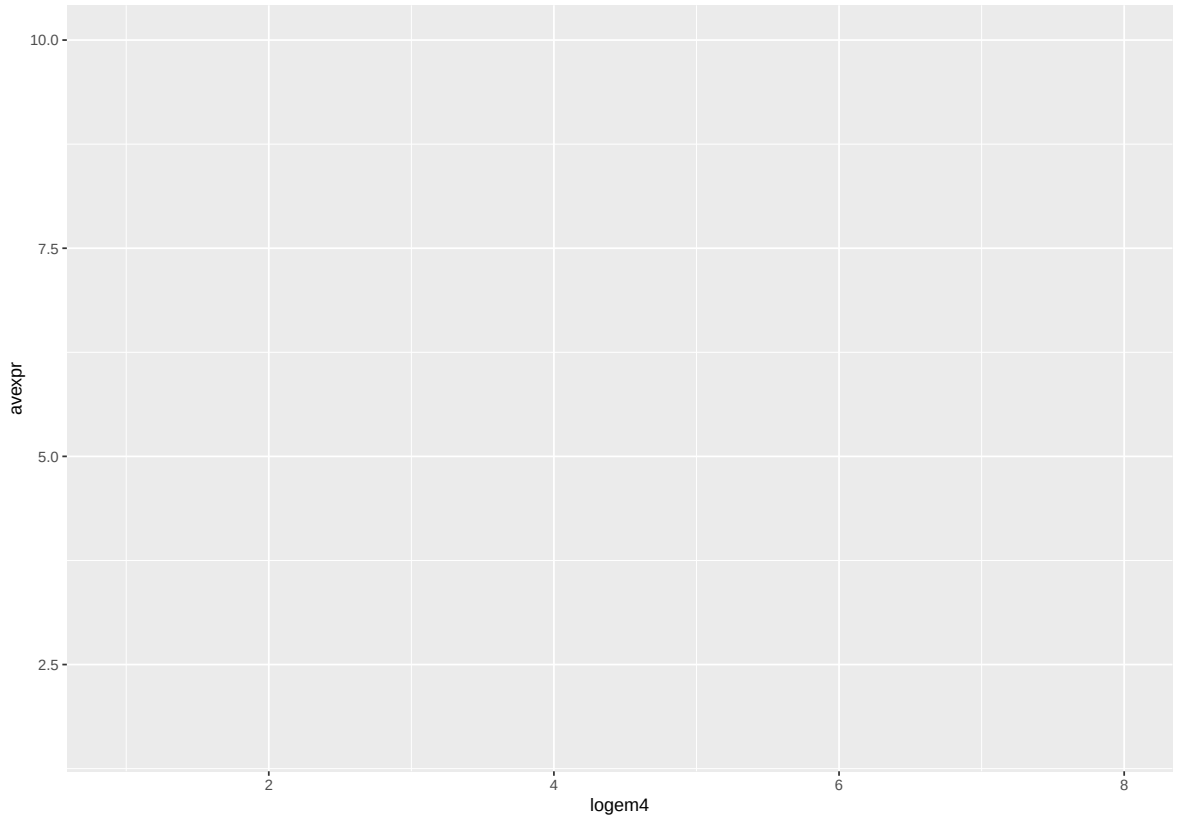
```
library(foreign)
df = read.dta("ajr.dta")
plot(x = df$logem4, y = df$avexpr)
```



Scatterplot

Base layer

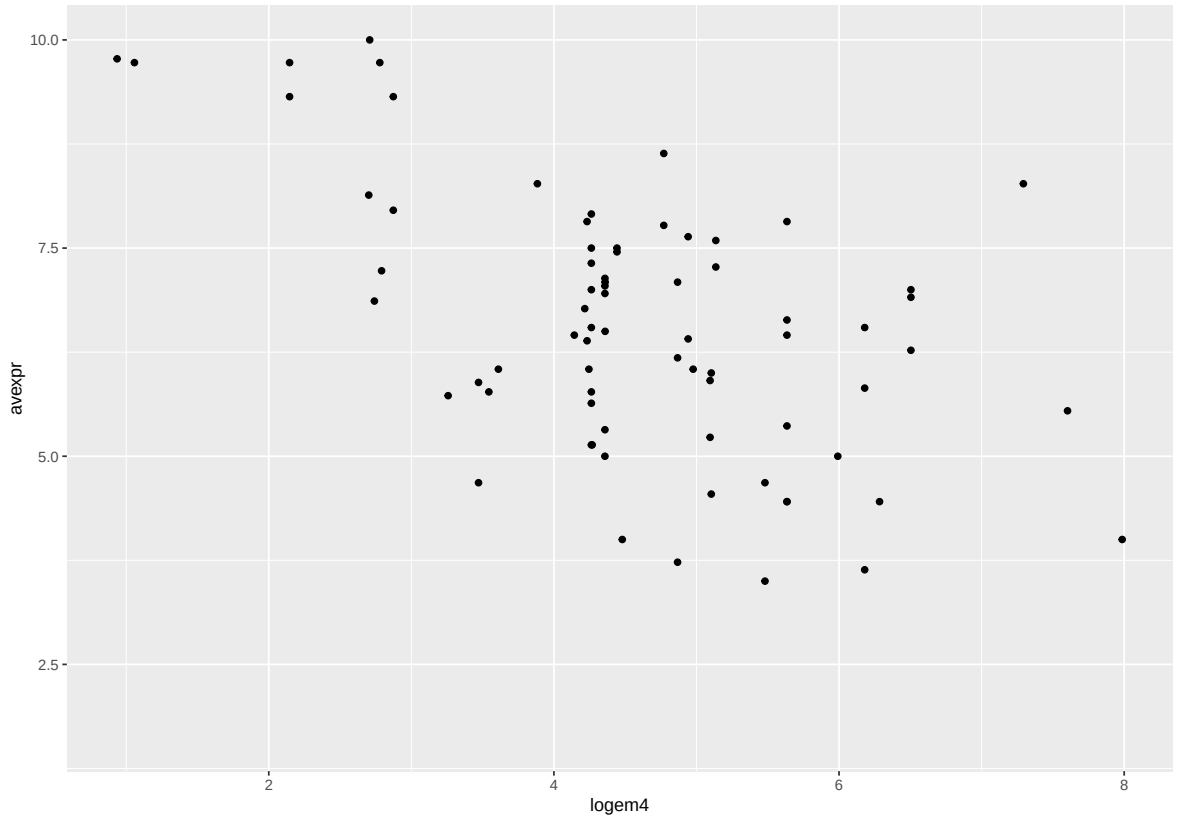
```
library(ggplot2)
ggplot(data = df, aes(x = logem4, y = avexpr))
```



Scatterplot

Adding a geom

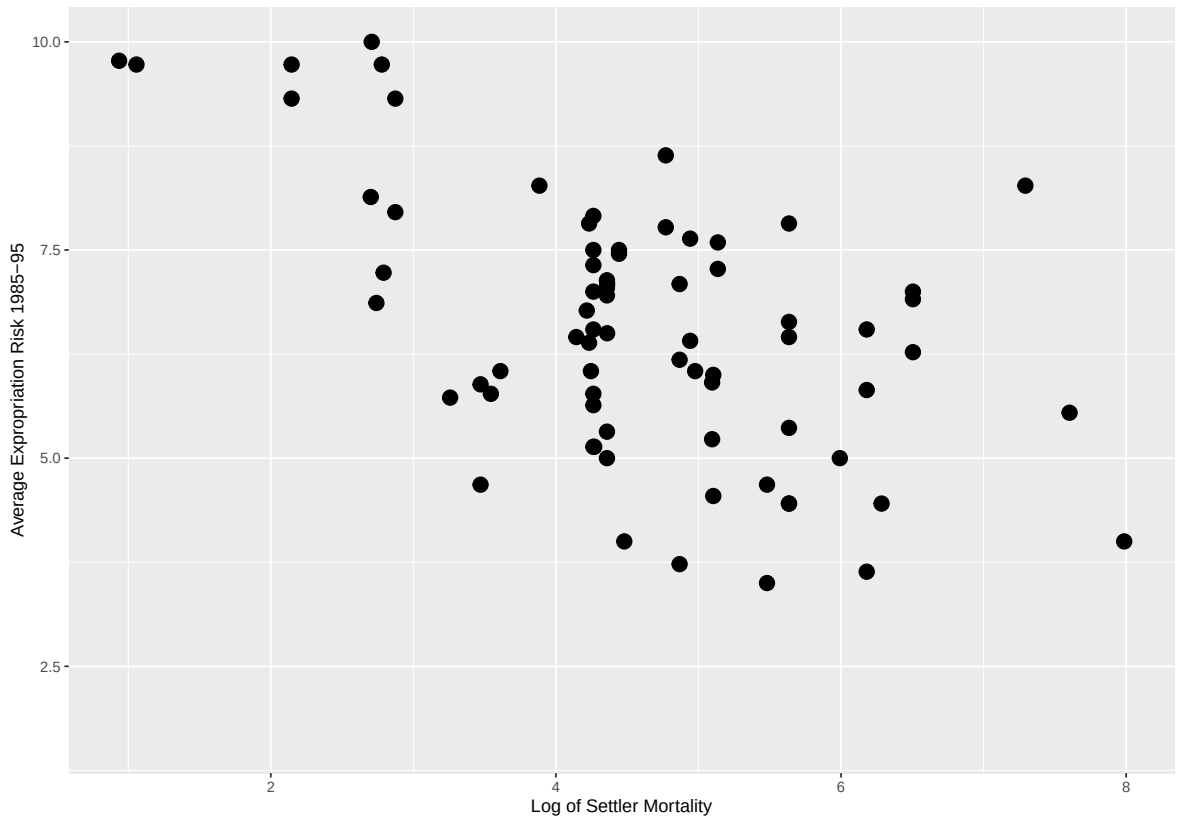
```
ggplot(data = df, aes(x = logem4, y = avexpr)) +
  geom_point()
```



Scatterplot

Labeling axes

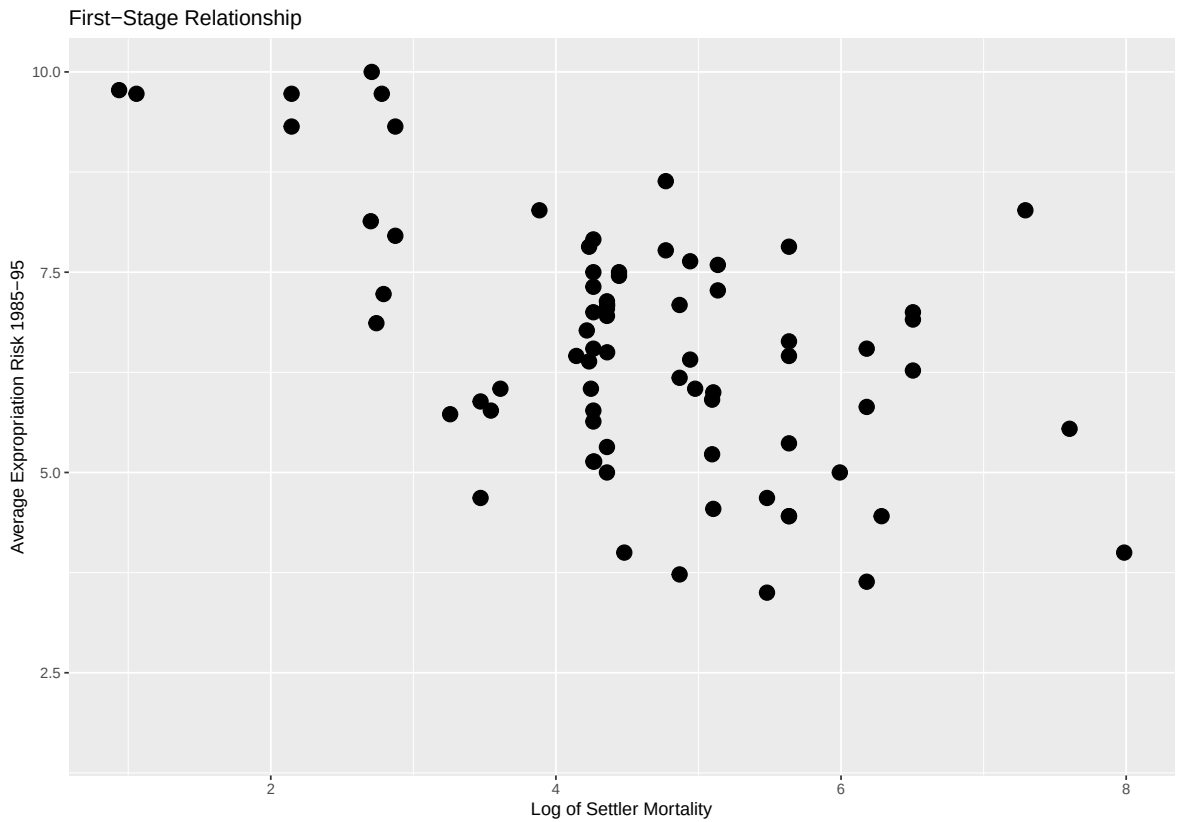
```
ggplot(data = df, aes(x = logem4, y = avexpr)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95")
```



Scatterplot

Adding the title

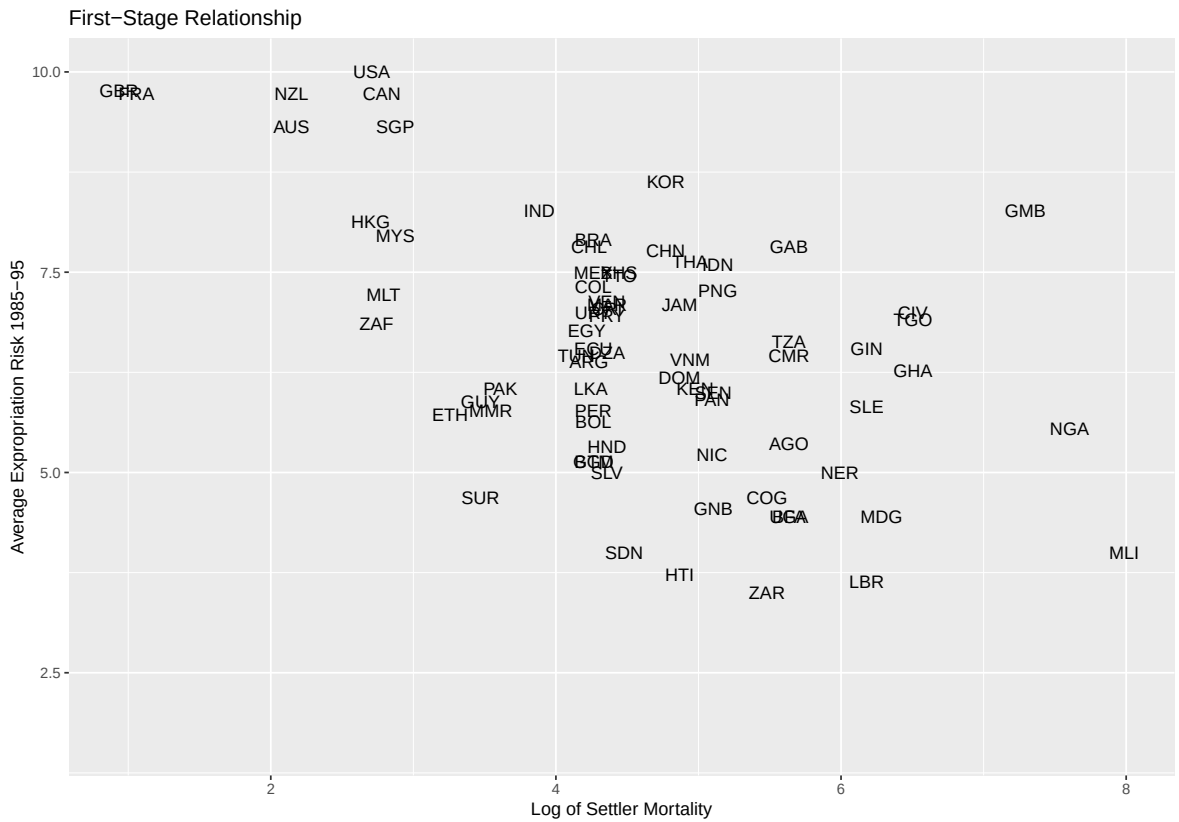
```
ggplot(data = df, aes(x = logem4, y = avexpr)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship")
```



Scatterplot

Labeling points

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text() +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship")
```



Scatterplot

Increasing label size

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship")
```

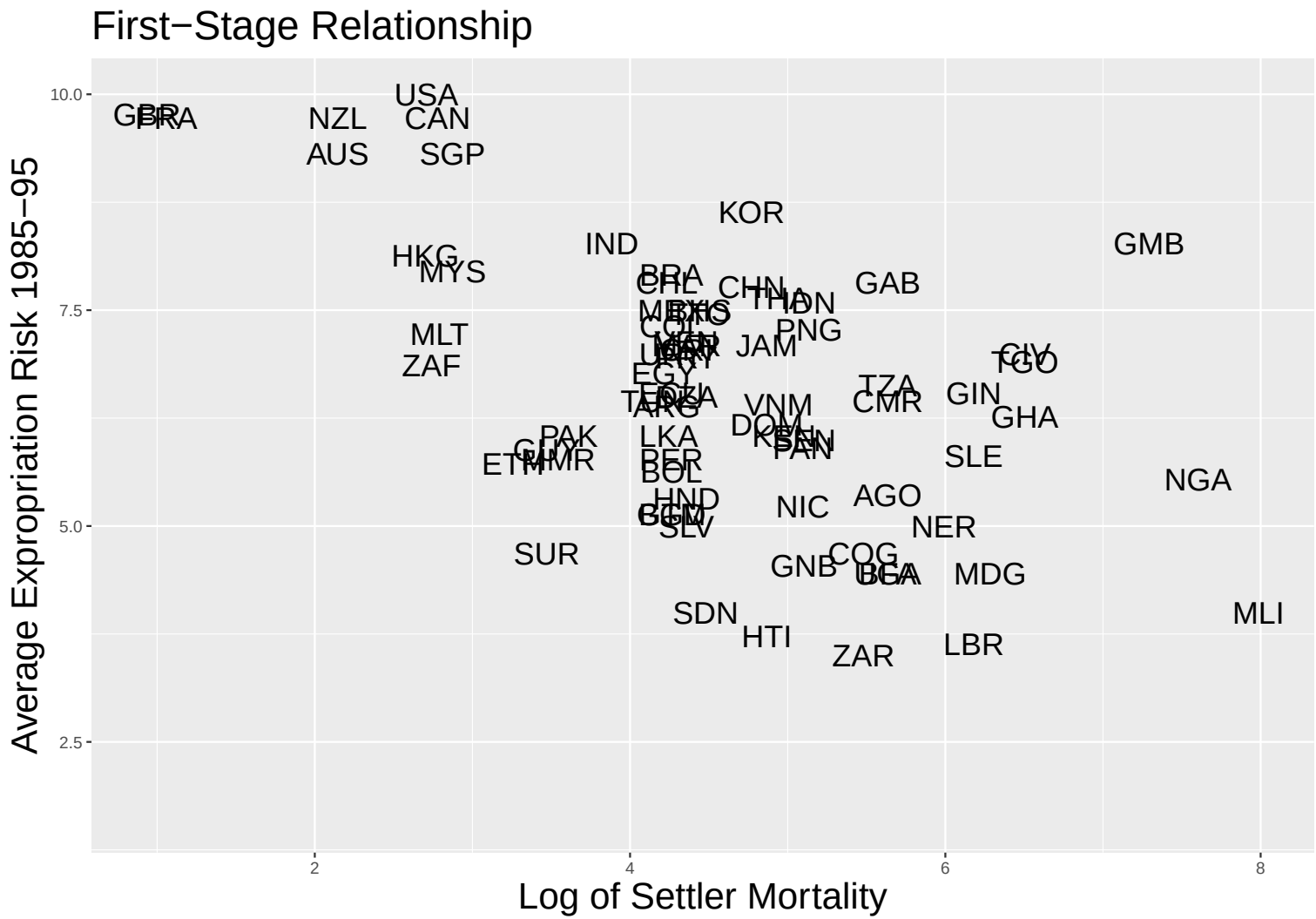
Scatterplot

Increasing title and axes labels font size

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20))
```

Scatterplot

Increasing title and axes labels font size



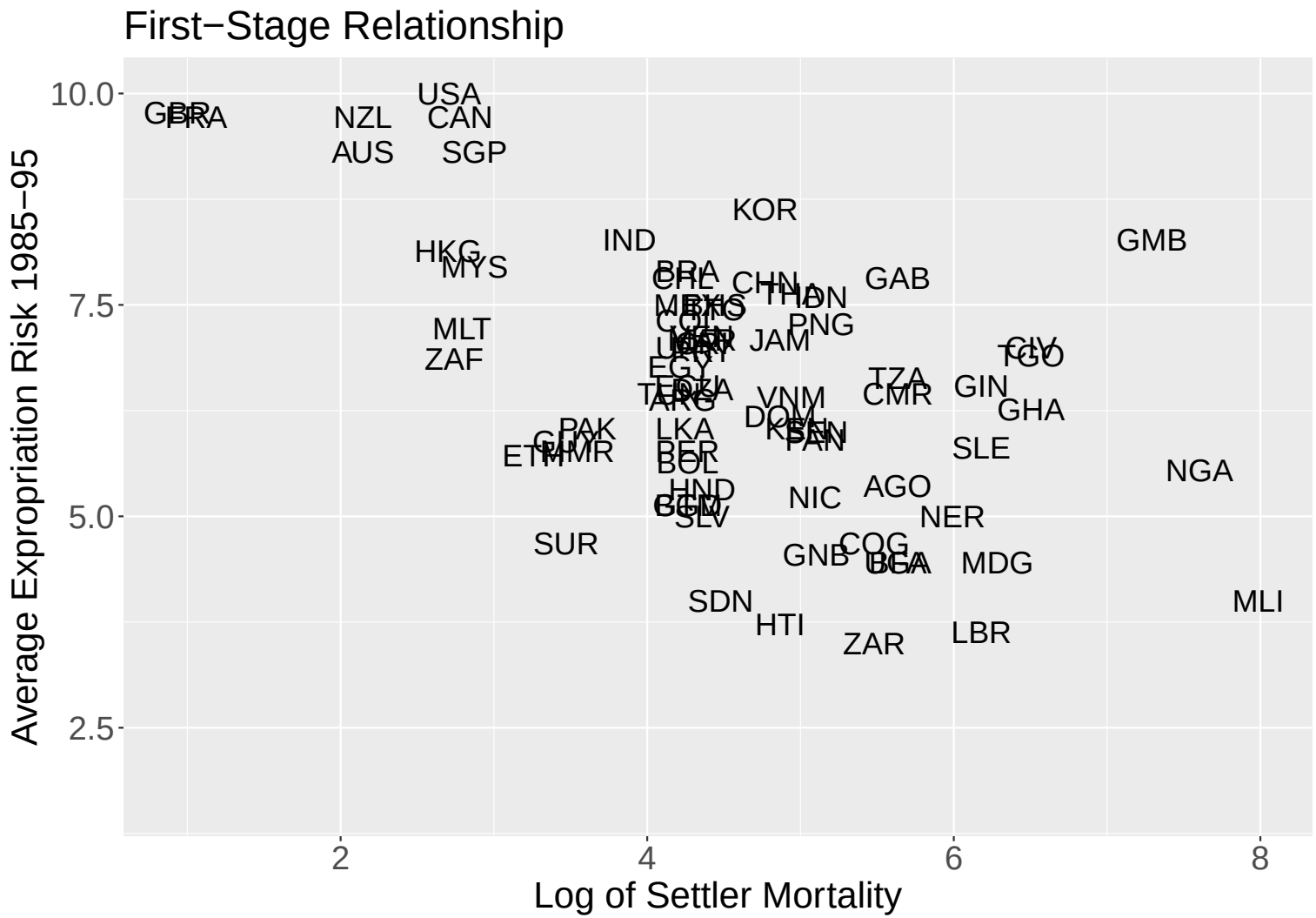
Scatterplot

Increasing axes tick mark labels font size

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

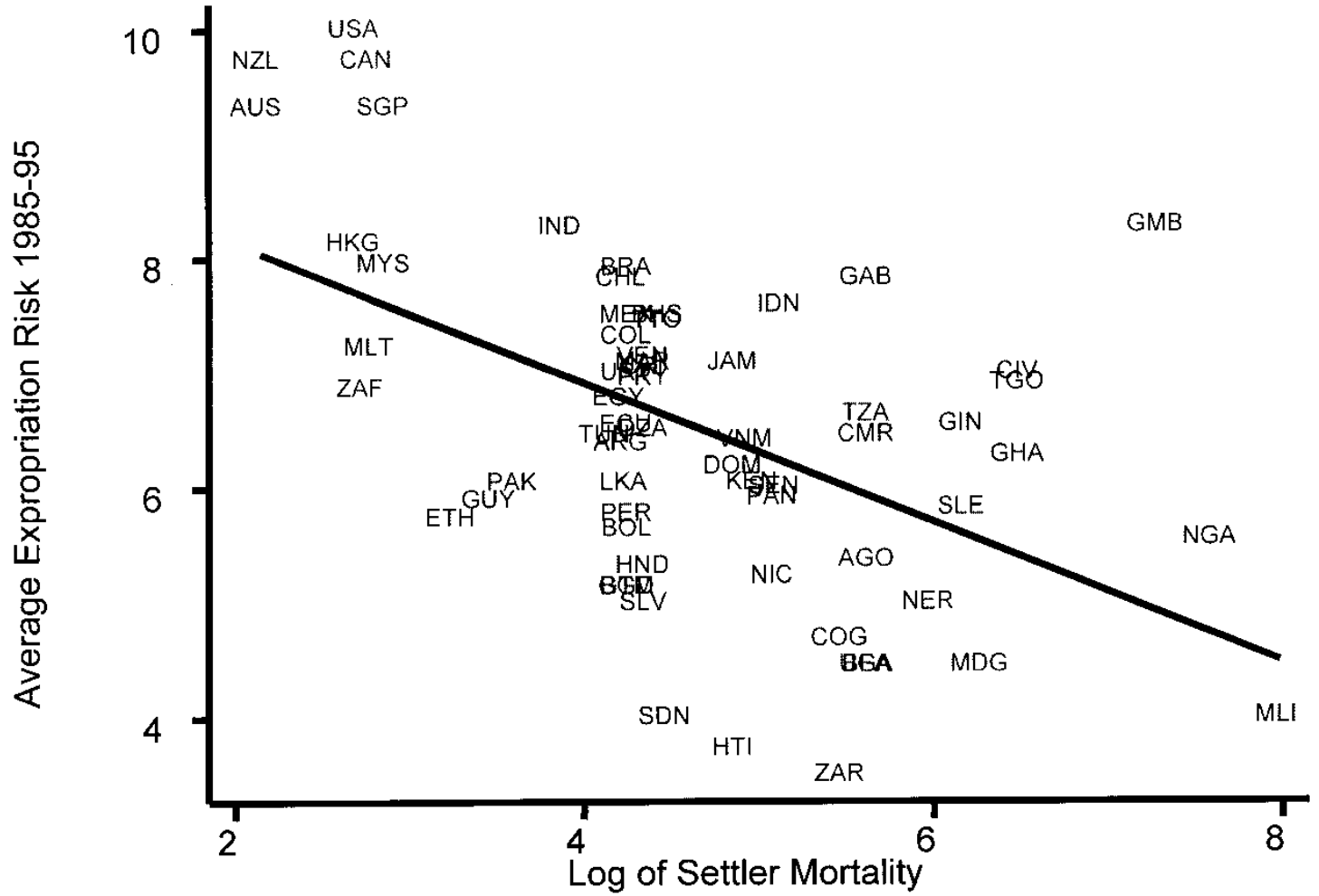
Scatterplot

Increasing axes tick mark labels font size



Scatterplot

Working with axes



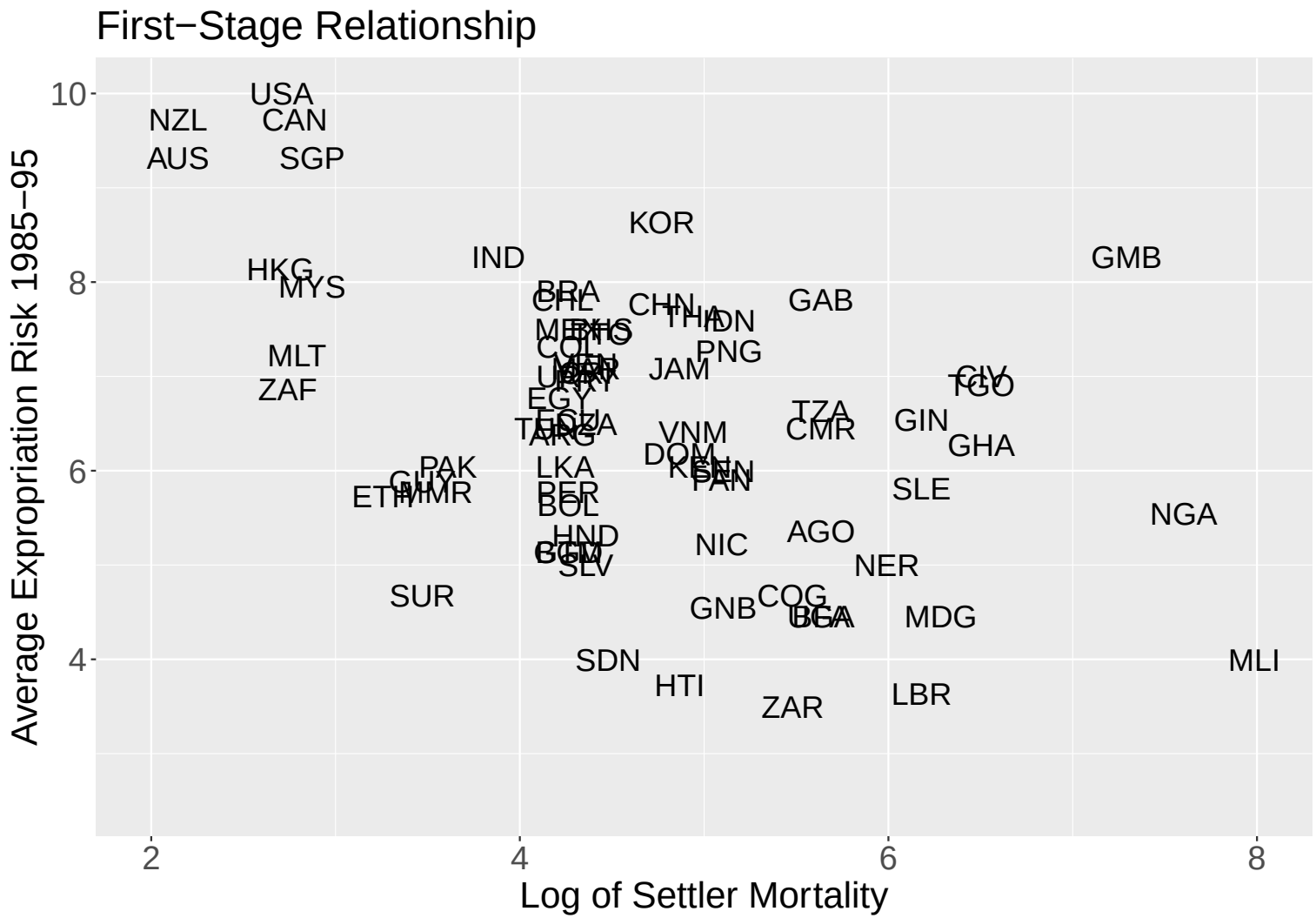
Scatterplot

Working with axes

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18)) +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10))
```

Scatterplot

Working with axes



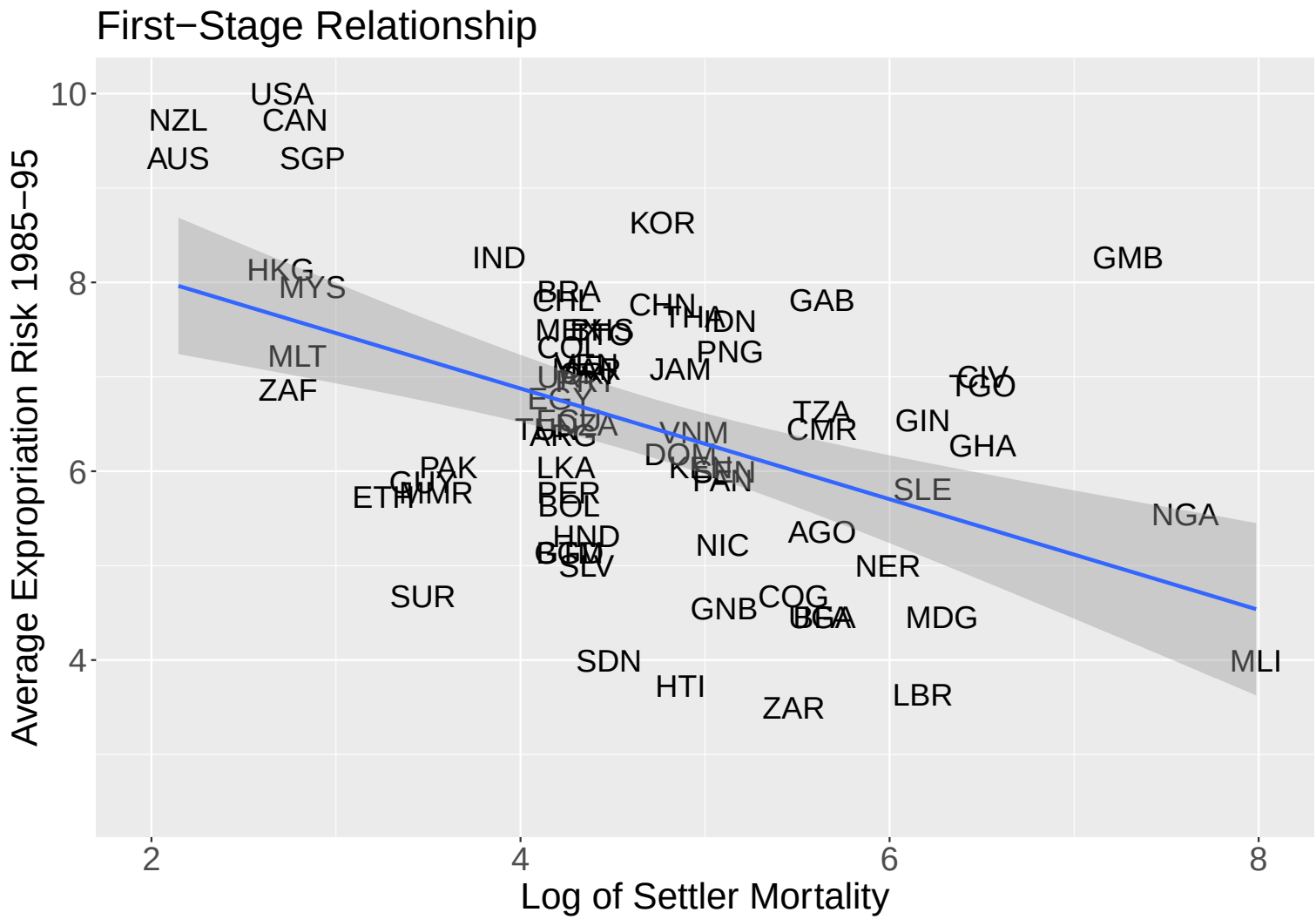
Scatterplot

Overlaying linear fit

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18)) +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm)
```

Scatterplot

Overlaying linear fit



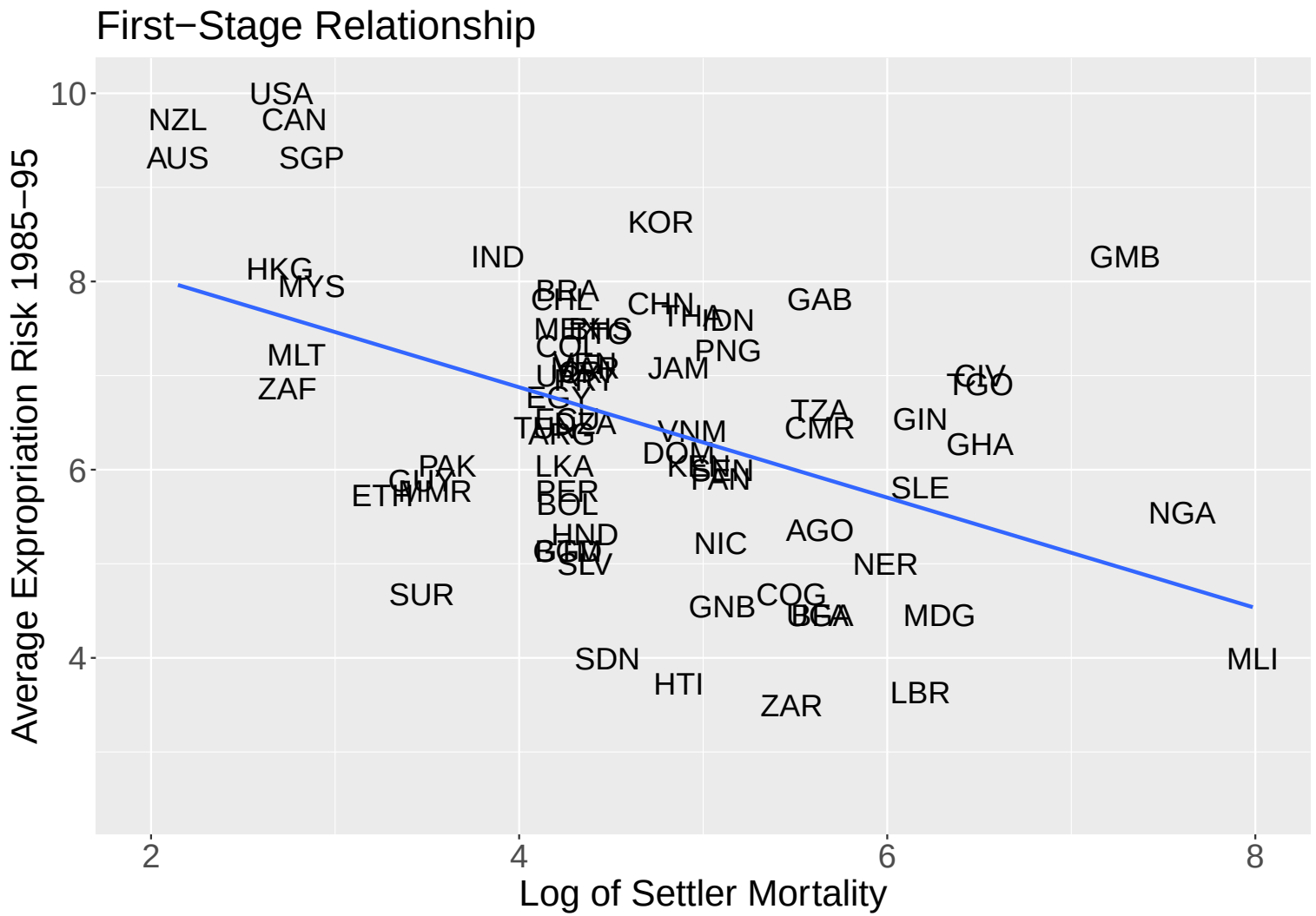
Scatterplot

Overlaying linear fit

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18)) +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE)
```

Scatterplot

Overlaying linear fit



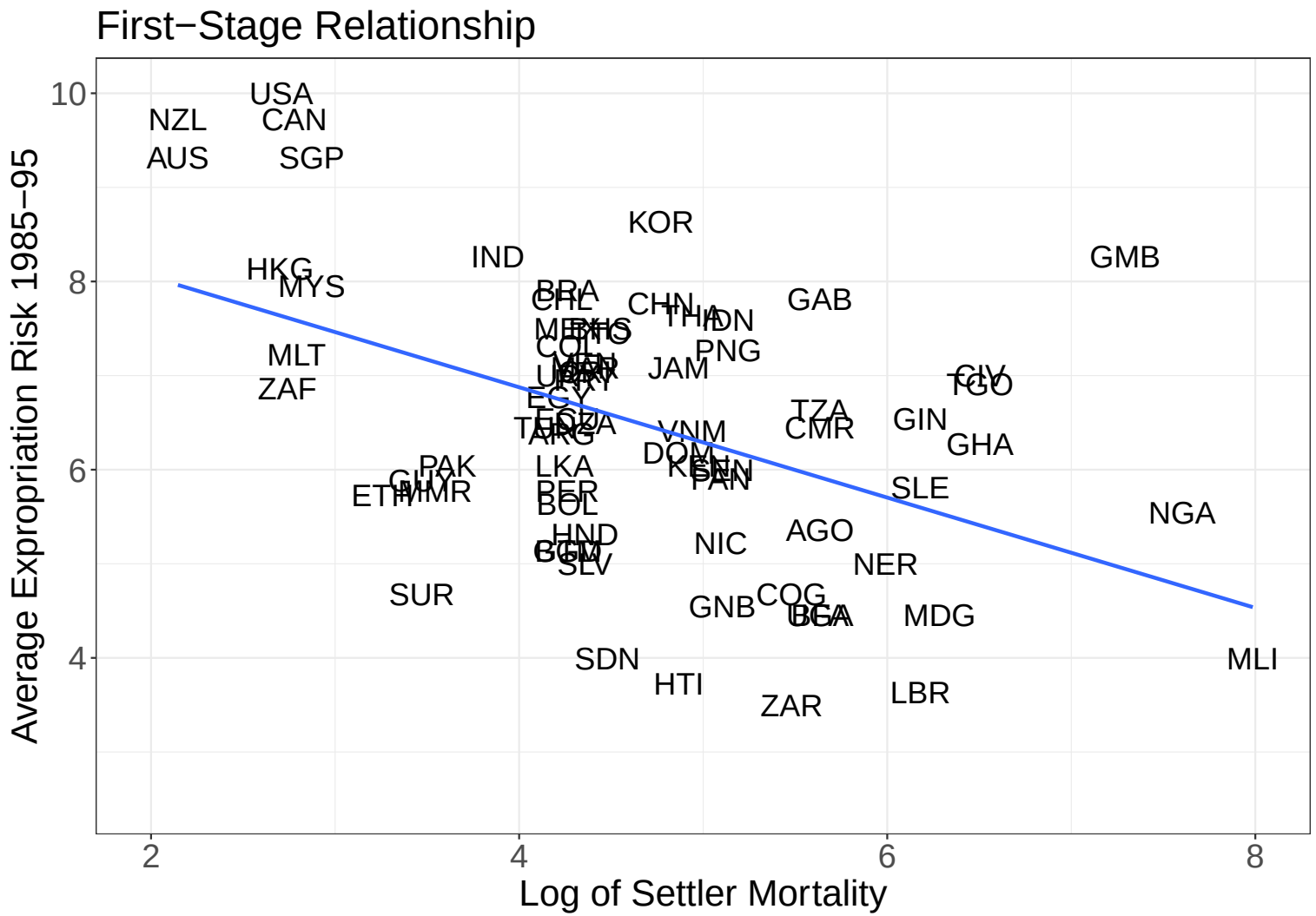
Scatterplot

Themes: `theme_bw()`

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Themes: `theme_bw()`



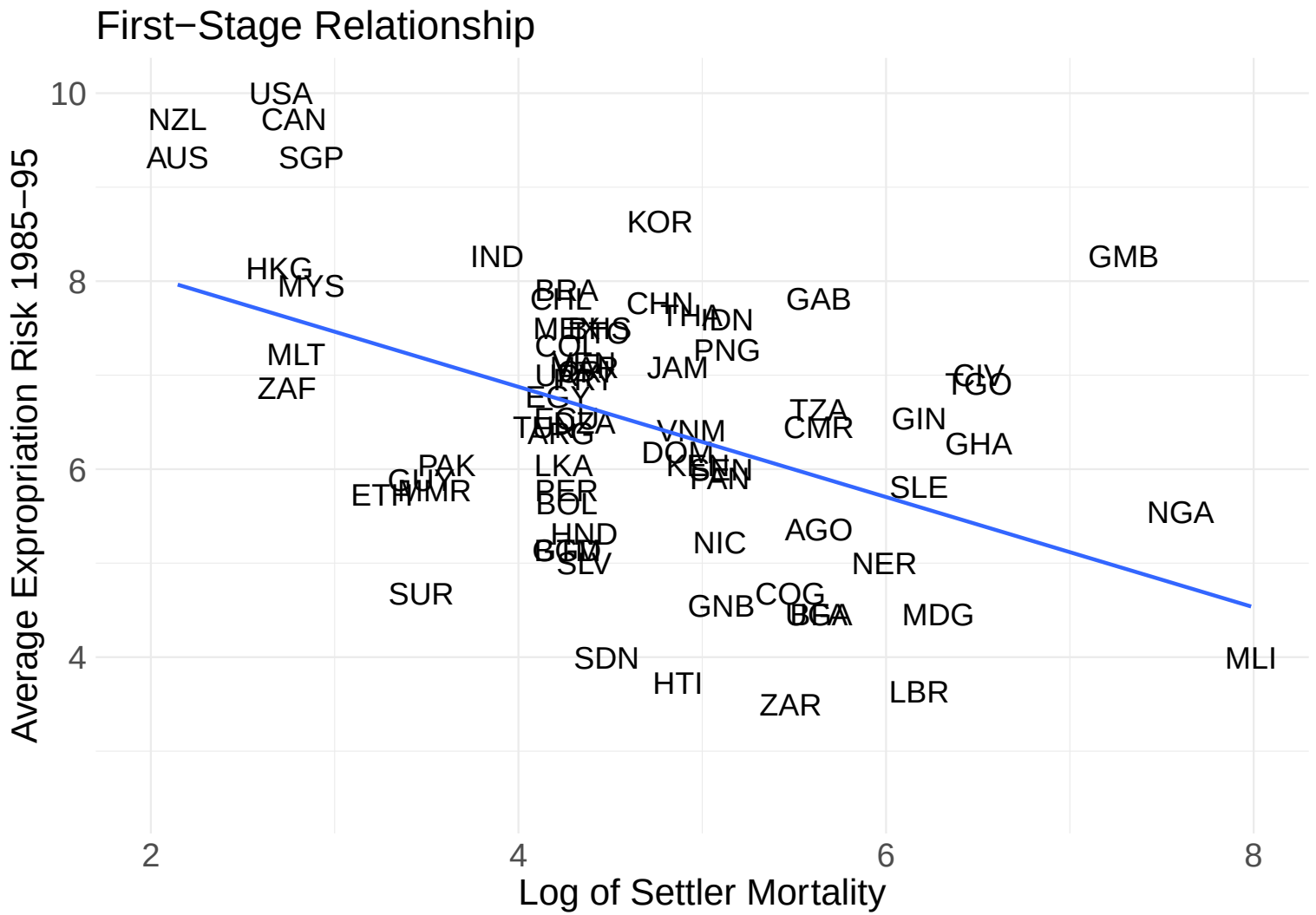
Scatterplot

Themes: `theme_minimal()`

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_minimal() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Themes: `theme_minimal()`



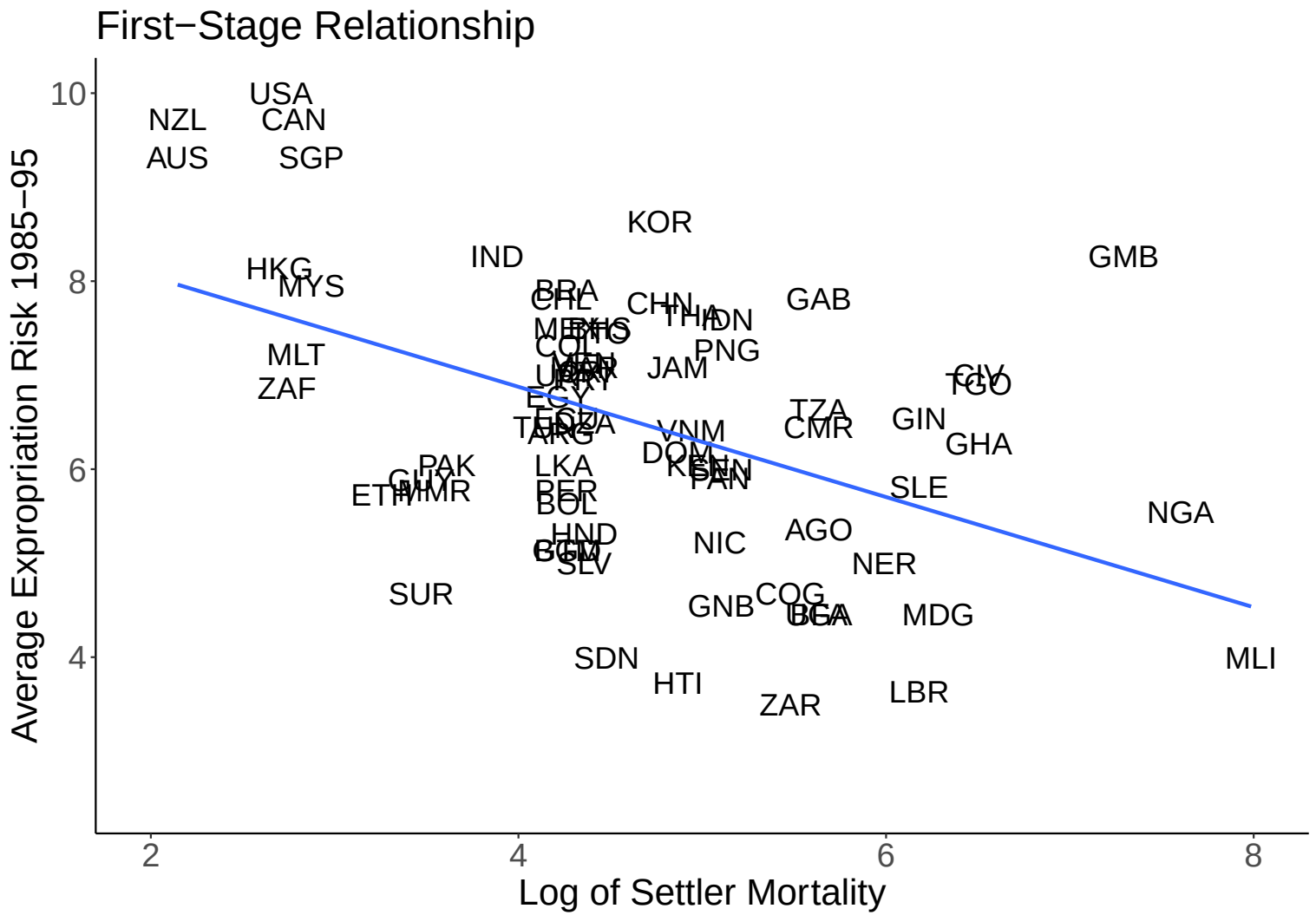
Scatterplot

Themes: `theme_classic()`

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_classic() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Themes: `theme_classic()`



Scatterplot

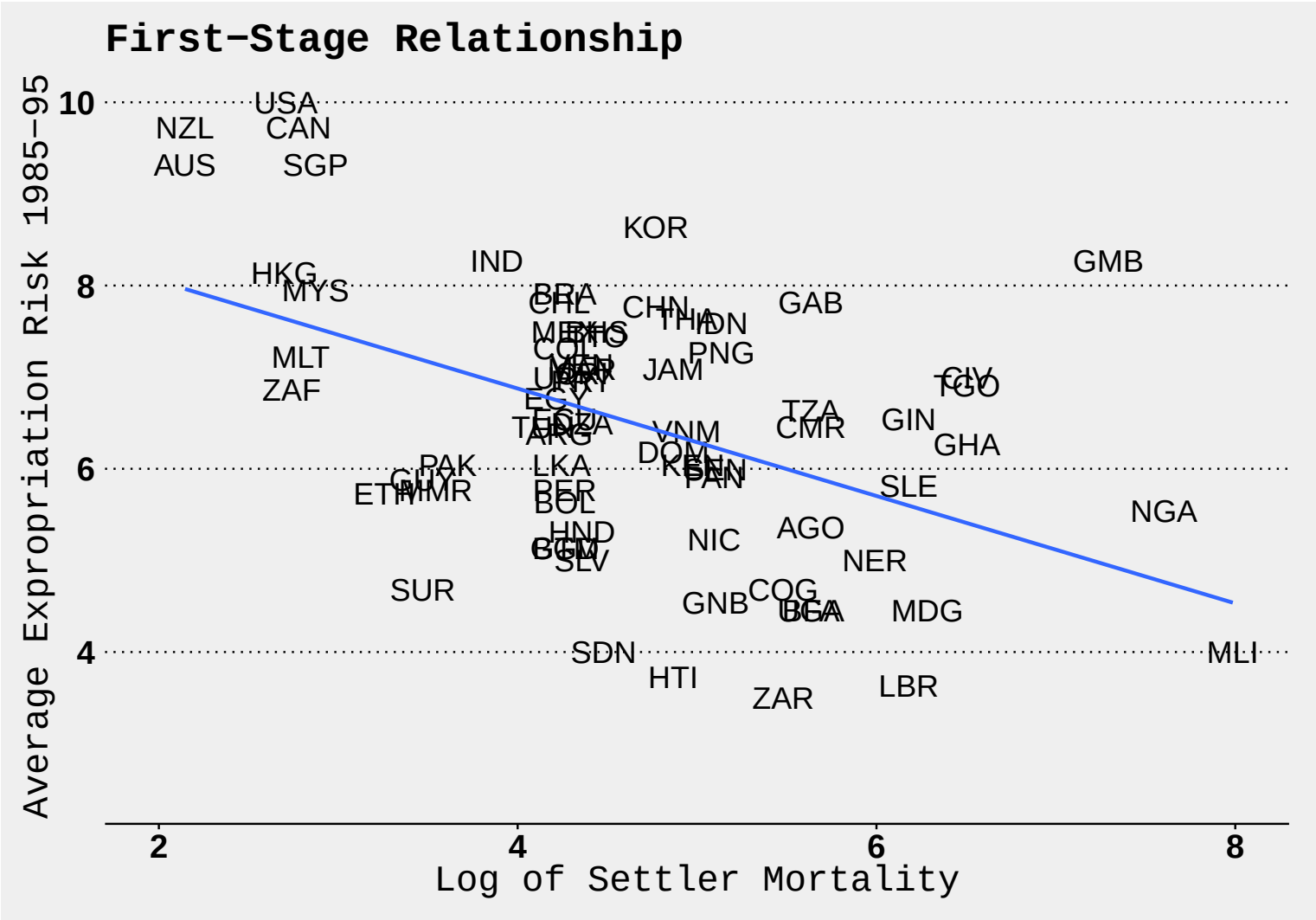
Themes: `theme_wsj()`

```
library(ggthemes)

ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_wsj(color='gray') +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Themes: `theme_wsj()`



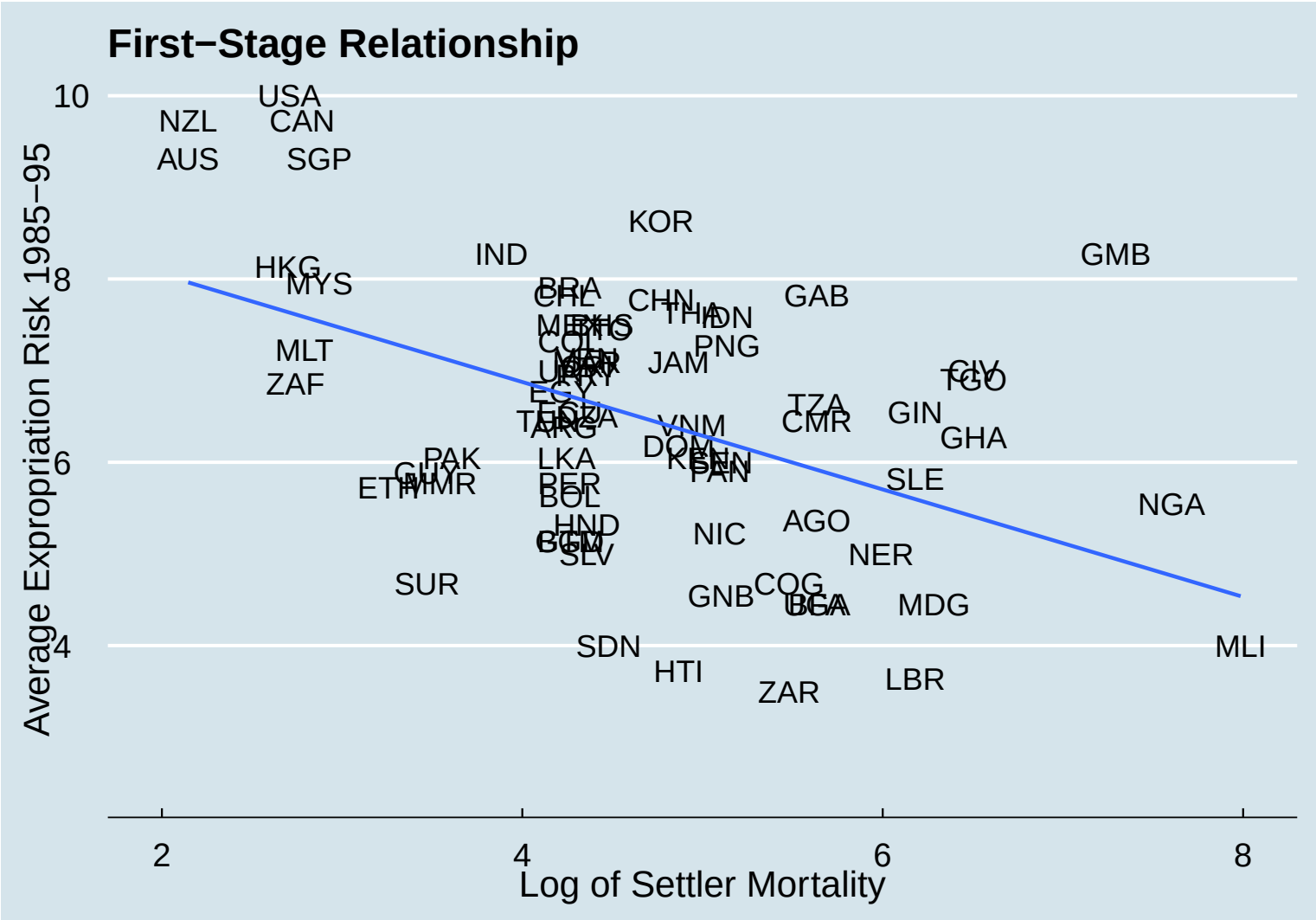
Scatterplot

Themes: `theme_economist()`

```
ggplot(data = df, aes(x = logem4, y = avexpr, label = shortnam)) +
  geom_text(size = 6) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_economist() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Themes: `theme_economist()`



Scatterplot

Annotating

```
p = ggplot(data = df, aes(x = logem4, y = avexpr)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  geom_smooth(method = lm, se = FALSE) +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

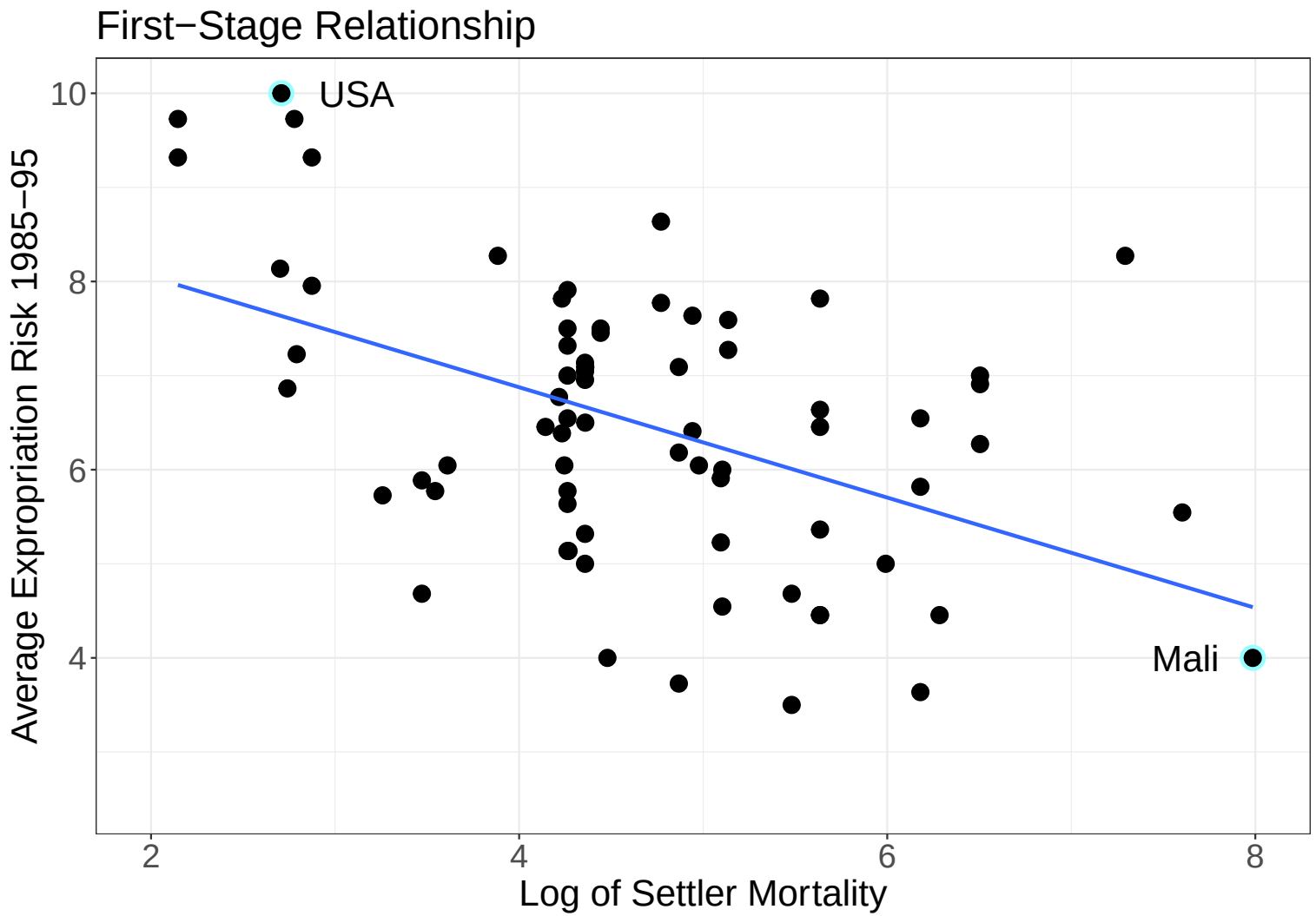
Annotating

```
usa = df %>% filter(shortnam == "USA") %>%
  select(avexpr, logem4) %>% unlist()
mali = df %>% filter(shortnam == "MLI") %>%
  select(avexpr, logem4) %>% unlist()

p +
  # USA
  annotate(geom = "point", x = usa["logem4"], y = usa["avexpr"],
           colour = "darkslategray1", size = 8) +
  annotate(geom = "point", x = usa["logem4"], y = usa["avexpr"],
           size = 4) +
  annotate(geom = "text", x = usa["logem4"], y = usa["avexpr"],
           label = "USA", size = 7, hjust = -0.5) +
  # Mali
  annotate(geom = "point", x = mali["logem4"], y = mali["avexpr"],
           colour = "darkslategray1", size = 8) +
  annotate(geom = "point", x = mali["logem4"], y = mali["avexpr"],
           size = 4) +
  annotate(geom = "text", x = mali["logem4"], y = mali["avexpr"],
           label = "Mali", size = 7, hjust = 1.5)
```

Scatterplot

Annotating



Scatterplot

Coloring by group

```
ggplot(data = df, aes(x = logem4, y = avexpr,
                      color = f_brit)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Scatterplot

Coloring by group

Scatterplot

Coloring by group

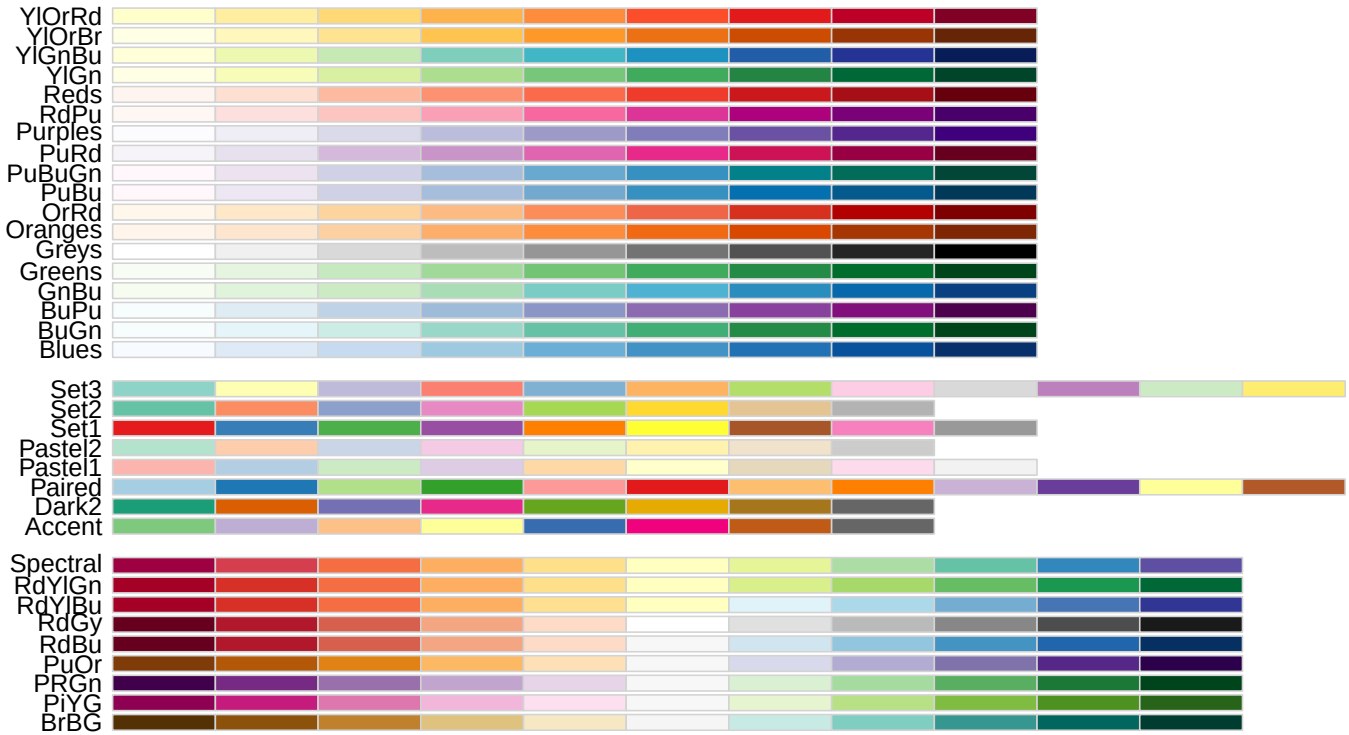
```
ggplot(data = df %>% filter(is.na(f_brit) == FALSE),
       aes(x = logem4, y = avexpr, color = f_brit)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  # MANUAL
  scale_color_manual(
    name = "British colony", # ADDS LEGEND TITLE
    values = c("0" = "red", "1" = "blue")) +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Scatterplot

Coloring by group

RColorBrewer

```
RColorBrewer::display.brewer.all()
```



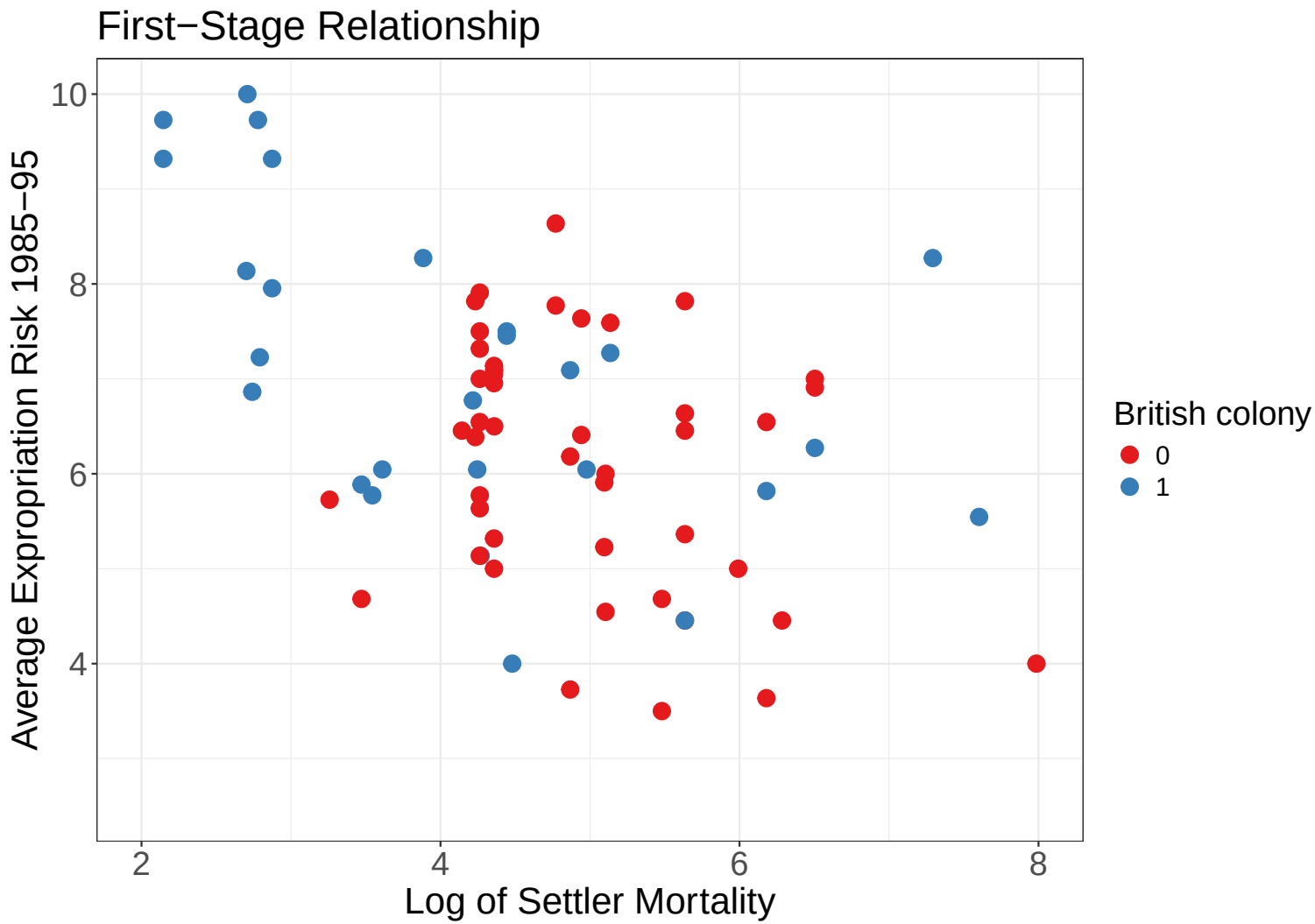
Scatterplot

Coloring by group

```
ggplot(data = df %>% filter(is.na(f_brit) == FALSE),
       aes(x = logem4, y = avexpr, color = f_brit)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  # BREWER
  scale_color_brewer(name = "British colony",
                     palette = "Set1") +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Scatterplot

Coloring by group



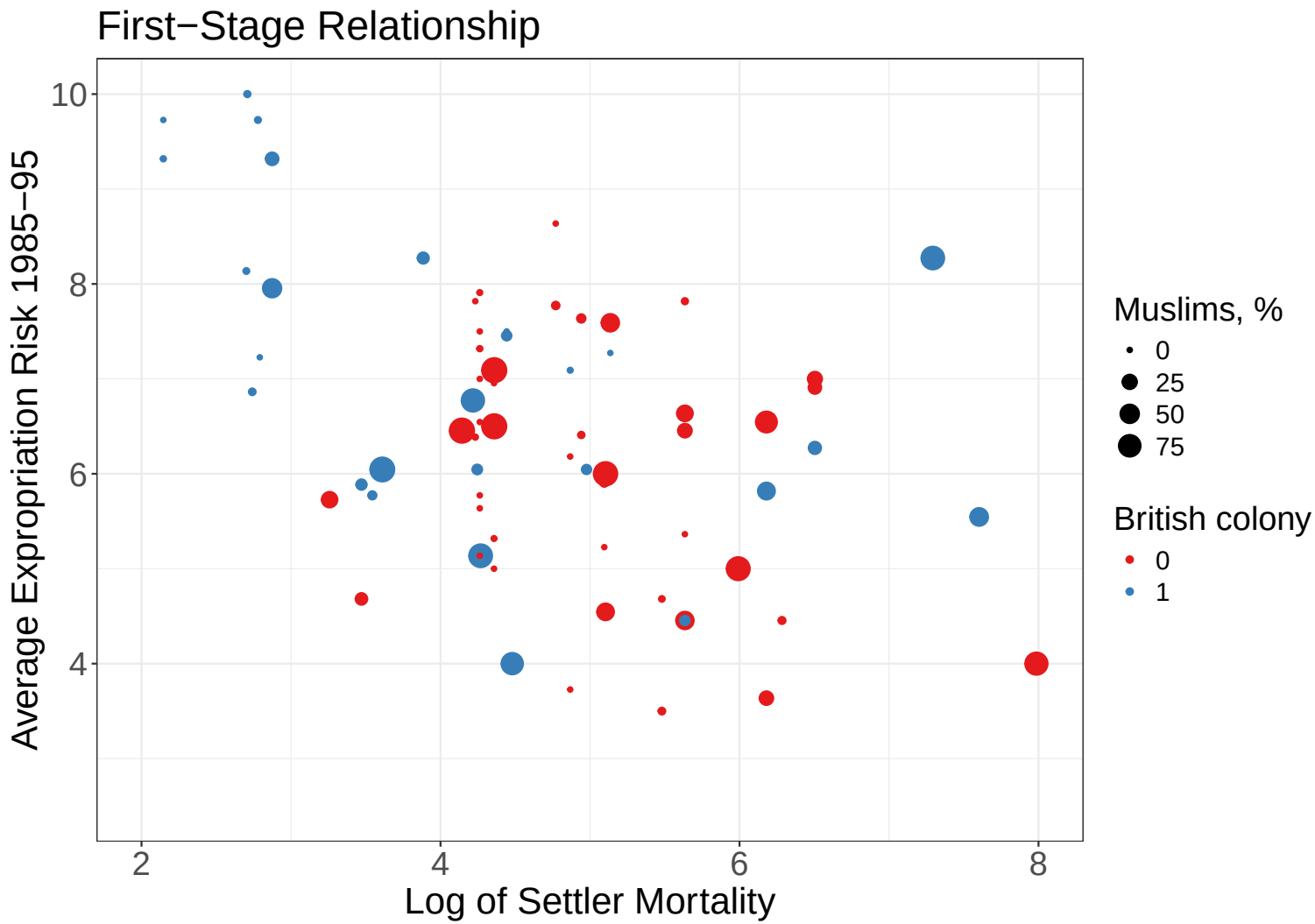
Scatterplot

Size by value

```
ggplot(data = df %>% filter(is.na(f_brit) == FALSE),
       aes(x = logem4, y = avexpr,
           color = f_brit, size = muslim80)) +
  geom_point() +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  scale_color_brewer(name = "British colony",
                     palette = "Set1") +
  scale_size_continuous(name = "Muslims, %") +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Scatterplot

Size by value



Scatterplot

Faceting

- Faceting generates and displays subsets of data defined by a categorical variable (or combination of multiple categorical variables)
- To create a faceted plot, you need to add a new layer with `facet_wrap()`
  - Specify a name of a grouping variable after the~ sign
- Faceted plots are displayed in a table
  - You can control the number of rows and columns using the `nrow` and `ncol` arguments
  - `dir` controls the direction of wrap: `"h"` for horizontal or `"v"` for vertical

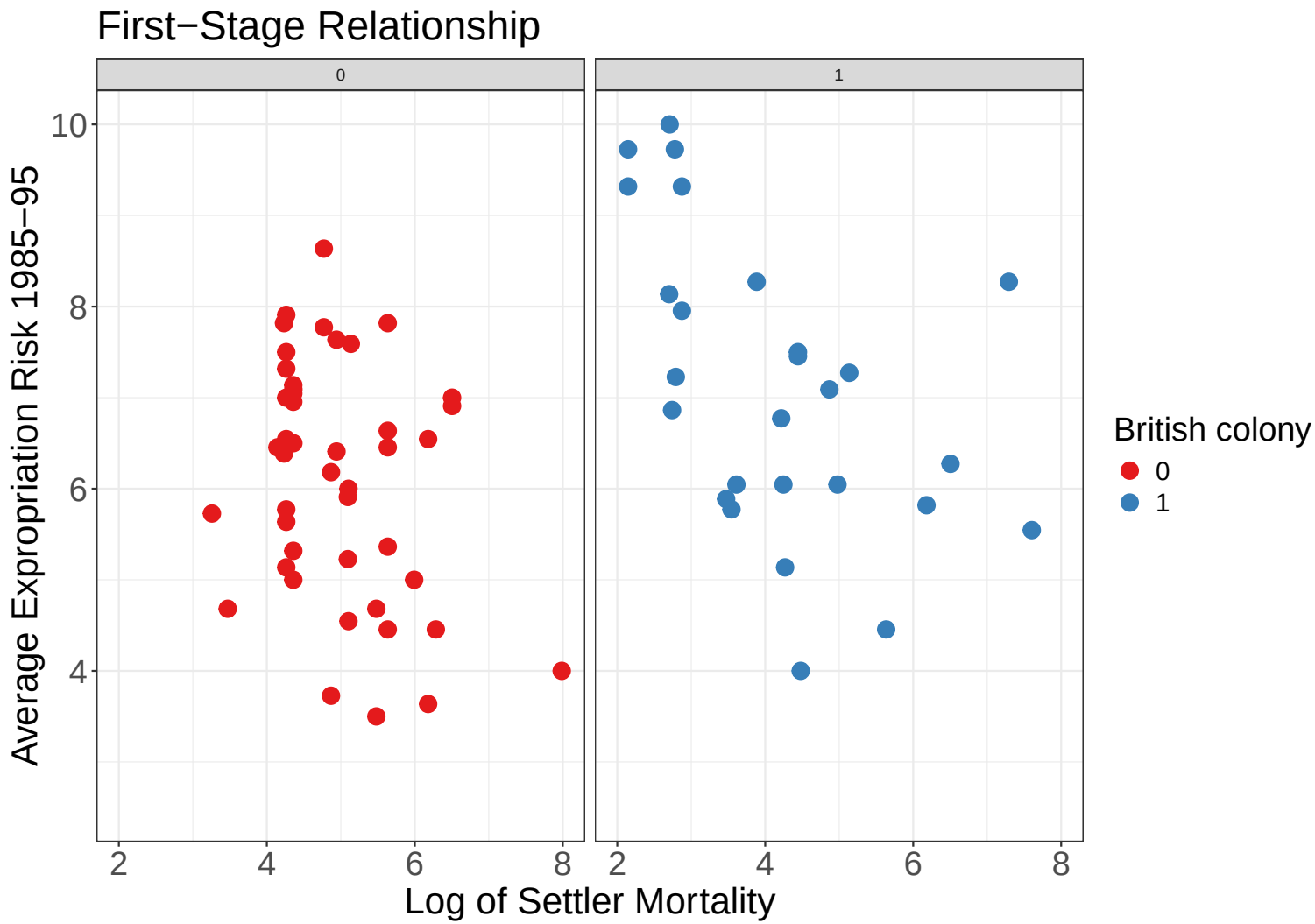
Scatterplot

Faceting

```
ggplot(data = df %>% filter(is.na(f_brit) == FALSE),
       aes(x = logem4, y = avexpr,
           color = f_brit)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  scale_color_brewer(name = "British colony",
                     palette = "Set1") +
  scale_size_continuous(name = "Muslims, %") +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14)) +
  facet_wrap(~factor(f_brit))
```

Scatterplot

Faceting



Scatterplot

Positioning a legend

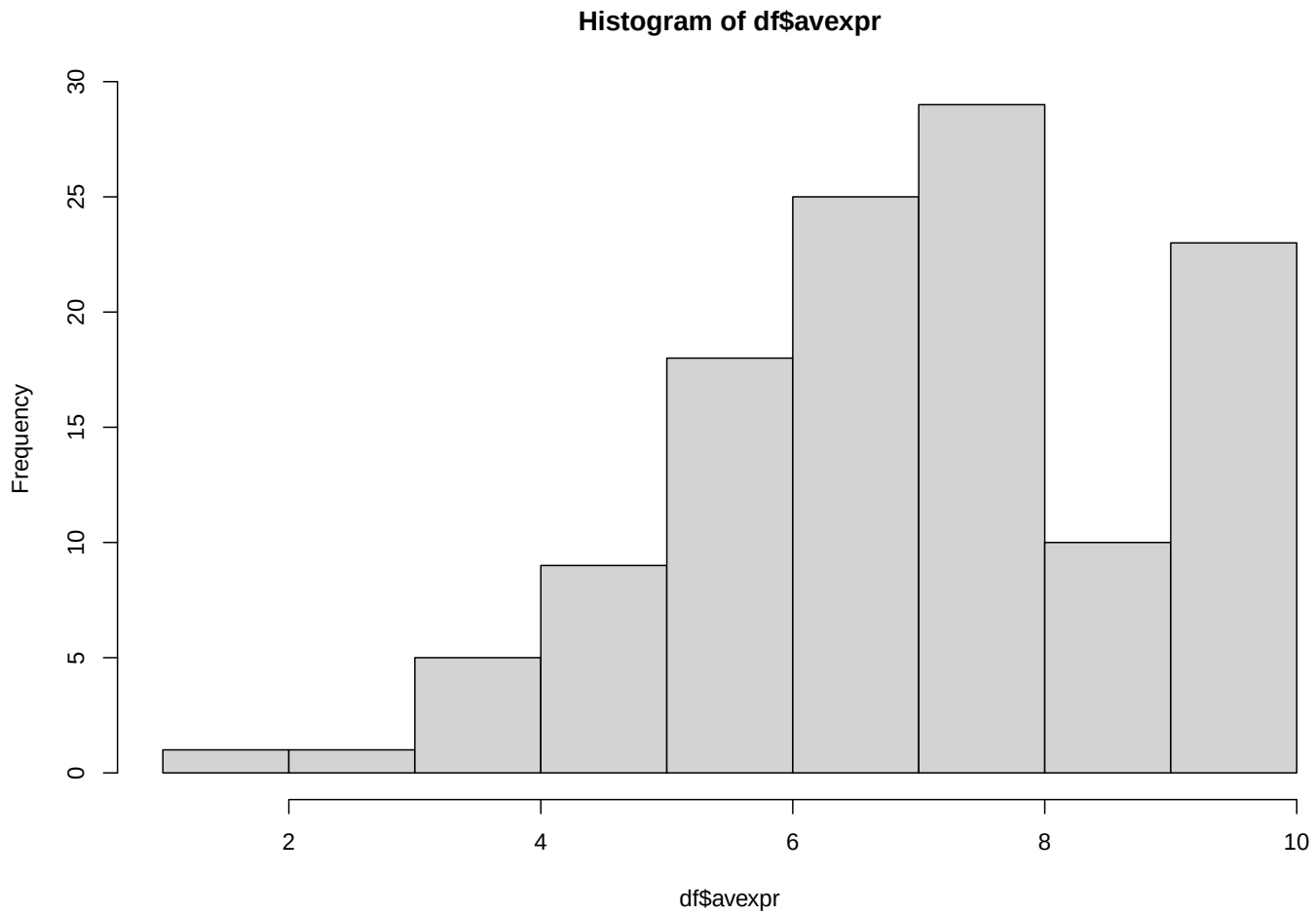
```
ggplot(data = df %>% filter(is.na(f_brit) == FALSE),
       aes(x = logem4, y = avexpr, color = f_brit)) +
  geom_point(size = 4) +
  labs(x = "Log of Settler Mortality",
       y = "Average Expropriation Risk 1985-95",
       title = "First-Stage Relationship") +
  scale_x_continuous(limits = c(2, 8)) +
  scale_y_continuous(limits = c(2.5, 10),
                     breaks = c(4, 6, 8, 10)) +
  scale_color_brewer(name = "British colony",
                     palette = "Set1") +
  scale_size_continuous(name = "Muslims, %") +
  theme_bw() +
  theme(plot.title = element_text(size = 22),
        axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14)) +
  facet_wrap(~factor(f_brit)) +
  theme(legend.position = "bottom")
```

Scatterplot

Positioning a legend

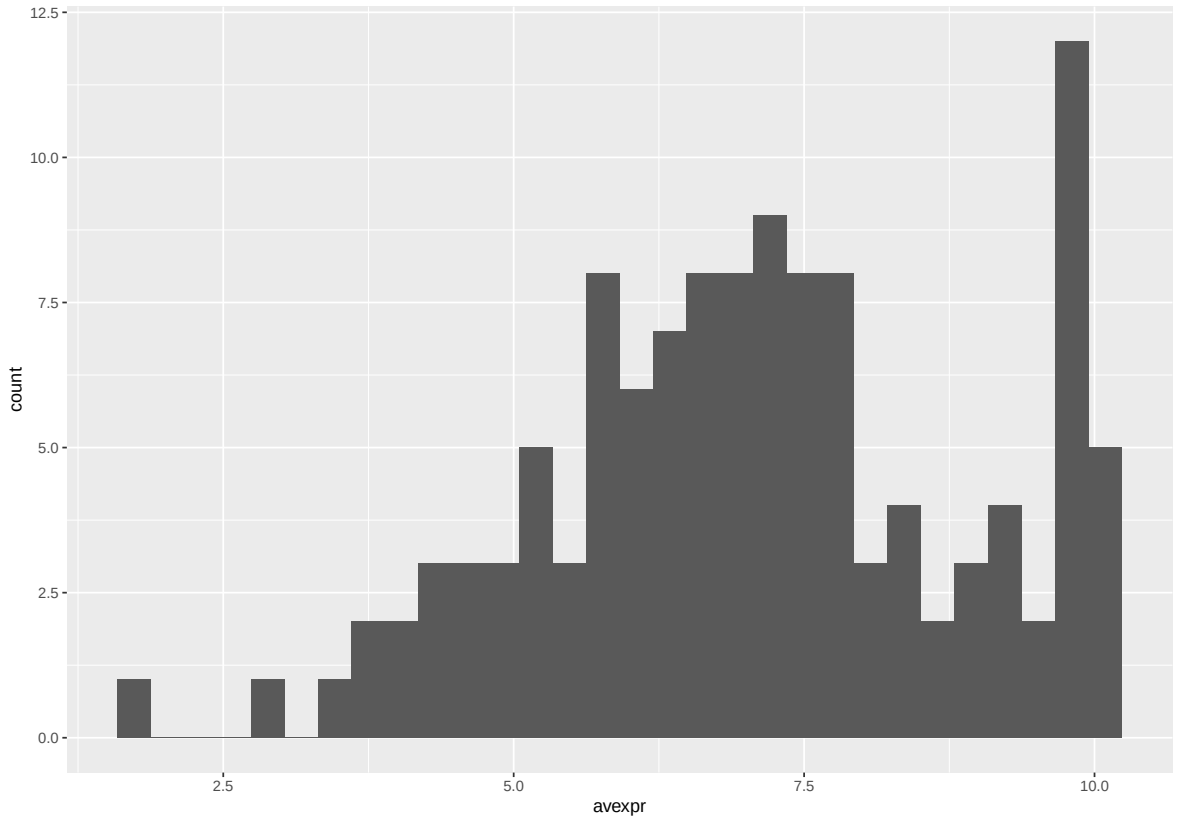
Histogram in base R

```
hist(df$avexpr)
```



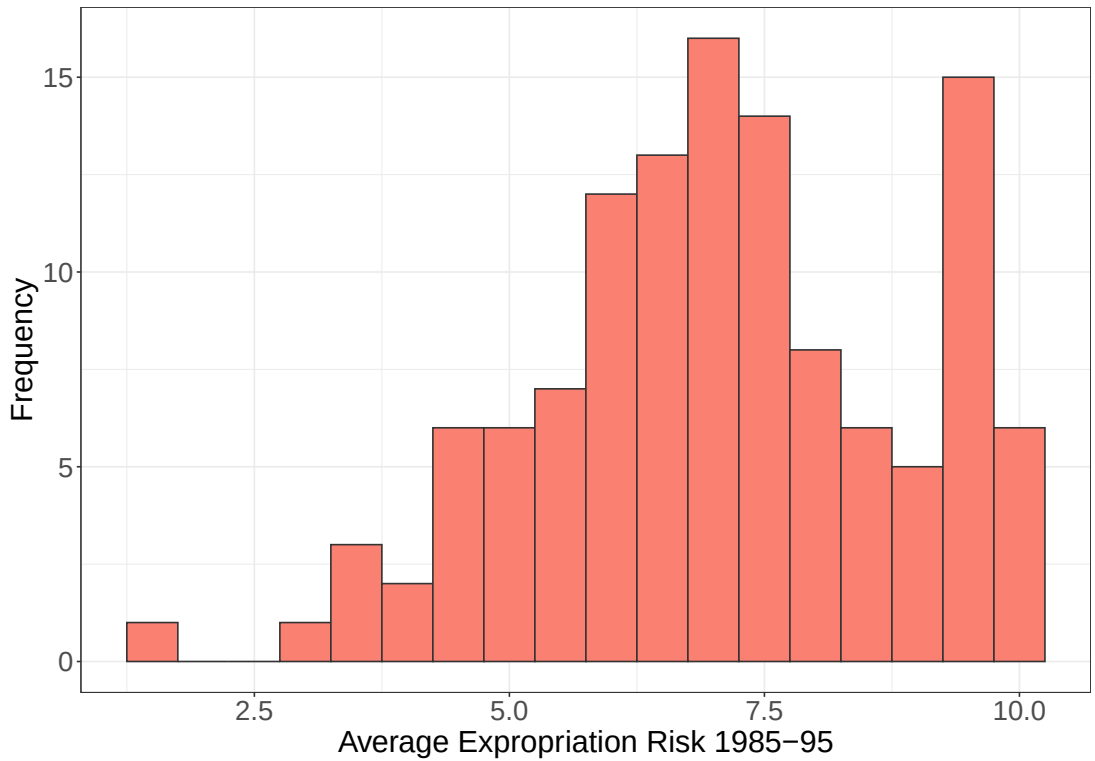
Histogram

```
ggplot(data = df, aes(x = avexpr)) +
  geom_histogram()
```



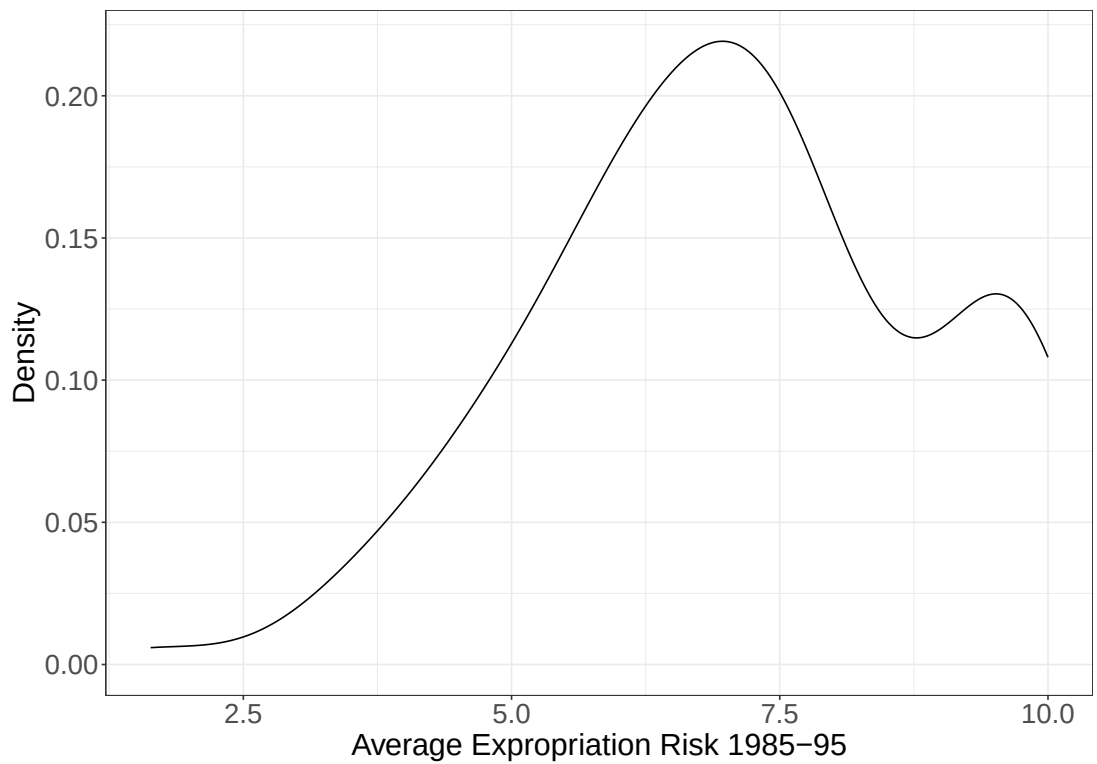
Histogram

```
ggplot(data = df, aes(x = avexpr)) +
  geom_histogram(binwidth = 0.5, fill = "salmon", color = "gray20") +
  labs(x = "Average Expropriation Risk 1985-95", y = "Frequency") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```



Density plot

```
ggplot(data = df, aes(x = avexpr)) +
  geom_density() +
  labs(x = "Average Expropriation Risk 1985-95", y = "Density") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```



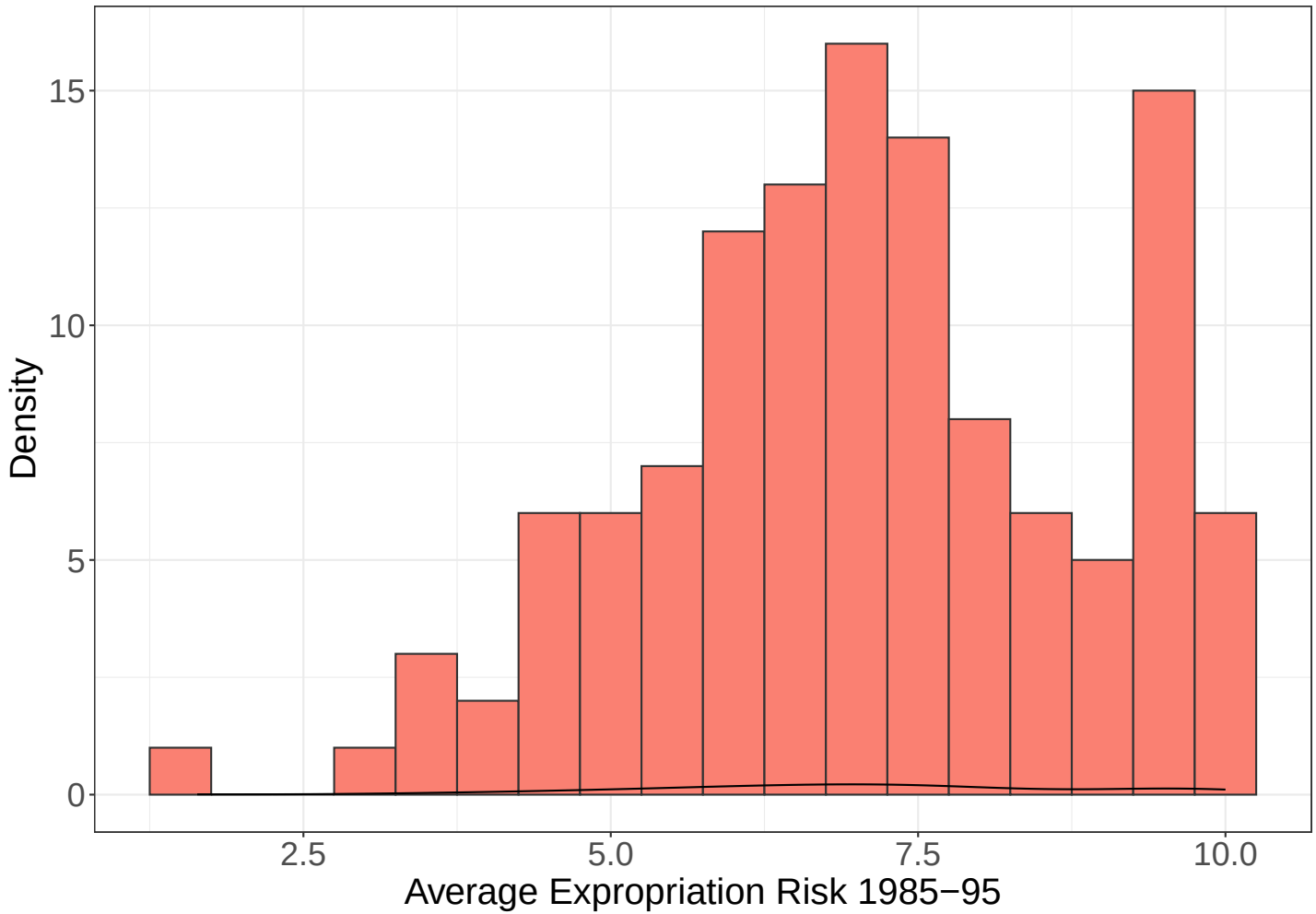
Overlaying histogram and density

We have a problem

```
ggplot(data = df, aes(x = avexpr)) +
  geom_histogram(binwidth = 0.5, fill = "salmon", color = "gray20") +
  geom_density() +
  labs(x = "Average Expropriation Risk 1985-95", y = "Density") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Overlaying histogram and density

We have a problem



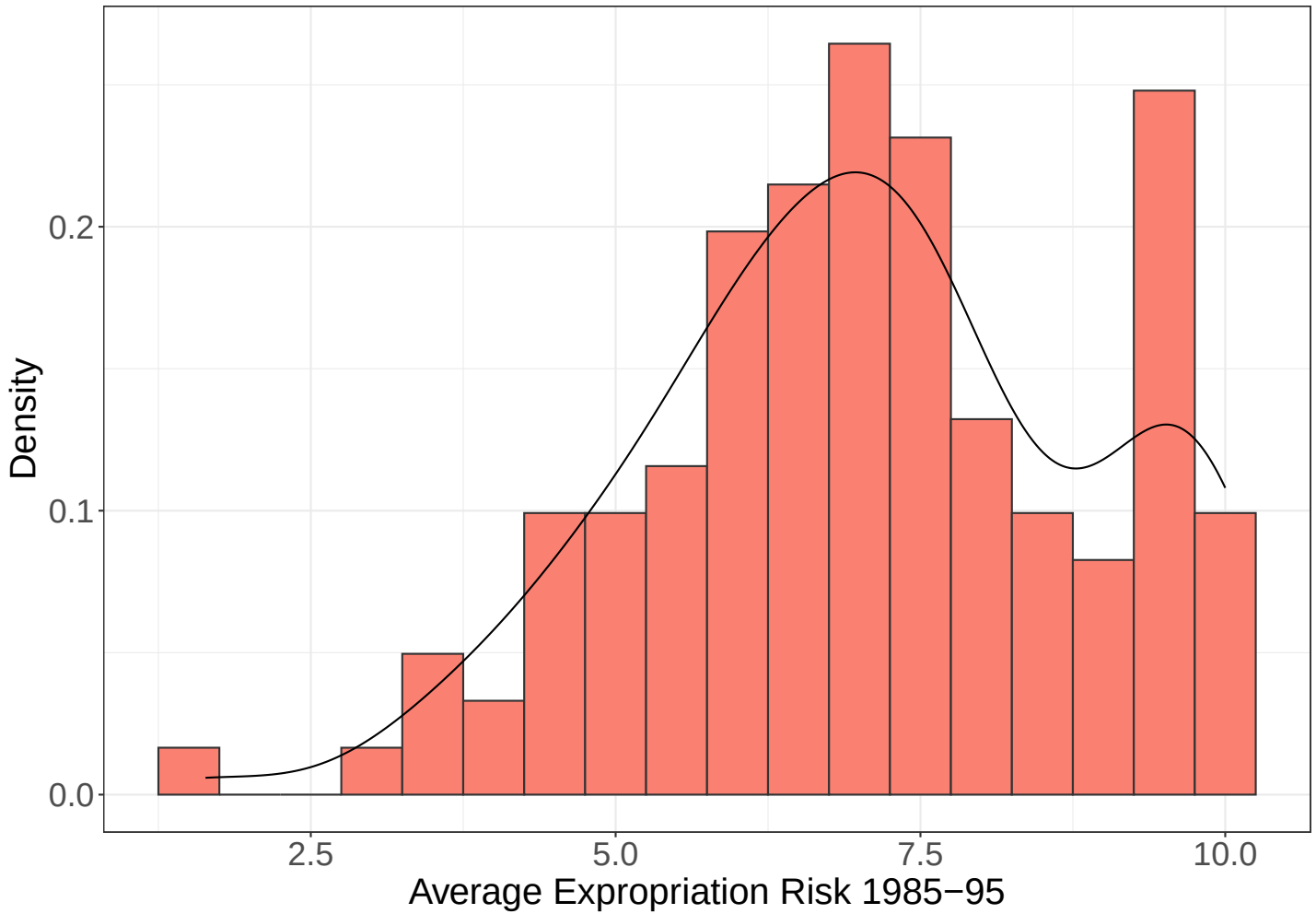
Overlaying histogram and density

Updating y-axis

```
ggplot(data = df, aes(x = avexpr, y = after_stat(density))) +
  geom_histogram(binwidth = 0.5, fill = "salmon", color = "gray20") +
  geom_density() +
  labs(x = "Average Expropriation Risk 1985-95", y = "Density") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18))
```

Overlaying histogram and density

Updating y-axis



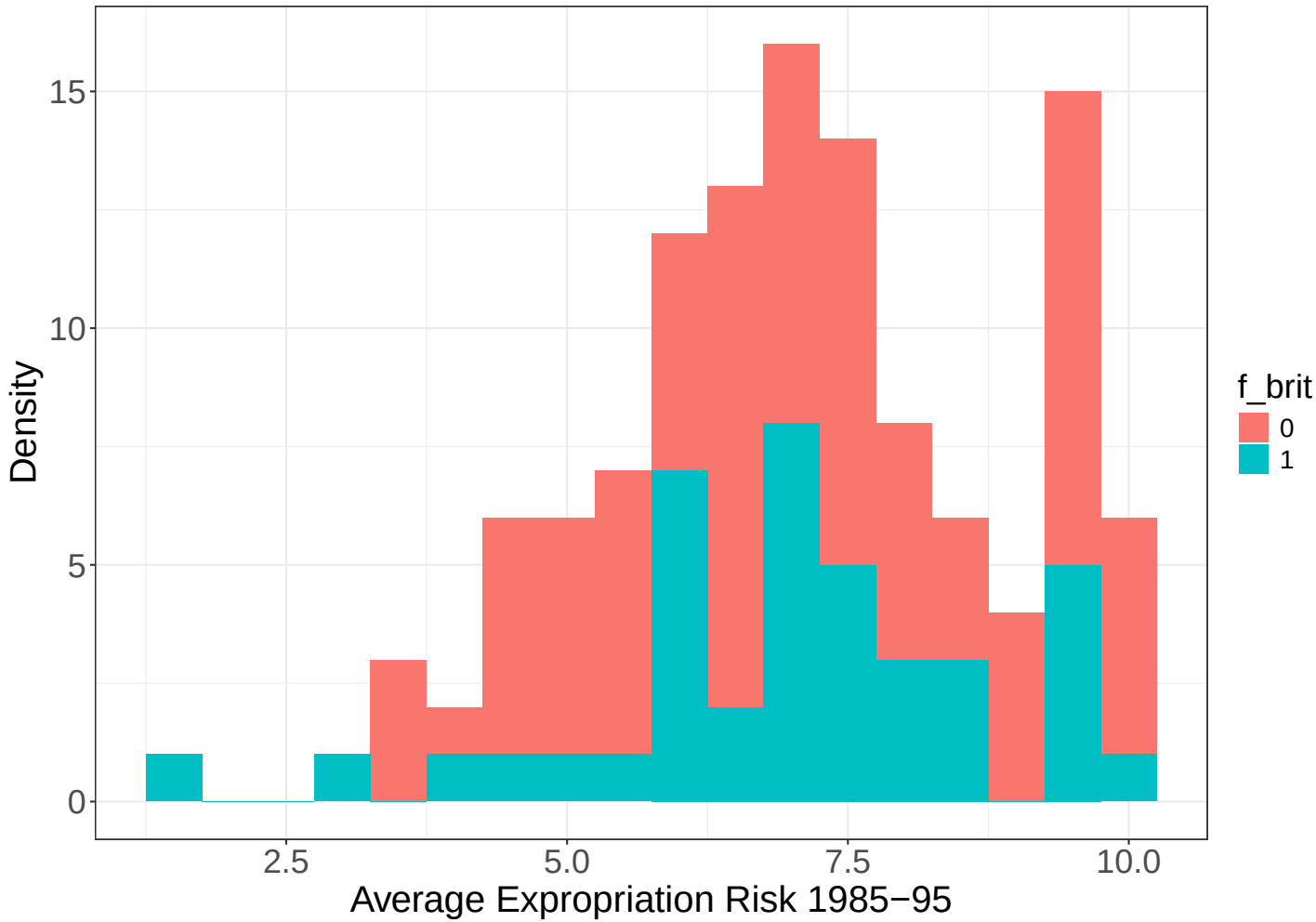
Histogram

Color by group

```
ggplot(data = df %>% filter(is.na(f_brit) == F),
       aes(x = avexpr, fill = f_brit)) +
  geom_histogram(binwidth = 0.5) +
  labs(x = "Average Expropriation Risk 1985-95", y = "Density") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Histogram

Color by group



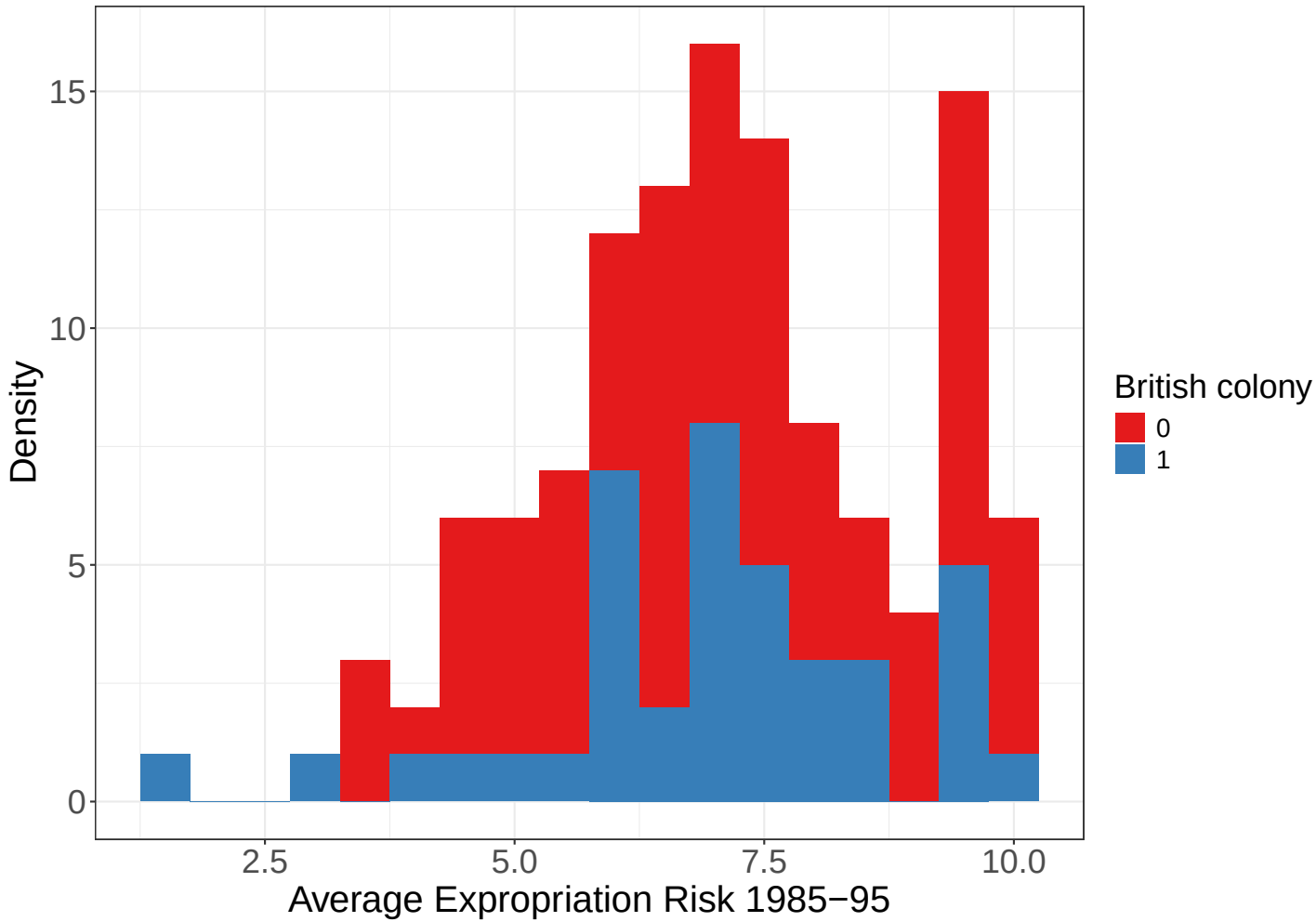
Histogram

Color by group

```
ggplot(data = df %>% filter(is.na(f_brit) == F),
       aes(x = avexpr, fill = f_brit)) +
  geom_histogram(binwidth = 0.5) +
  scale_fill_brewer(name = "British colony",
                    palette = "Set1") +
  labs(x = "Average Expropriation Risk 1985-95", y = "Density") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Histogram

Color by group



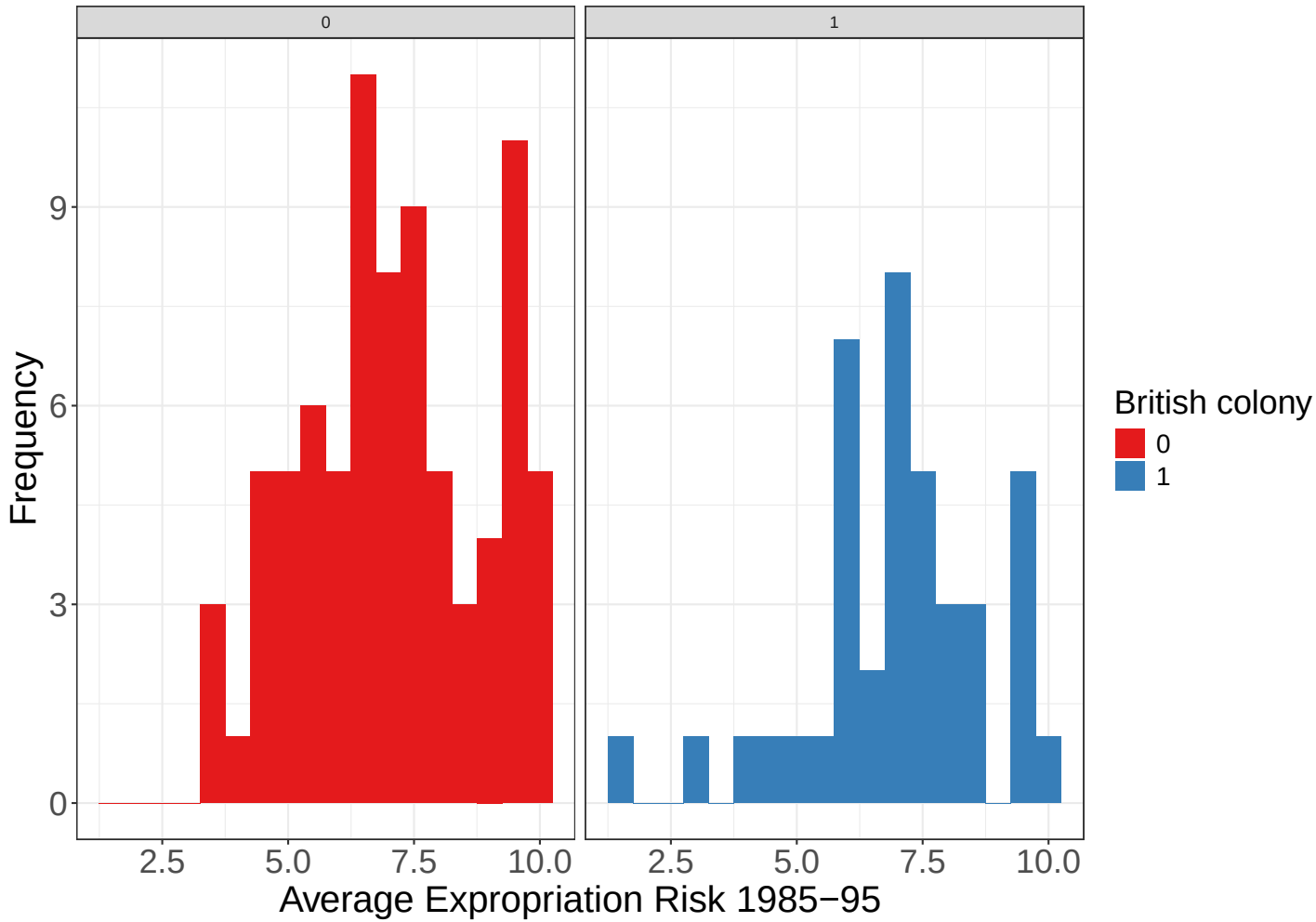
Histogram

Faceting

```
ggplot(data = df %>% filter(is.na(f_brit) == F),
       aes(x = avexpr, fill = f_brit)) +
  geom_histogram(binwidth = 0.5) +
  scale_fill_brewer(name = "British colony",
                    palette = "Set1") +
  labs(x = "Average Expropriation Risk 1985-95", y = "Frequency") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14)) +
  facet_wrap(~f_brit)
```

Histogram

Faceting



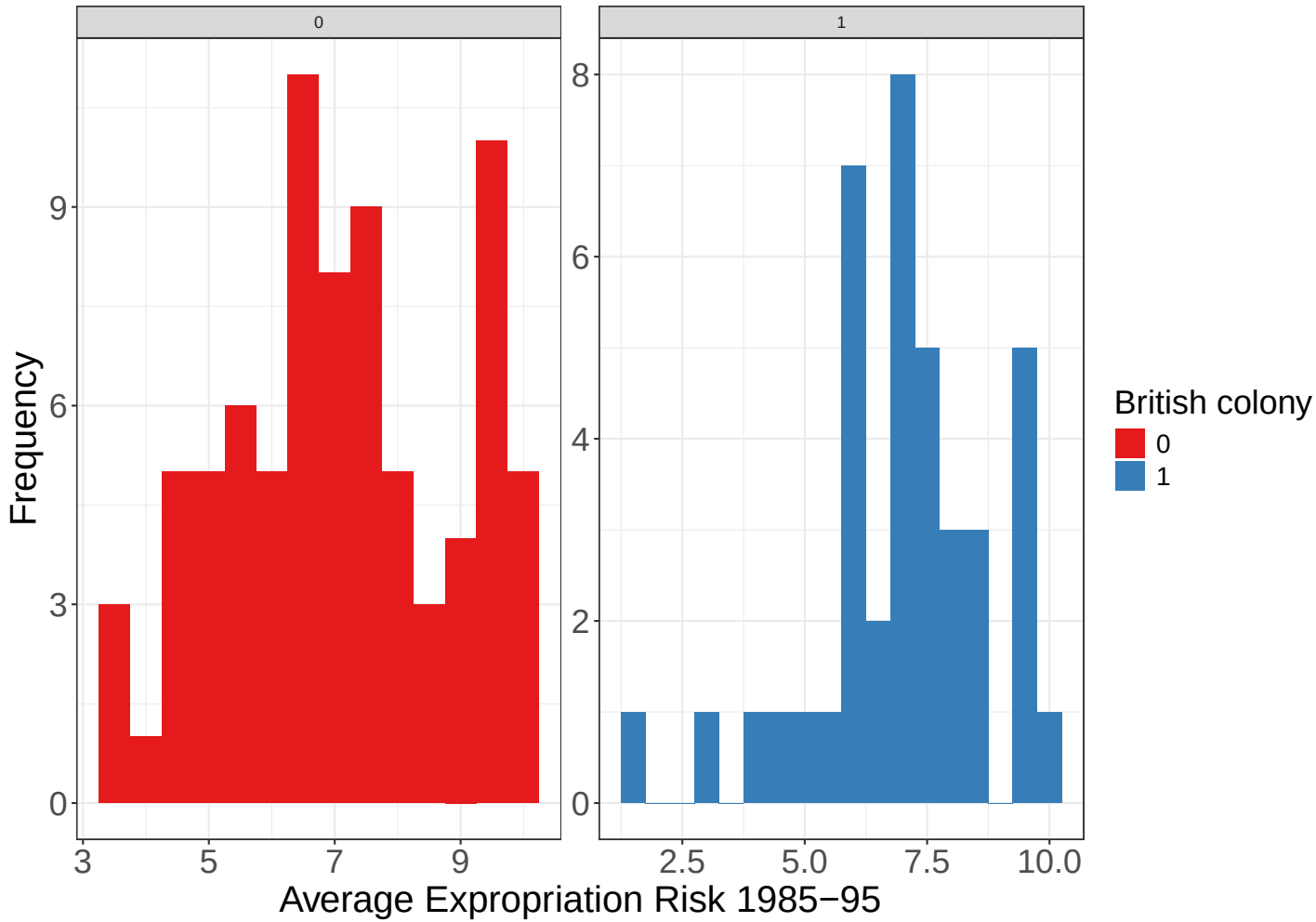
Histogram

Controlling scales

```
ggplot(data = df %>% filter(is.na(f_brit) == F),
       aes(x = avexpr, fill = factor(f_brit))) +
  geom_histogram(binwidth = 0.5) +
  scale_fill_brewer(name = "British colony",
                    palette = "Set1") +
  labs(x = "Average Expropriation Risk 1985-95", y = "Frequency") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14)) +
  facet_wrap(~f_brit, scales = "free")
```

Histogram

Controlling scales



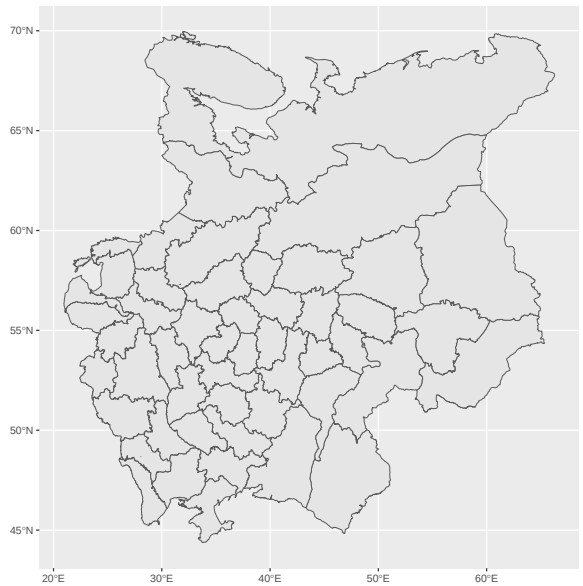
Intro to maps

- A lot of social processes have a spatial dimension
  - *Everything is related to everything else, but near things are more related than distant things*
  - Visualizing data in space might be really illuminating
- To plot a map, you need a file that contains the coordinates of objects you want to display (it is usually called a shapefile)
- Use `st_read()` from the `sf` package to load a shapefile in R
- In `ggplot2`, use `geom_sf()` to add a map layer

Intro to maps

```
library(sf)
re = st_read("re/re.shp")
```

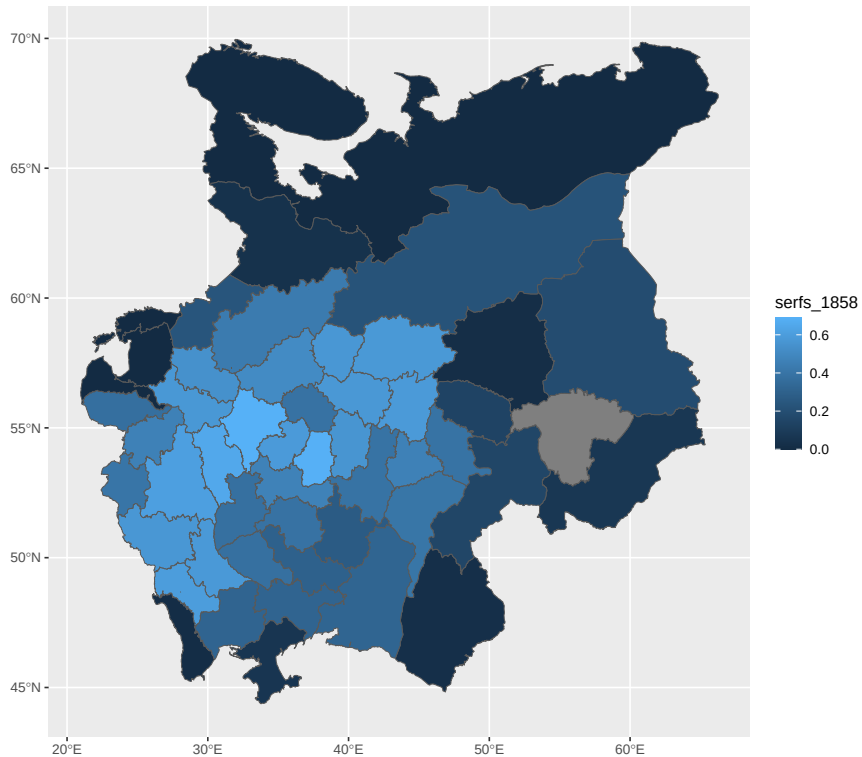
```
ggplot(data = re) +
  geom_sf()
```



Intro to maps

Coloring a map

```
ggplot(data = re, aes(fill = serfs_1858)) +
  geom_sf()
```



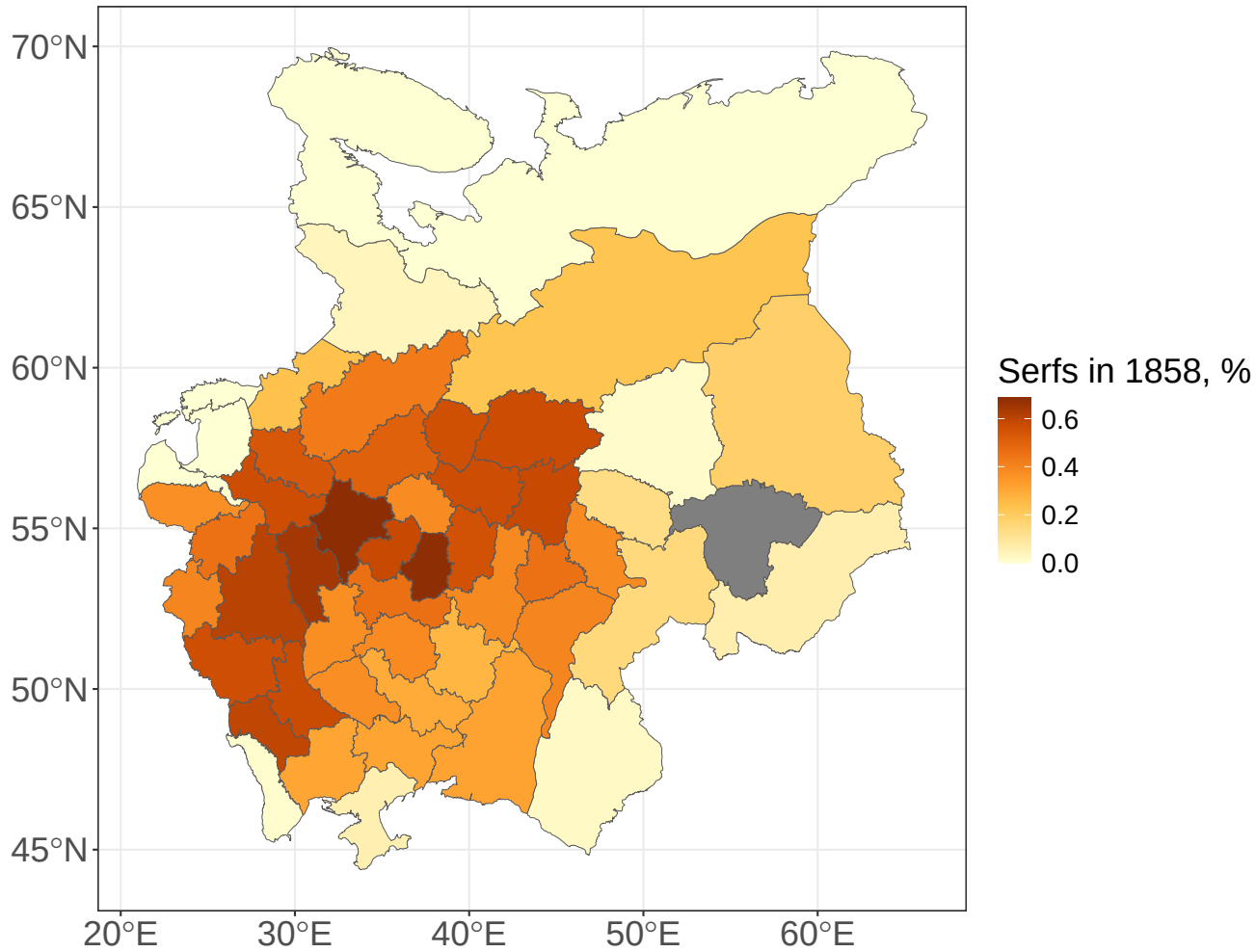
Intro to maps

Customizing a map

```
ggplot(data = re, aes(fill = serfs_1858)) +
  geom_sf() +
  scale_fill_distiller(palette = "YlOrBr",
                       direction = 1,
                       name = "Serfs in 1858, %") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 18),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 14))
```

Intro to maps

Customizing a map



Exporting a plot

```
ggplot(data = re, aes(fill = serfs_1858)) +
  geom_sf()
ggsave("my-first-map.pdf")
```

Activity

1. Plot a histogram of settler mortality rates (`logem4`). Set `binwidth` equal to 1. Label axes and customize colors. Examine the histograms of settler mortality rates across British and non-British colonies.
   - Here you can access the list of colors available in R
2. Plot a scatterplot of settler mortality rates and the logarithm of GDP per capita in 1995 (`logpgp95`). Label points and add a regression line.
3. Build a boxplot of settler mortality rates setting `aes(y = logem4)` and adding `geom_boxplot()`
   - Suppose you want to zoom in on countries for which log mortality rates lie between 3 and 7 Set the limits of the y scale. What do you notice?
   - Now use `coord_cartesian(ylim = c(3, 7))` instead of setting the scale. What changes now?
   - Throughout the problem, you can also plot the median adding a layer with a horizontal line using `geom_hline(yintercept = median(df$logem4, na.rm = T))` (don't forget to change a color)

Further reads & useful resources

- Textbooks
  - Winston Chang, R Graphics Cookbook
  - Hadley Wickham, ggplot2: Elegant Graphics for Data Analysis
- Colors
  - Colors in R
  - Color palettes generator
- Maps
  - David Rumsey Map Center workshops
  - Look up shapefiles at Stanford EarthWorks