# PBA: 2.
# Data Visualization

Fall 2026

Jaewon (“Jay-one”) Yoo
National Tsing Hua University

# Where are we? Where are we going?

- **Where we have been**
  - Week 1: R and RStudio, functions and packages, Quarto
  - Assigned readings for Weeks 1–2: the R basics (operators, types, vectors, data frames); we start using them today

# Where are we? Where are we going?

- **Where we have been**
  - Week 1: R and RStudio, functions and packages, Quarto
  - Assigned readings for Weeks 1–2: the R basics (operators, types, vectors, data frames); we start using them today
- **Today: a first look at data visualization**
  - This is how R is learned here: on data, from the first week. The grammar comes when you need it; nothing today assumes prior programming
  - Slides and the R script are on the course website every week; after the lecture you run today's code yourself, then E1 at your own pace with the TAs

# Where are we? Where are we going?

- **Where we have been**
  - Week 1: R and RStudio, functions and packages, Quarto
  - Assigned readings for Weeks 1–2: the R basics (operators, types, vectors, data frames); we start using them today
- **Today: a first look at data visualization**
  - This is how R is learned here: on data, from the first week. The grammar comes when you need it; nothing today assumes prior programming
  - Slides and the R script are on the course website every week; after the lecture you run today's code yourself, then E1 at your own pace with the TAs
- **Where we are going**
  - Next week: data wrangling. Then causality, prediction, regression, and uncertainty

**Visualization** → Wrangling → Causality → Prediction → Regression → Uncertainty

# Roadmap

1. **A first look at data, and a first plot**
    - Midwest counties: one row is a county

2. **Anatomy of a ggplot**
    - Building plots by layers
    - Geoms (i.e., geometric objects) are functions!
    - Adding informative labels
    - Mapping versus setting aesthetics

3. **Histograms/boxplots and grouped data**
    - Histogram and boxplots
    - Grouped data

Rest of the class: In-class exercise E1 (Gapminder)

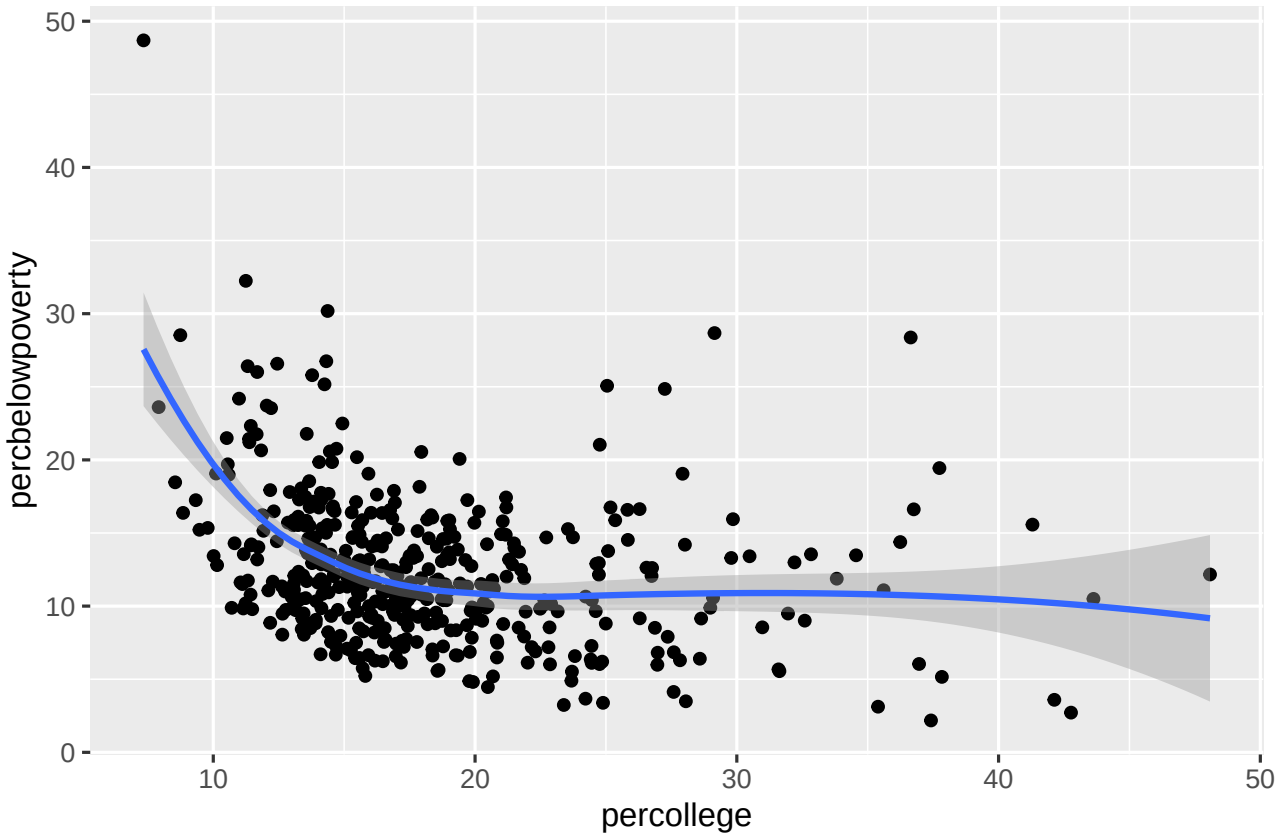
# Midwest Data

```
# Install once: install.packages(c("pacman", "ggplot2"))
# Load with library() or require(), or let pacman do both:
> pacman::p_load(ggplot2)
> midwest
# A tibble: 437 × 28
     PID county  state  area poptotal popdensity popwhite popblack popamerindian popasian popother
   <int> <chr>   <chr> <dbl>    <int>      <dbl>    <int>    <int>         <int>    <int>    <int>
 1   561 ADAMS   IL    0.052    66090      1271.    63917     1702            98      249      124
 2   562 ALEXAN  IL    0.014    10626       759      7054     3496            19       48        9
 3   563 BOND    IL    0.022    14991       681.    14477      429            35       16       34
 4   564 BOONE   IL    0.017    30806      1812.    29344      127            46      150     1139
 5   565 BROWN   IL    0.018     5836       324.     5264      547            14        5        6
#   percother <dbl>, popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,
#   poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,
#   percchildbelowpovert <dbl>, percadultpoverty <dbl>, percelderlypoverty <dbl>, inmetro <int>,
#   category <chr>
```

- The table comes with `ggplot2`; nothing to type in.
- One row is a **county**: 437 counties in five Midwest states (IL, IN, MI, OH, WI).
- Two columns for the first plots: `percollege` (% with a college degree), `percbelowpoverty` (% below the poverty line).
- `<int>`: integer; `<chr>`: text; `<dbl>`: numeric.

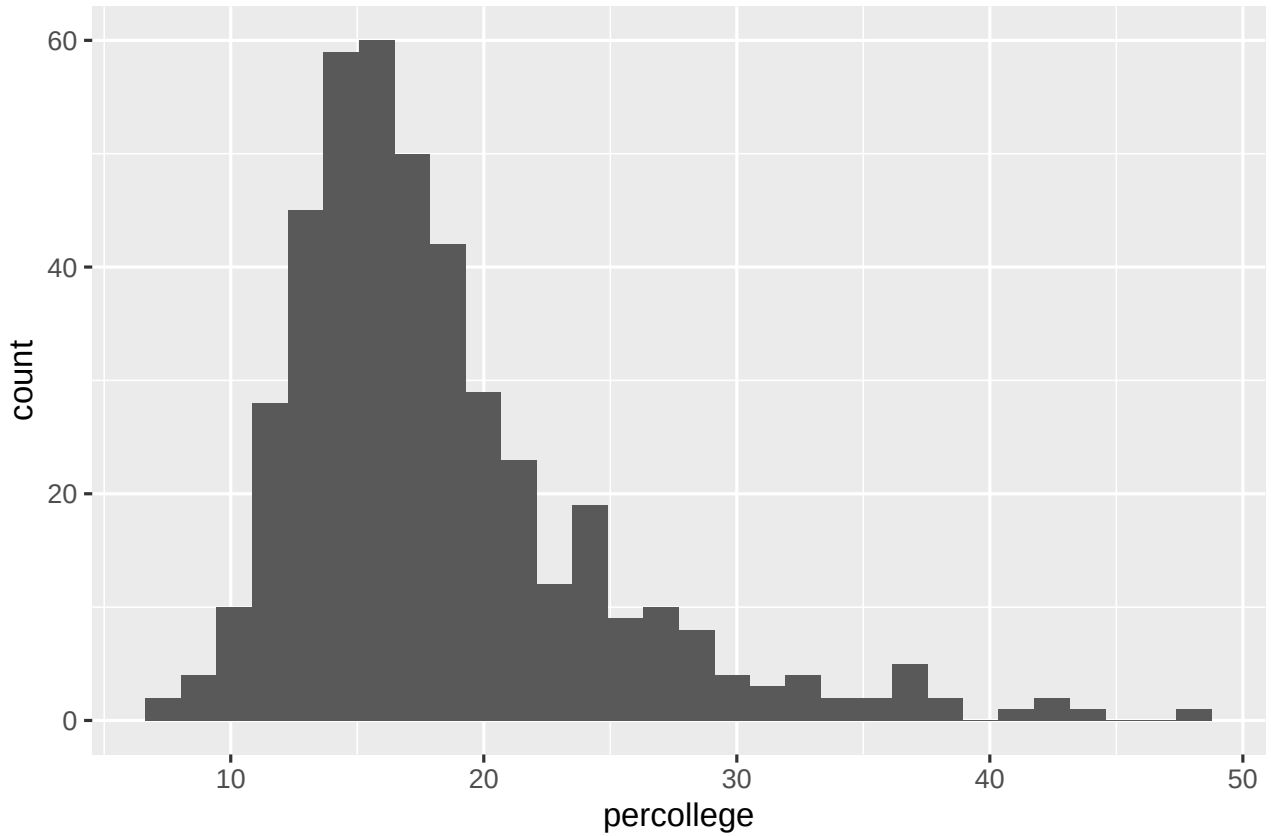
# College Education and Poverty

```
> ggplot(midwest, mapping = aes(x = percollege, y = percbelowpoverty)) +
    geom_point() + geom_smooth()   # the curve: loess, local smoothing
```



# A Histogram of College Education

```
> ggplot(midwest, mapping = aes(x = percollege)) +
    geom_histogram()
```



**Two lines of code drew that.**

**What did each piece do?**

The rest of today: the anatomy of a ggplot, the pieces you will use every week, then time to build your own.

# 1/ Anatomy of a ggplot

# Building Up a Graph in Pieces

- Create a ggplot object and direct it to the correct data:

```
> p <- ggplot(data = midwest)
```

- **Mapping:** tell ggplot what visual aesthetics correspond to which variables:

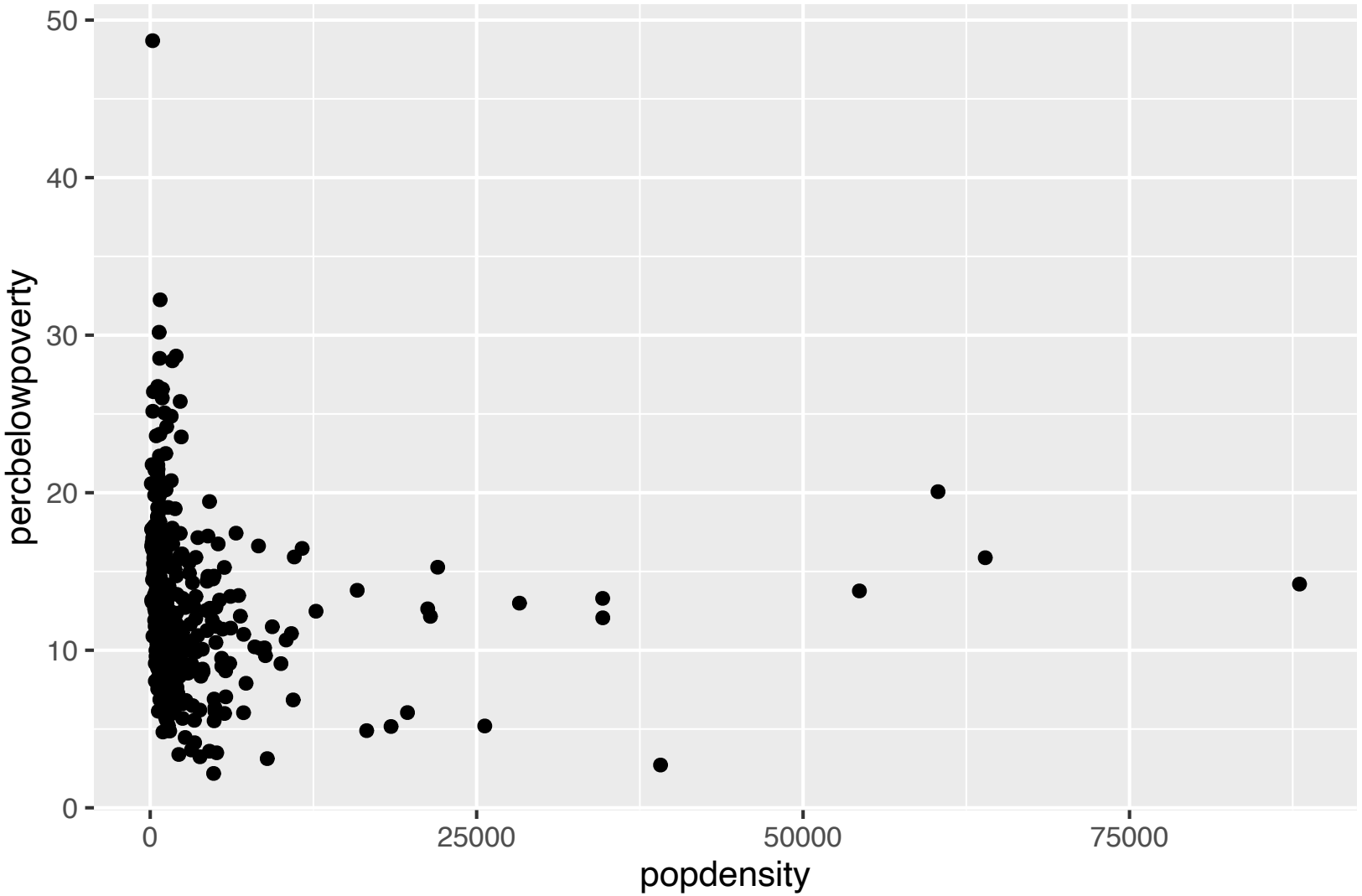
```
> p <- ggplot(data = midwest,
                mapping = aes(x = popdensity,
                              y = percbelowpoverty))
```

  - Other aesthetic mapping parameters: `color`, shape, `size`, etc.

# Adding a geom Layer

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point()
```

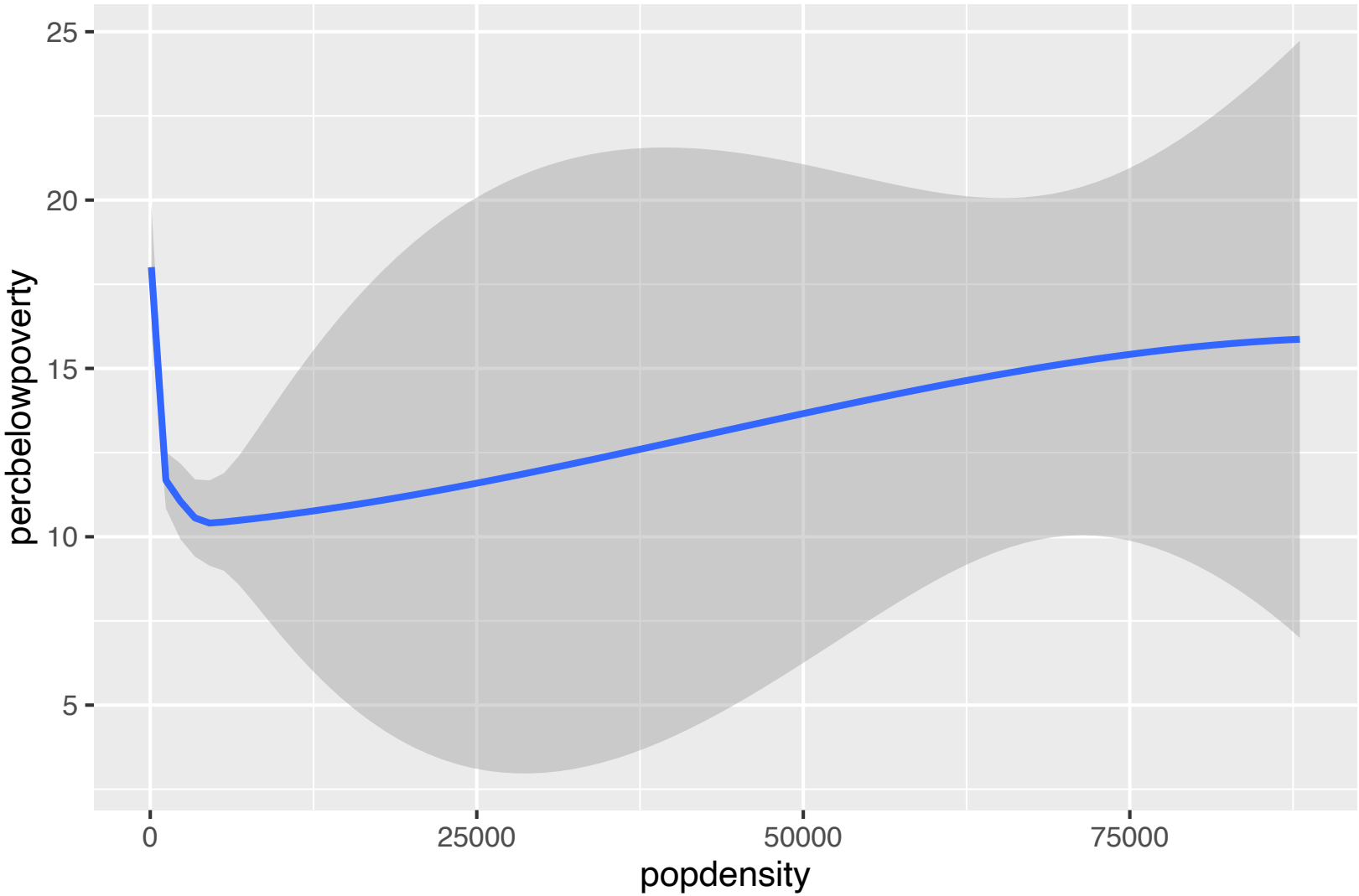
# Adding a geom Layer



# Trying a New geom

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_smooth()
```

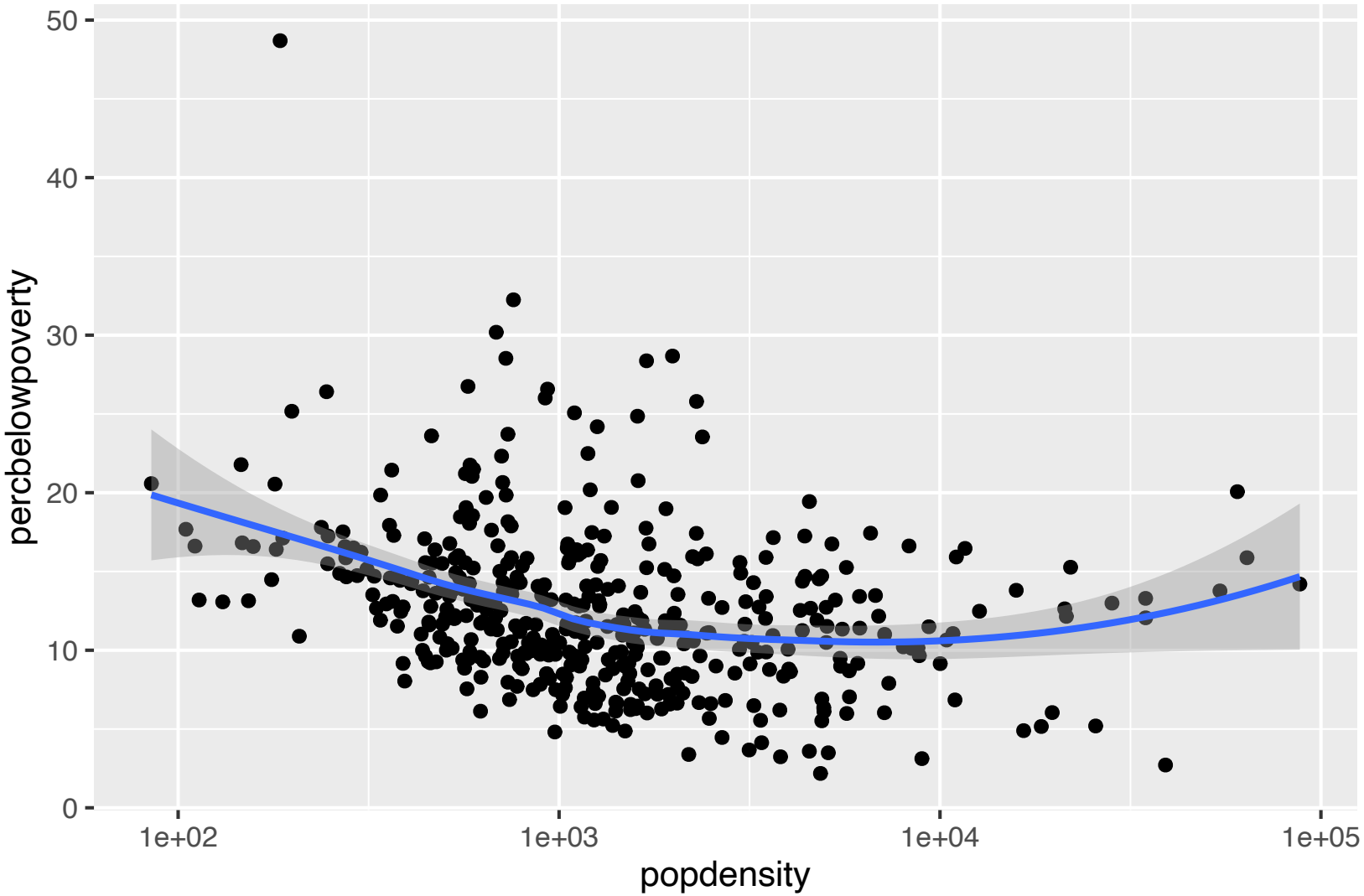
# Trying a New geom



# Layering geom is Additive!

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point() +
    geom_smooth() +
    scale_x_log10() # Log scale for the skewed x variable
```

# Layering geom is Additive!

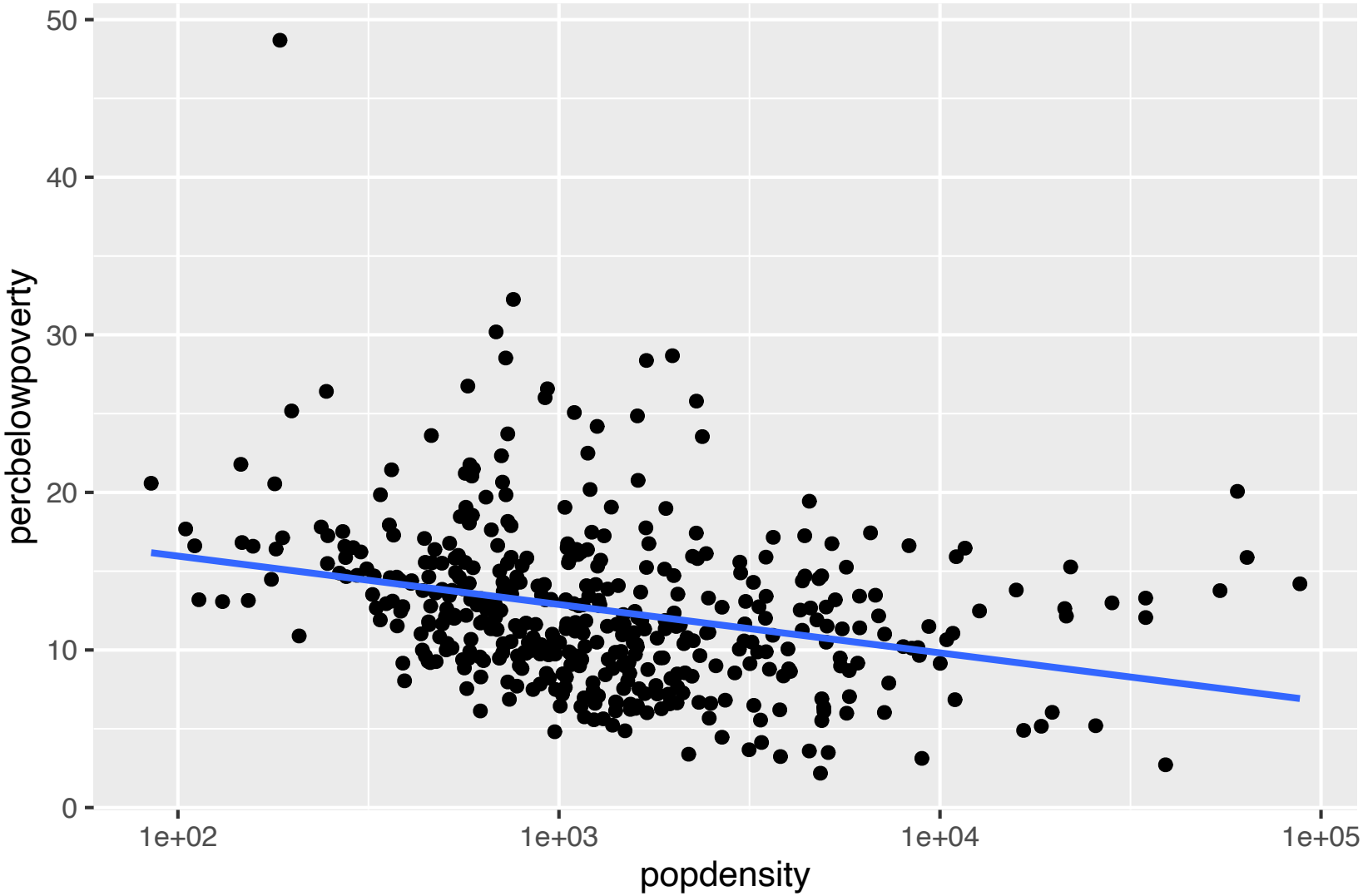


# Geoms are Functions

- Geoms can take input arguments:

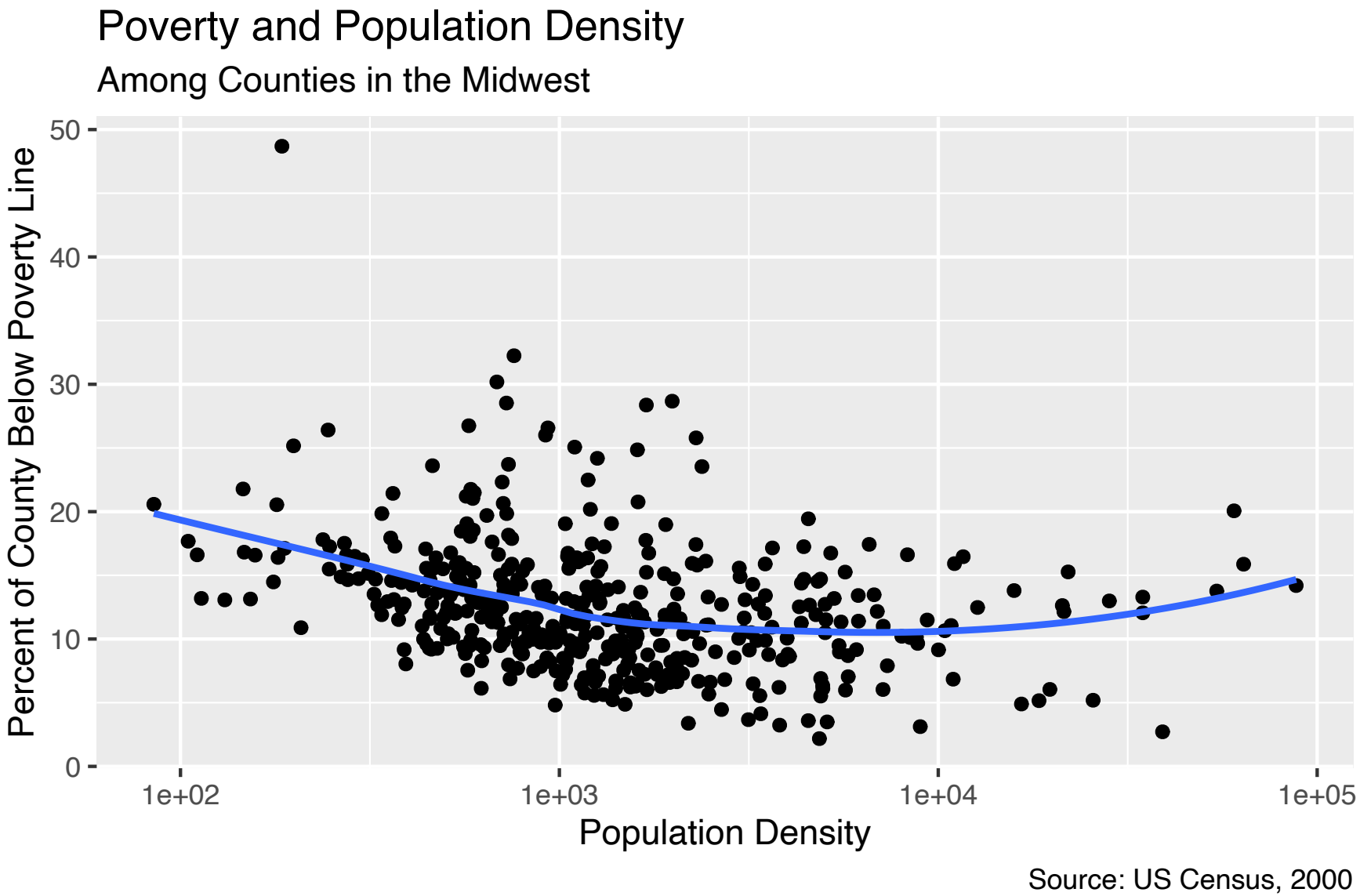
```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point() +
    geom_smooth(method = "lm", se = F) +
    scale_x_log10()
```

⇝ Tells `geom_smooth()` to do a linear fit with no error region.


50
40
30
20
10
0
percbelowpoverty
1e+02
1e+03
1e+04
1e+05
popdensity

# Adding Informative Labels

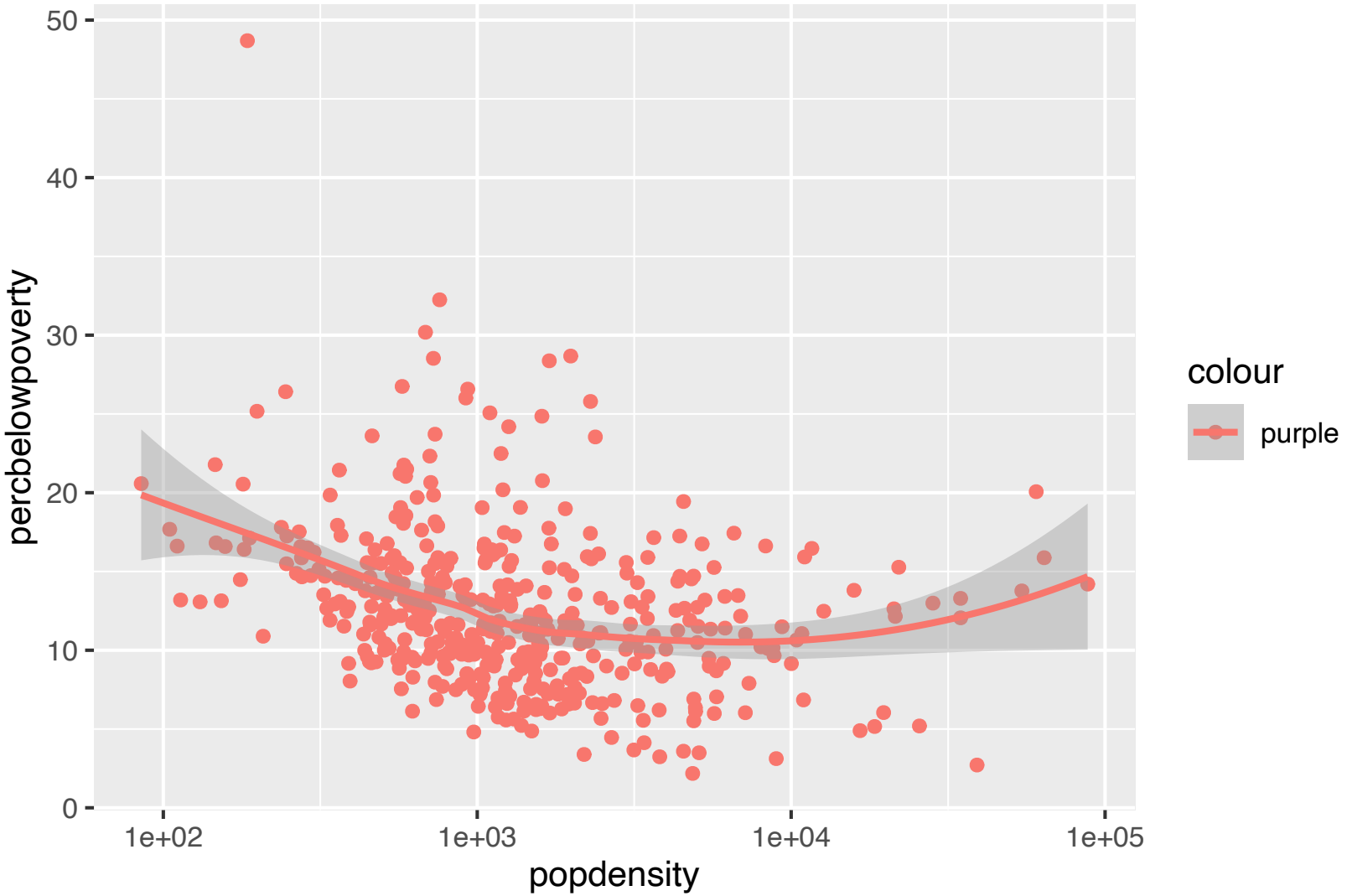
```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point() +
    geom_smooth(method = "loess", se = F) +
    scale_x_log10() +
    labs(x = "Population Density",
         y = "Percent of County Below Poverty Line",
         title = "Poverty and Population Density",
         subtitle = "Among Counties in the Midwest",
         caption = "Source: US Census, 2000")
```


Poverty and Population Density
Among Counties in the Midwest
Percent of County Below Poverty Line
50
40
30
20
10
0
1e+02
1e+03
1e+04
1e+05
Population Density
Source: US Census, 2000

# Mapping vs. Setting Aesthetics

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty,
                       color = "purple")) +
    geom_point() +
    geom_smooth() +
    scale_x_log10()
```

# Wait, What Happened?



# Wait, What Happened?

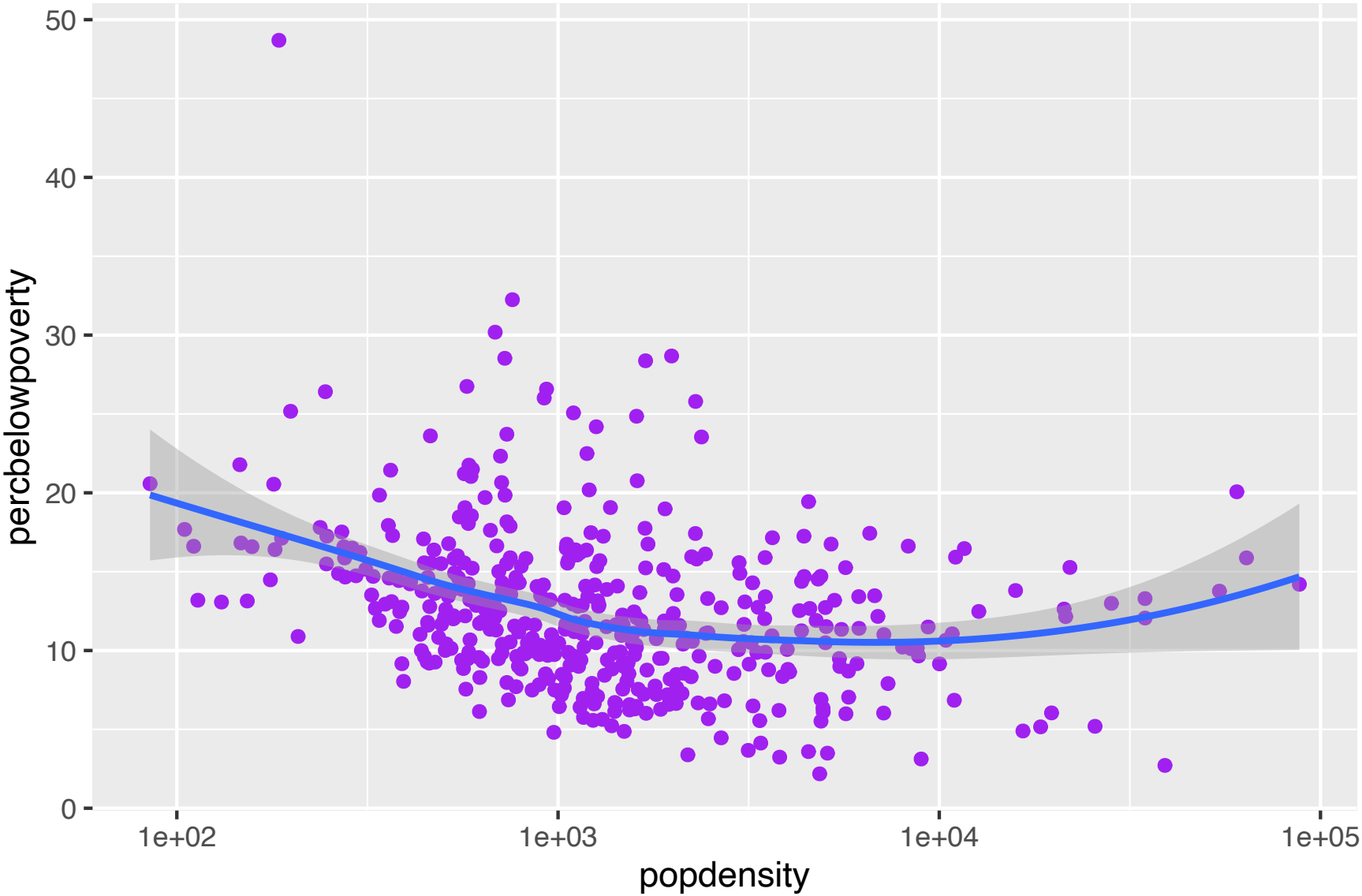
- Mapping always refers to variables!

  ⇝ If passed a value other than a variable name, ggplot will implicitly create a variable with that value (in this case “purple” that is constant.)

# Setting Aesthetics

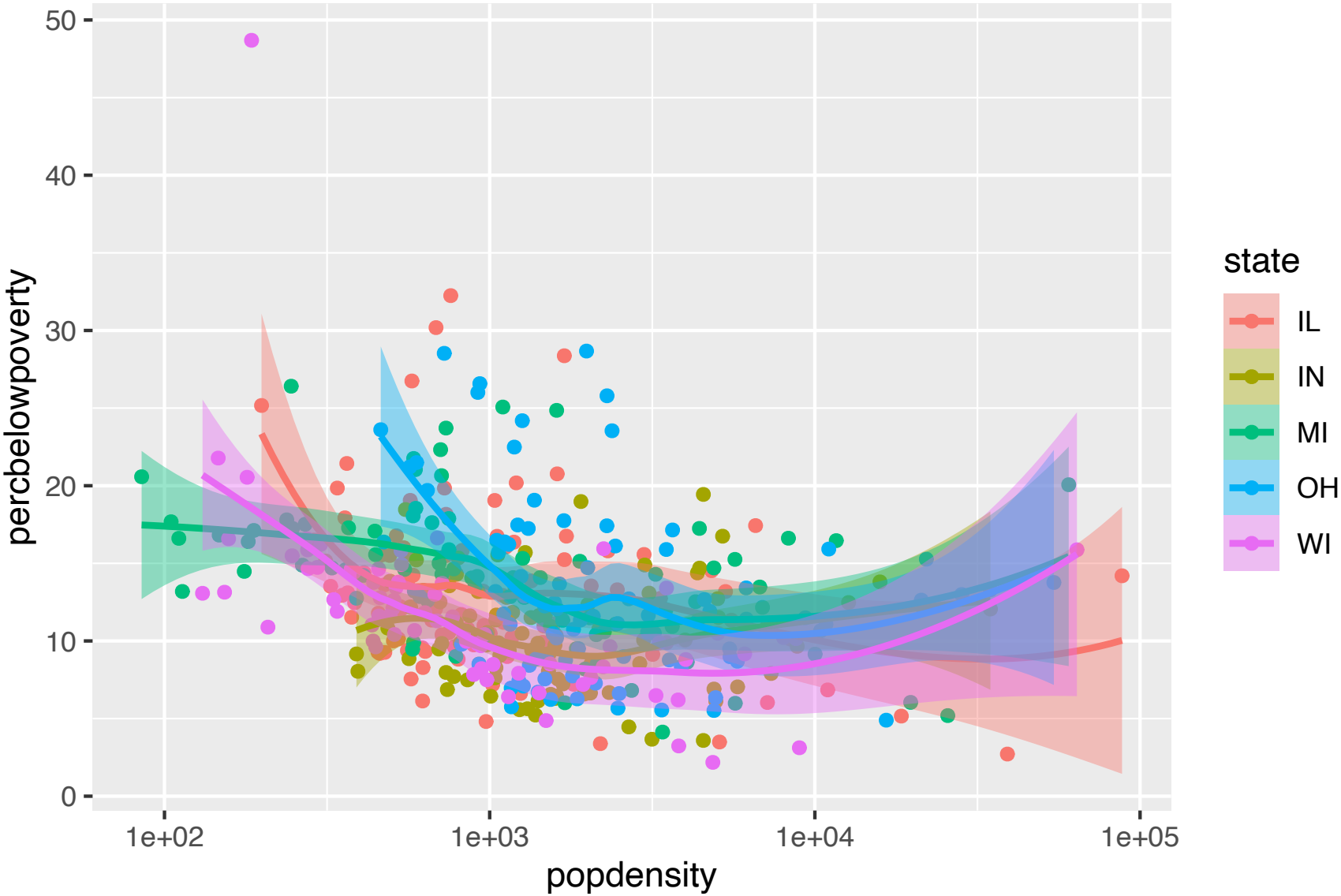
- Set the color outside the mapping (i.e., aes()) format:

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point(color = "purple") +
    geom_smooth() +
    scale_x_log10()
```


50
40
30
20
10
0
percbelowpoverty
1e+02
1e+03
1e+04
1e+05
popdensity

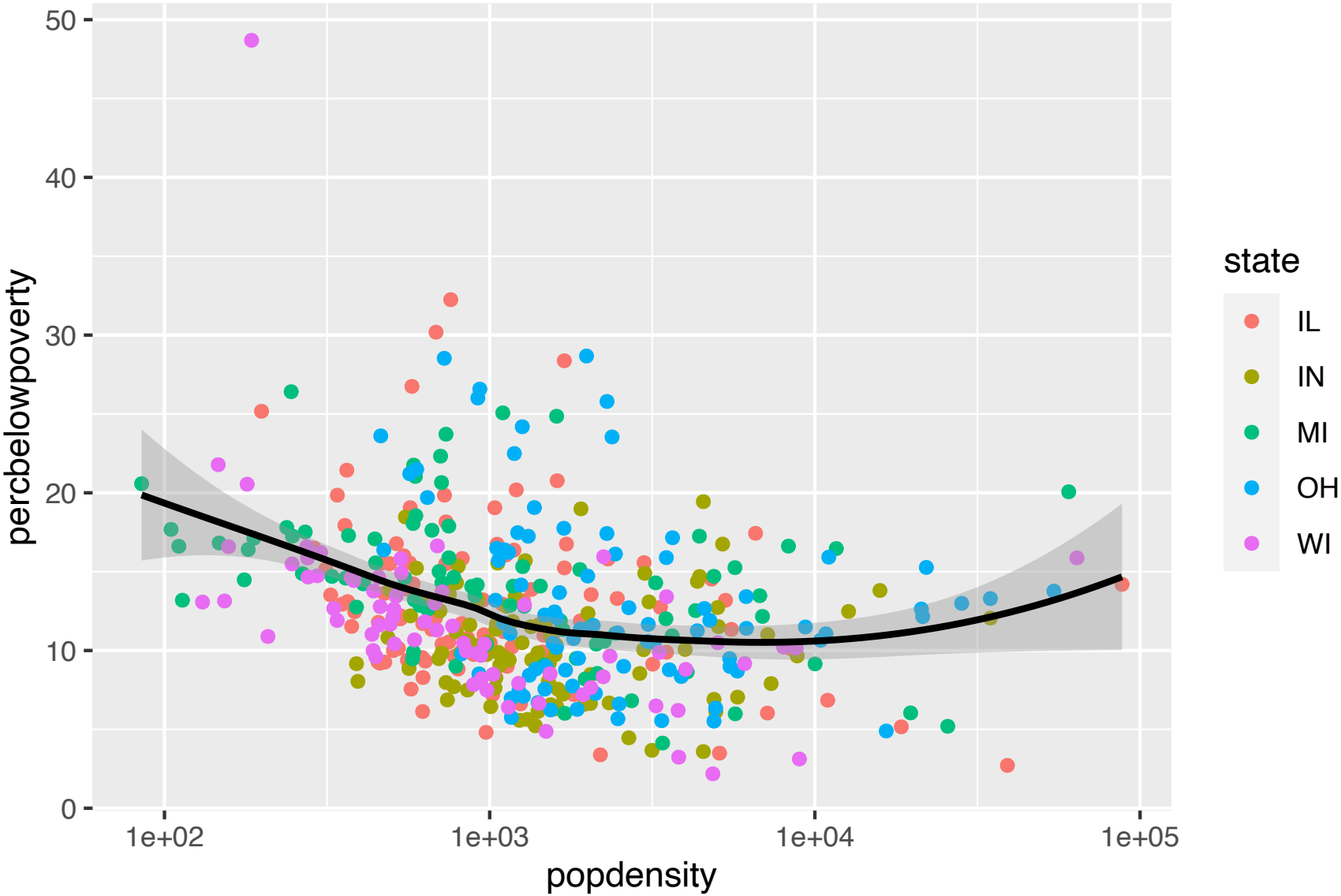
# Mapping More Aesthetics

```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty,
                       color = state,
                       fill = state)) +
    geom_point() +
    geom_smooth() +
    scale_x_log10()
```


50
40
30
20
10
0
percbelowpoverty
1e+02
1e+03
1e+04
1e+05
popdensity
state
IL
IN
MI
OH
WI

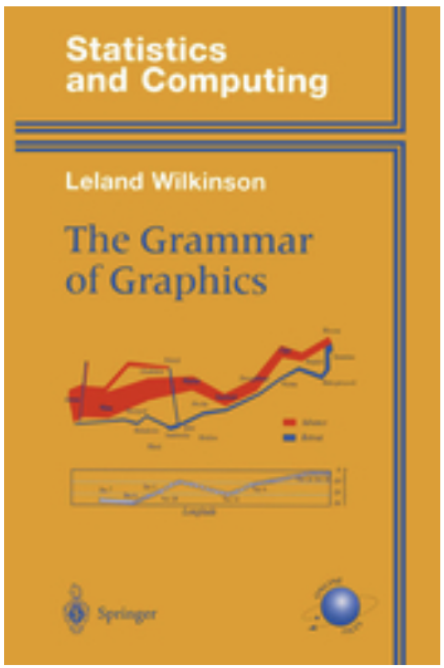
# Mapping Can be Done on a Per Geom Basis

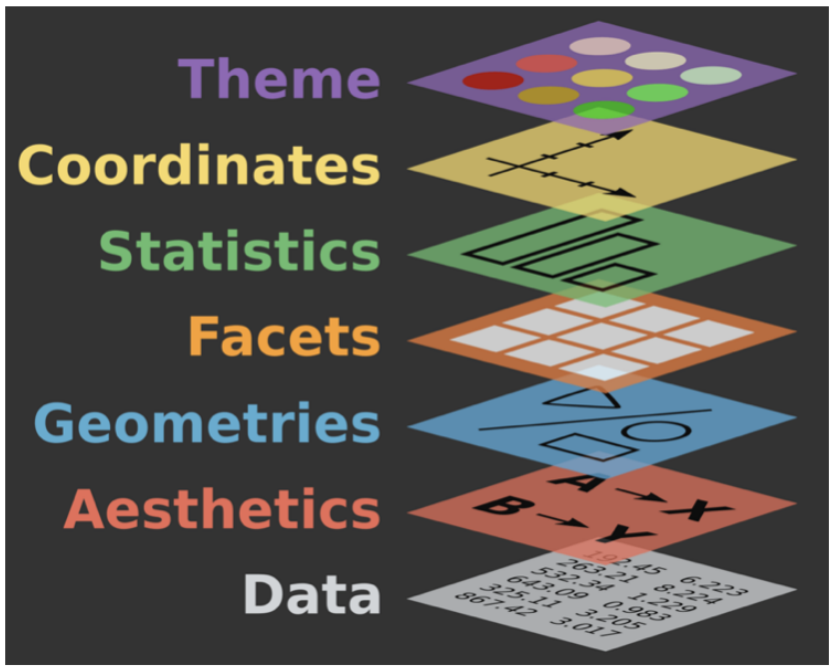
```
> ggplot(data = midwest,
         mapping = aes(x = popdensity,
                       y = percbelowpoverty)) +
    geom_point(mapping = aes(color = state)) +
    geom_smooth(color = "black") +
    scale_x_log10()
```


50
40
30
20
10
0
percbelowpoverty
1e+02
1e+03
1e+04
1e+05
popdensity
state
IL
IN
MI
OH
WI

# ggplot2: The Grammar of Graphics

*"...the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinate system" – Leland Wilkinson*



# 2/ Histograms, Boxplots, and Grouped Data

# Midwest Data Redux

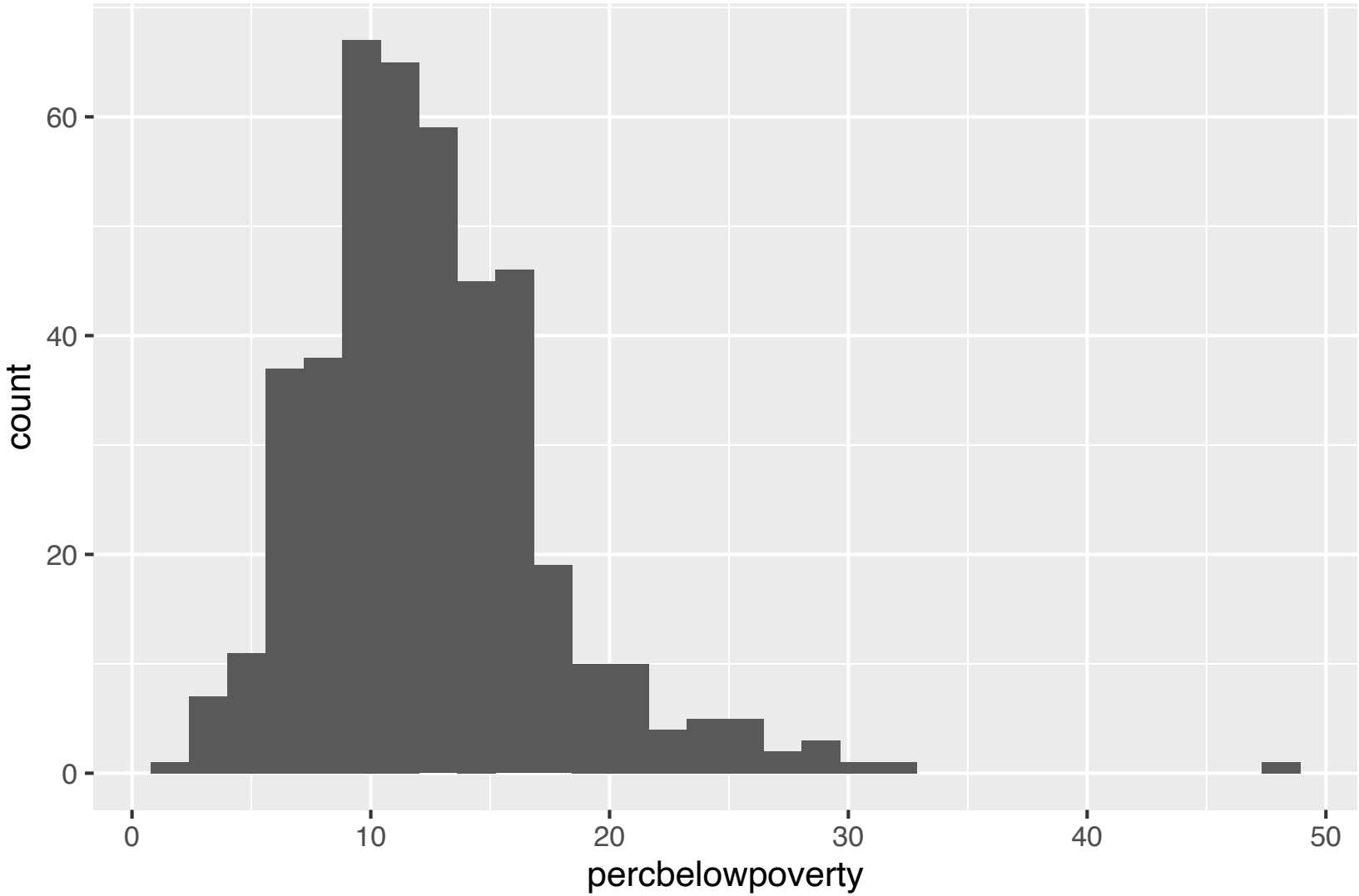
```
> ggplot2::midwest ## If needed: install.packages("ggplot2")

# A tibble: 437 × 28
     PID county  state  area poptotal popdensity popwhite popblack popamerindian popasian popother
   <int> <chr>   <chr> <dbl>    <int>      <dbl>    <int>    <int>         <int>    <int>    <int>
 1   561 ADAMS   IL    0.052    66090      1271.    63917     1702            98      249      124
 2   562 ALEXAN  IL    0.014    10626       759      7054     3496            19       48        9
 3   563 BOND    IL    0.022    14991       681.    14477      429            35       16       34
 4   564 BOONE   IL    0.017    30806      1812.    29344      127            46      150     1139
 5   565 BROWN   IL    0.018     5836       324.     5264      547            14        5        6
 6   566 BUREAU  IL    0.05     35688       714.    35157       50            65      195      221
 7   567 CALHOUN IL    0.017     5322       313.     5298        1             8       15        0
 8   568 CARROLL IL    0.027    16805       622.    16519      111            30       61       84
 9   569 CASS    IL    0.024    13437       560.    13384       16             8       23        6
10   570 CHAMPA  IL    0.058   173025      2983.   146506    16559           331     8033     1596
# ℹ 427 more rows
# ℹ 17 more variables: percwhite <dbl>, percblack <dbl>, percamerindan <dbl>, percasian <dbl>,
#   percother <dbl>, popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,
#   poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,
#   percchildbelowpovert <dbl>, percadultpoverty <dbl>, percelderlypoverty <dbl>, inmetro <int>,
#   category <chr>
# ℹ Use `print(n = ...)` to see more rows
```

# Histograms

- **Histograms** show where there are more or fewer observations of a numeric variable.

```
> require(ggplot2)
> ggplot(data = midwest,
         mapping = aes(x = percbelowpoverty)) +
    geom_histogram()
```

- Split up range of variable into bins, count how many are in each bin.
- y aesthetic calculated automatically.



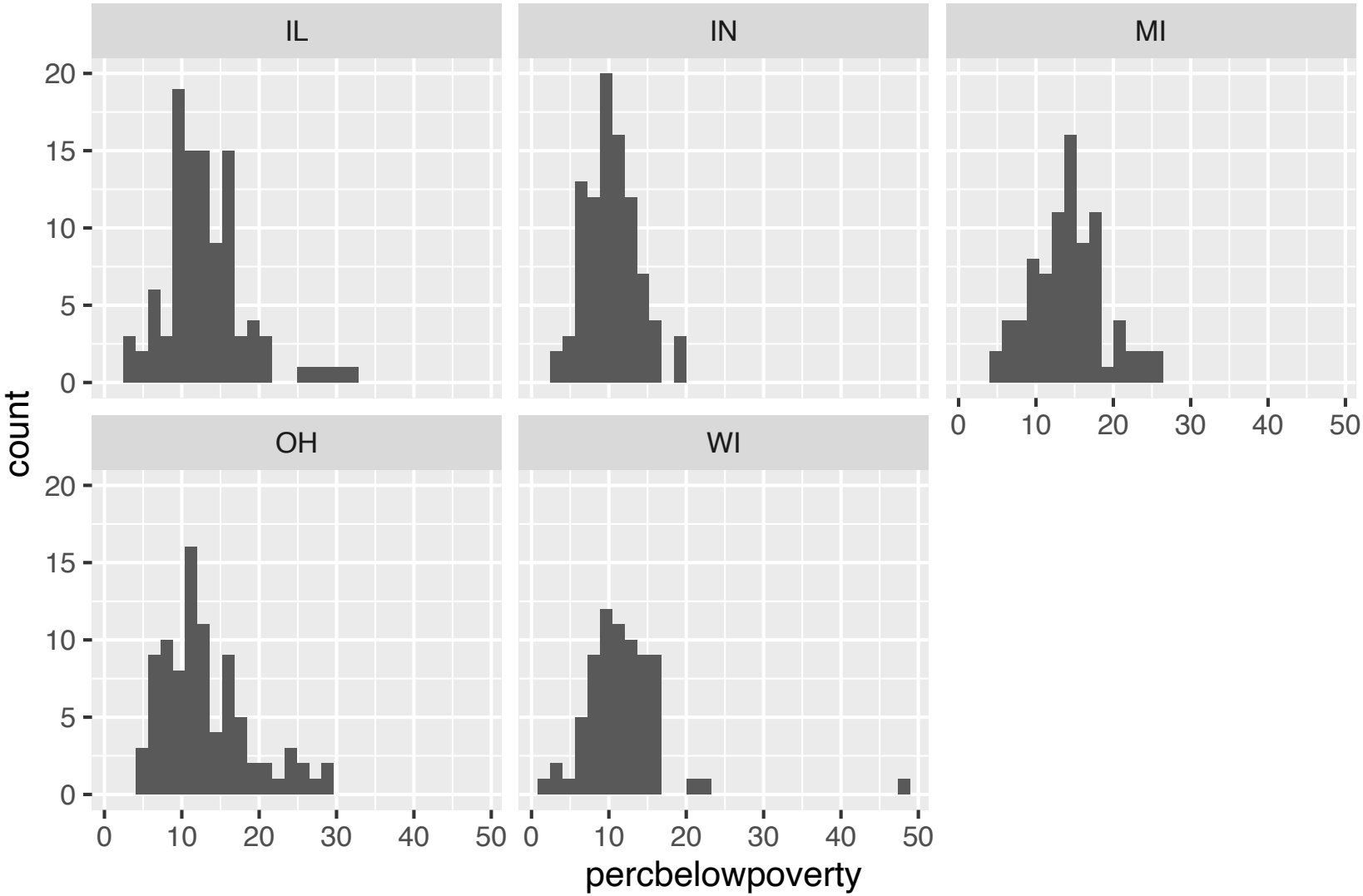

**Take away:** a typical county has somewhere around 10 to 15% of its population below the poverty line, and the distribution is right-skewed: a long tail of a few counties far out at 30% and beyond.

# Creating Small Multiples with Facets

- Small multiples: a series of similar graphs with the same scale/axes to help with comparing different partitions of a dataset

```
> ggplot(data = midwest,
         mapping = aes(x = percbelowpoverty)) +
    geom_histogram() +
    facet_wrap(~ state)
```

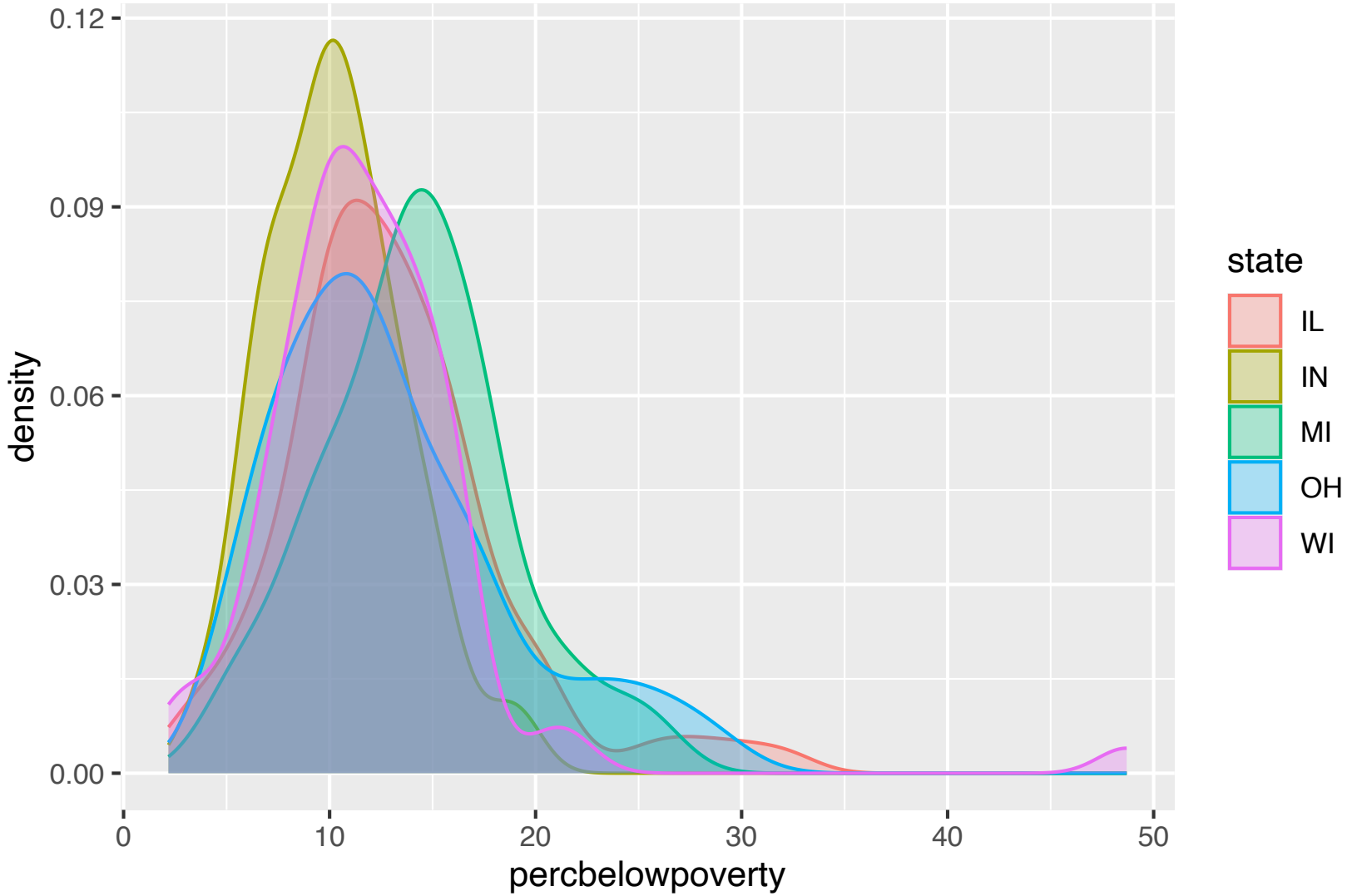
- The variable used for faceting/spliting must be placed on the right-hand side of the tilde (`~`) operator
  - We'll see more of the `~ variable` syntax (called a formula) throughout the semester!


IL
IN
MI
OH
WI
count
percbelowpoverty
0
5
10
15
20
0
10
20
30
40
50

# Density as Alternative to Histogram

- A kernel density plot is a smoothed version of a histogram and slightly easier to overlay.

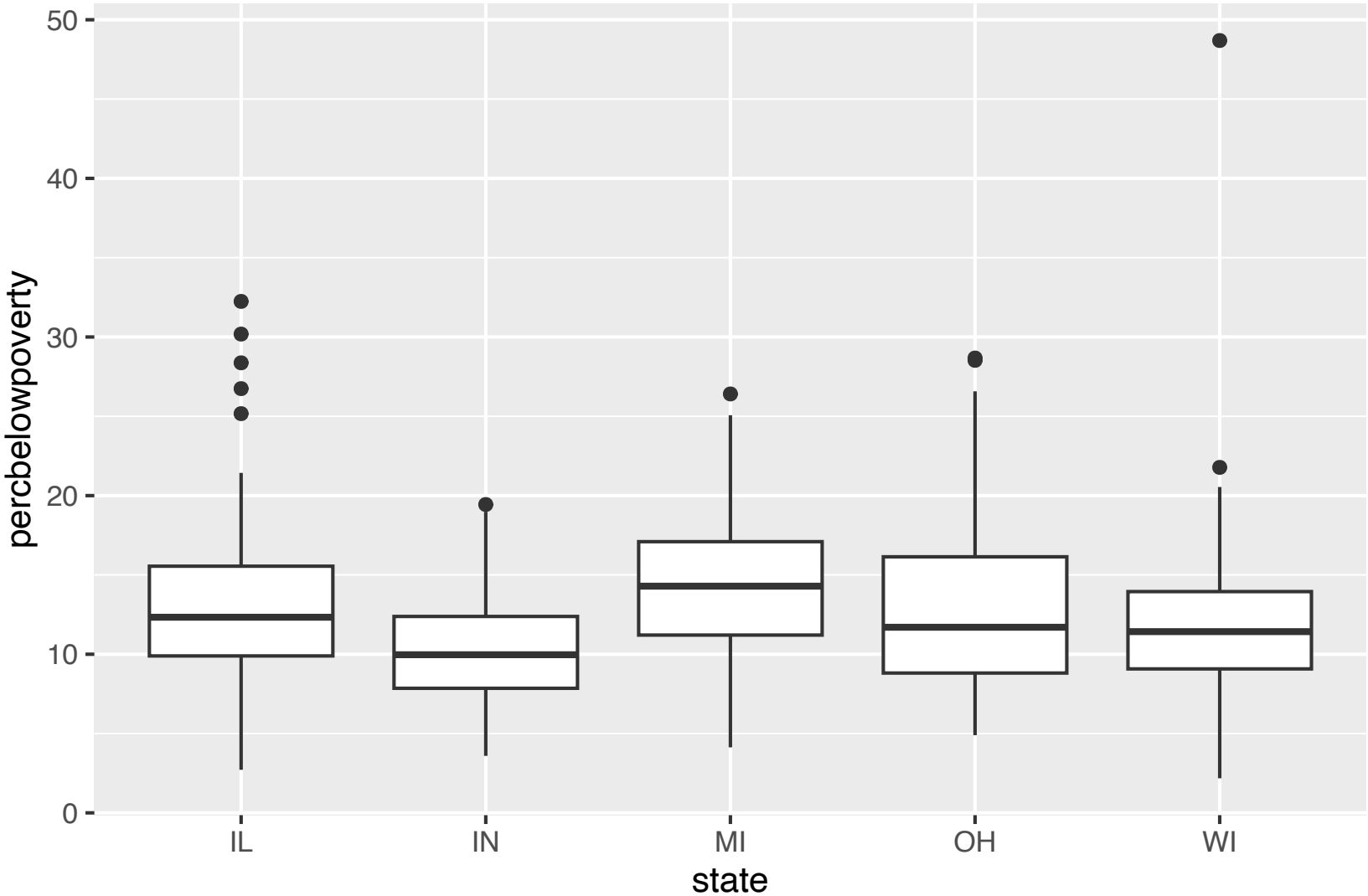
```
> ggplot(data = midwest,
         mapping = aes(x = percbelowpoverty,
                       fill = state, color = state)) +
    geom_density(alpha = 0.3)
```


0.12
0.09
0.06
0.03
0.00
density
0
10
20
30
40
50
percbelowpoverty
state
IL
IN
MI
OH
WI

# Boxplots

- Boxplots are another way to compare distributions across discrete groups.

```
> ggplot(data = midwest,
         mapping = aes(x = state,
                       y = percbelowpoverty)) +
    geom_boxplot()
```


50
40
30
20
10
0
percbelowpoverty
IL
IN
MI
OH
WI
state

# More on Boxplots

- "Box" represents middle 50% of the data.
  - 25% of the data above the box, 25% below
  - Height of the box is called the inter quartile range (IQR)
- Horizontal line in the box is the median
  - 50% of the data above the median, 50% below
- "Whiskers" extend to either:
  - the farthest data point within $1.5 \times$ IQR of the box, or the maximum (minimum for the lower whisker), whichever is closer
  - Data points beyond whiskers are outliers, by this rule; other definitions exist, and we return to them later in the semester

**Visualizing grouped data accurately is key to conveying the right insights and avoiding misleading interpretation.**

# Gapminder Data

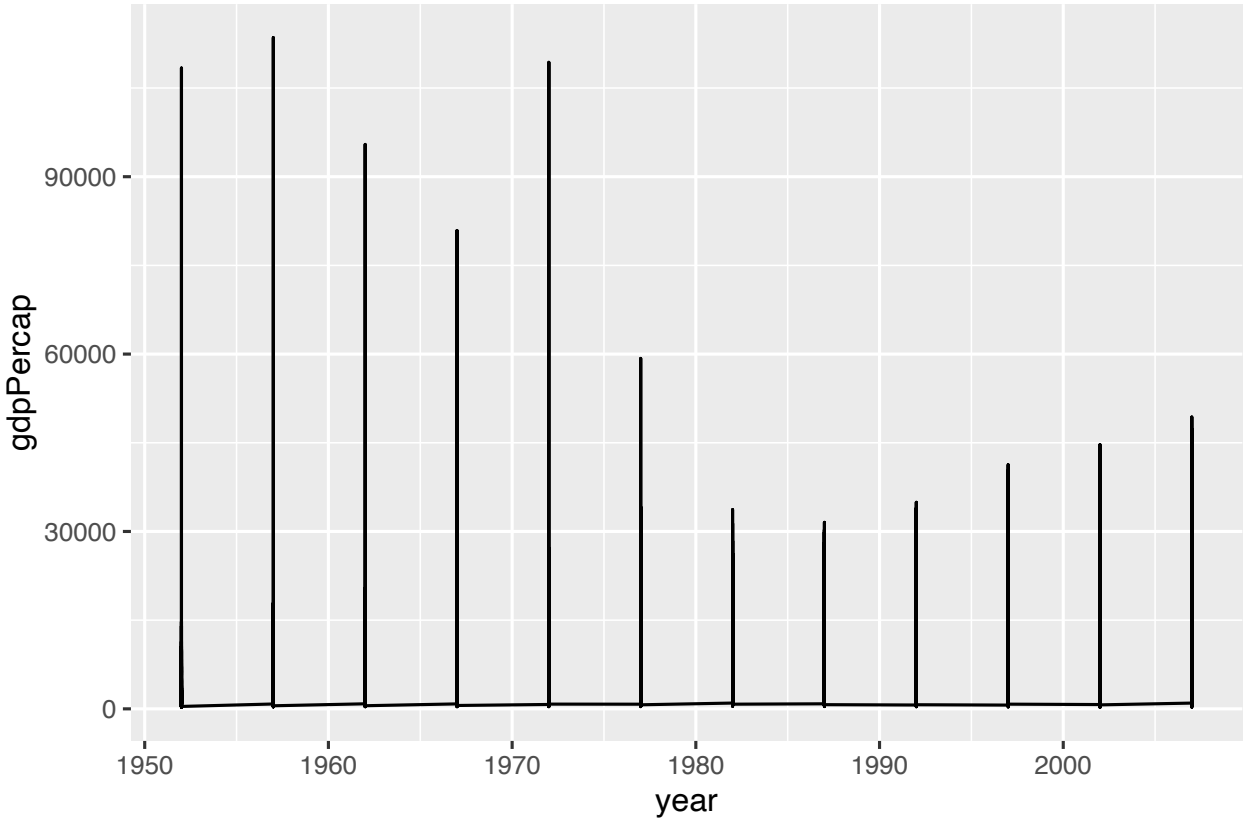
```
> require(gapminder)
> dplyr::glimpse(gapminder)

Rows: 1,704
Columns: 6
$ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", "Afgha. . .
$ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, . . .
$ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, 2002, . . .
$ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.822, 41. . .
$ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12881816. . .
$ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, 978.01. . .
```

- Let's say we are interested in learning about the trend in income in different countries.
  - ⇝ Use `geom_line()` to create a line plot.

# Let’s plot the trend in income.. wait what?

```
> ggplot(data = gapminder,
         mapping = aes(x = year,
                       y = gdpPercap)) +
    geom_line() # Didn't specify how to group the lines..
```

# Let's plot the trend in income.. wait what?

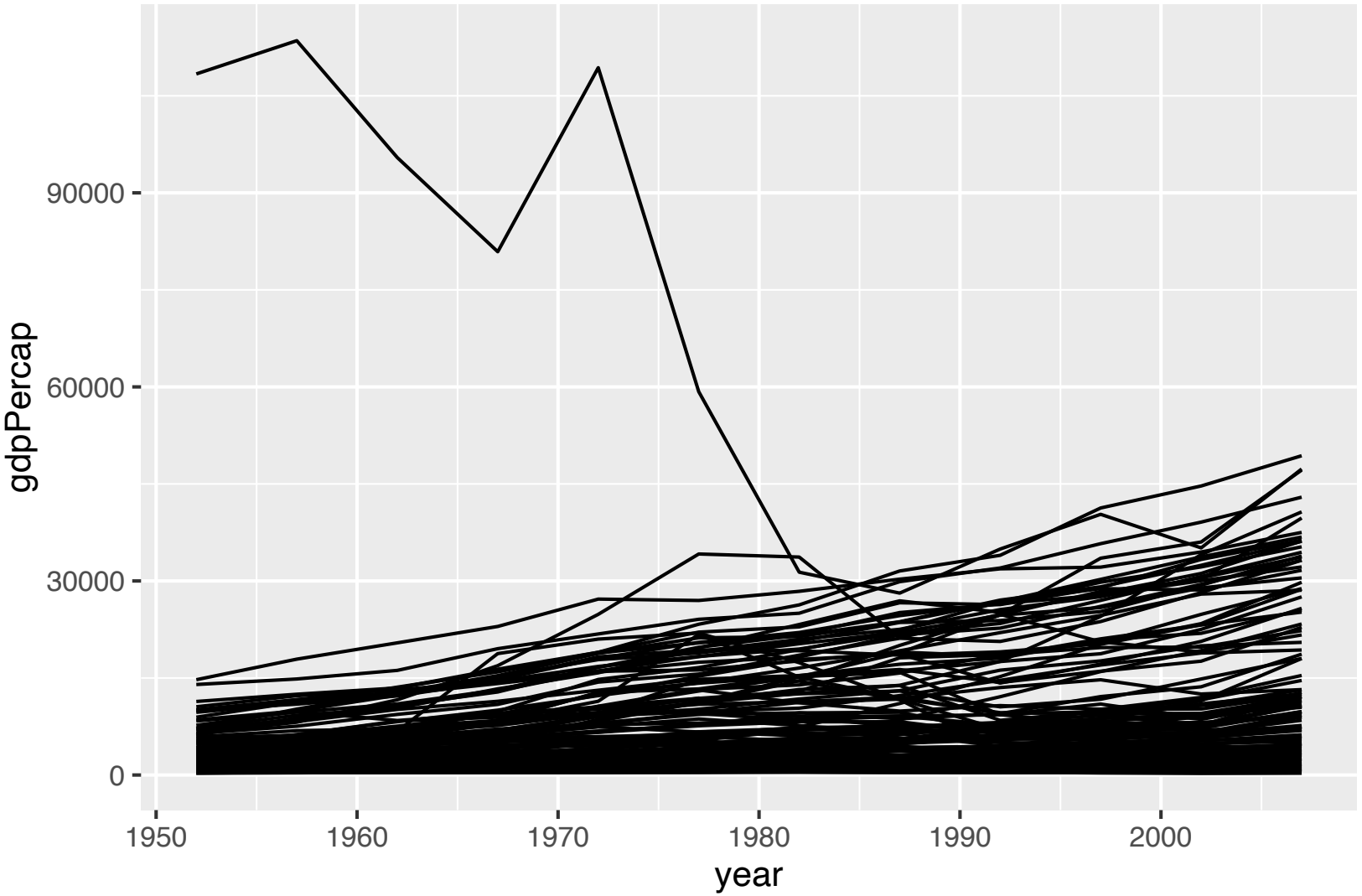



- `geom_line()` connected points from different countries in the same year.
  - What we wanted was to connect the points from same countries in different years (i.e., time trends)!

# Tell `geom_line()` how to group the lines

```
> ggplot(data = gapminder,
         mapping = aes(x = year,
                       y = gdpPercap)) +
    geom_line(mapping = aes(group = country))
```

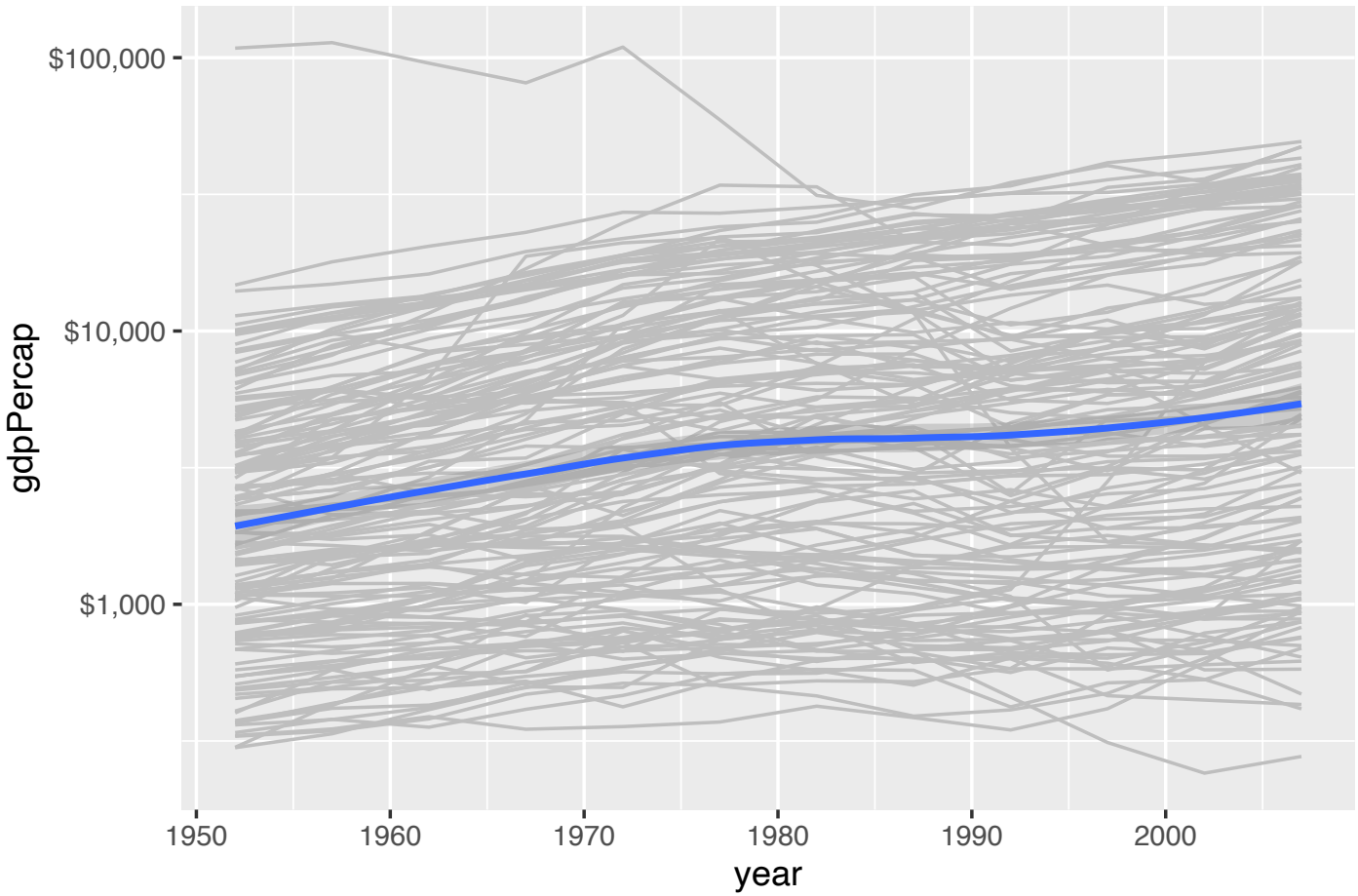
# Tell `geom_line()` how to group the lines



# Adjusting scales and adding a trendline

```
> ggplot(data = gapminder,
         mapping = aes(x = year,
                       y = gdpPercap)) +
    geom_line(mapping = aes(group = country), color = "gray") +
    geom_smooth(method = "loess") +
    scale_y_log10(labels = scales::dollar)
```

# Adjusting scales and adding a trendline



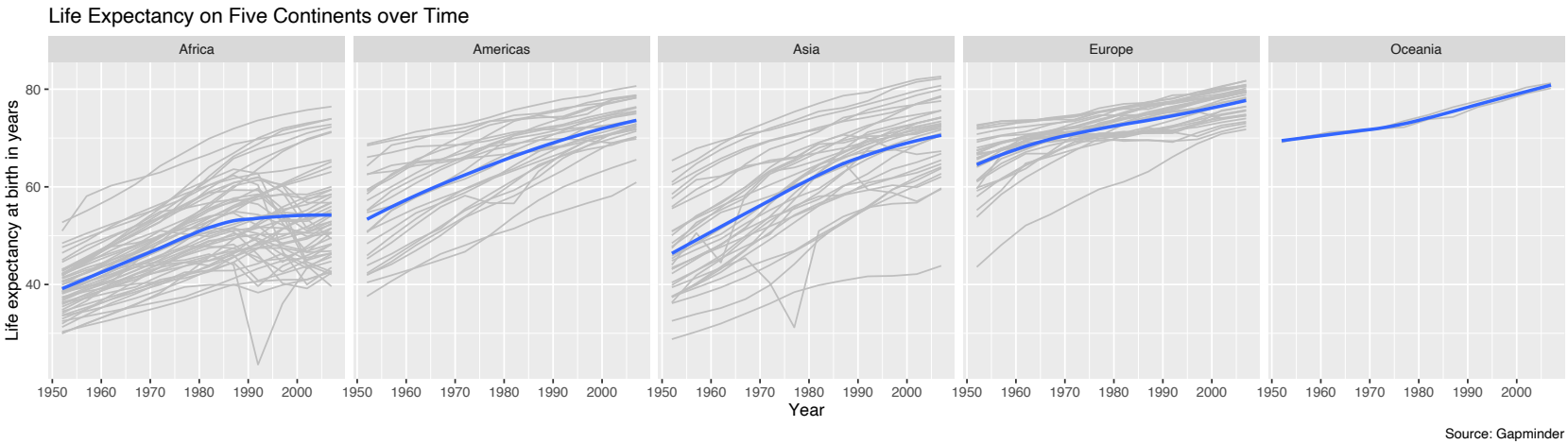

Q: What do we observe? Why do you think this is the case? (innovation? capital accumulation?)

# In-Class Exercises: Background

- We want to find out the answer to the following question:
  - Q: how does life expectancy on five continents (e.g., Asia, Europe, Americas) change over time?
- The `gapminder` dataset contains some information about life expectancy across different continent, thus we could visualize the data to answer the question.
  - Some relevant variables:
    - `country`: name of the country
    - `continent`: name of the country's continent
    - `year`: year of the measurement, ranging from 1952 to 2007 in 5-year increments
    - `lifeExp`: life expectancy at birth, in years

# In-Class Exercises: Task

- Recreate the following plot:




- The plot shows each country's life expectancy trajectory over time, broken out by continent with smoothed average lines overlayed for each continent.
    - To get started, we'll give you a few clues about what was done to create the plot in the following slide.

# In-Class Exercises: Instructions

- The lines for each country use the color "gray"
- The width of the smoothed line is 1.1 (`linewidth = 1.1`) and the method used is the loess smoother.
- Turn off the standard errors.
- Make sure that the facets are all on one row. Look at the `facet_wrap` documentation if need help with this.
- Make sure that the labels are correctly specified.

Q: What does the visualization reveal? Explain what you observe verbally.

# What Comes Next?

- Today: a table became a graph, layer by layer; the same data compared across distributions and groups.
- Next week: **data wrangling** with `dplyr`. Choose rows and columns, create variables, summarize groups, and draw barplots from the summaries.
- Before next class: complete the Required reading in the weekly schedule and finish swirl lessons 1–3. Use the R Basics reference materials as needed. Assignment 1 is out today.
- E1 (today's exercise): submit on eeclass. Check your section's eeclass announcement for the submission deadline.

# See You Next Week! :)

*Contact Information:*
jaewon.yoo@iss.nthu.edu.tw
https://j1yoo.github.io/