Introduction to the course (and R) and data visualization

Seung-Ho An, University of Arizona

Agenda

1. Introduction to the course
2. Course details
3. R and Rmarkdown Introduction
4. Data visualization

1. Introduction to the course

- Why do we need to learn statistics?
- Applied statistics? Data science?

Why do we need to learn statistics?

Why do we need to learn statistics?

1. To test ideas empirically

Why do we need to learn statistics?

1. To test ideas empirically
2. To make inferences from sample to population

Why do we need to learn statistics?

1. To test ideas empirically
2. To make inferences from sample to population
3. To make evidence-based decisions/policy

Applied statistics? Data science?

- Applied statistics includes planning for the collection of data, managing data, analyzing, interpreting and drawing conclusions from data, and identifying problems, solutions and opportunities using the analysis (UM-Dearborn)

Applied statistics? Data science?

- Applied statistics includes planning for the collection of data, managing data, analyzing, interpreting and drawing conclusions from data, and identifying problems, solutions and opportunities using the analysis (UM-Dearborn)
- How about the data science process?

Applied statistics? Data science?

- Applied statistics includes planning for the collection of data, managing data, analyzing, interpreting and drawing conclusions from data, and identifying problems, solutions and opportunities using the analysis (UM-Dearborn)
- How about the data science process?

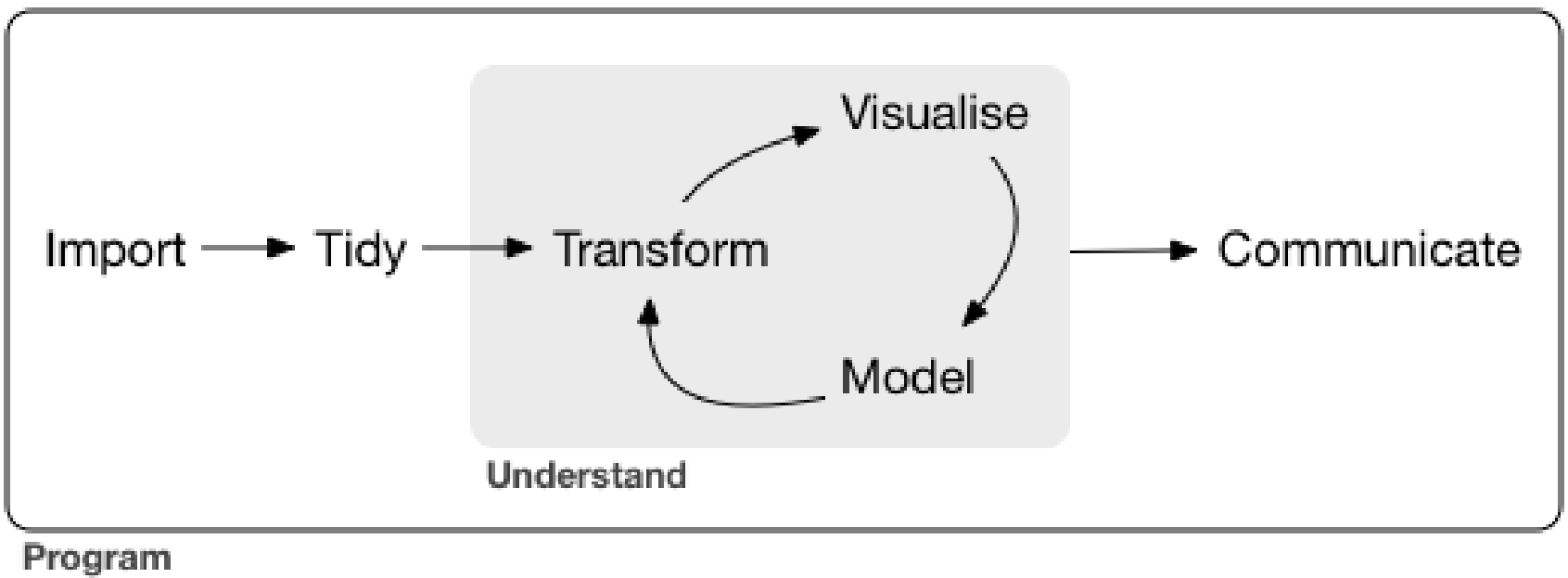


Figure 1: Data science process (Hadley)

Our approach

The mix of applied statistics and data science, slightly leaning more toward data science

Our approach

The mix of applied statistics and data science, slightly leaning more toward data science

E.g., calculating a sample variance of a set $\{2, 7, 7, 4, 5, 1, 3\}$

The mix of applied statistics and data science, slightly leaning more toward data science

E.g., calculating a sample variance of a set $\{2, 7, 7, 4, 5, 1, 3\}$

Instead of this,

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = \frac{\sum_{i=1}^{n} x^2 - \frac{(\sum_{i=1}^{n} x)^2}{n}}{n-1}$$

$$= \frac{(4+49+49+16+25+1+9) - \frac{(2+7+7+4+5+1+3)^2}{7}}{7-1} = \frac{153 - \frac{841}{7}}{6}$$

$$= 5.47619$$

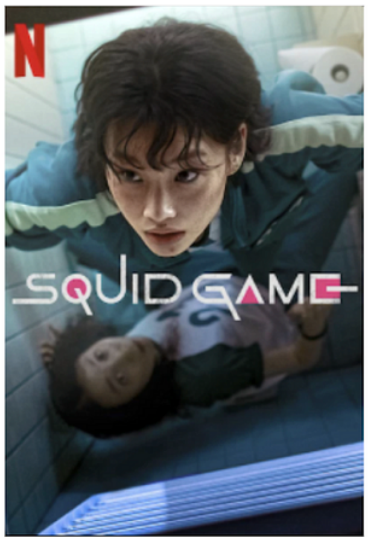
We will focus more on

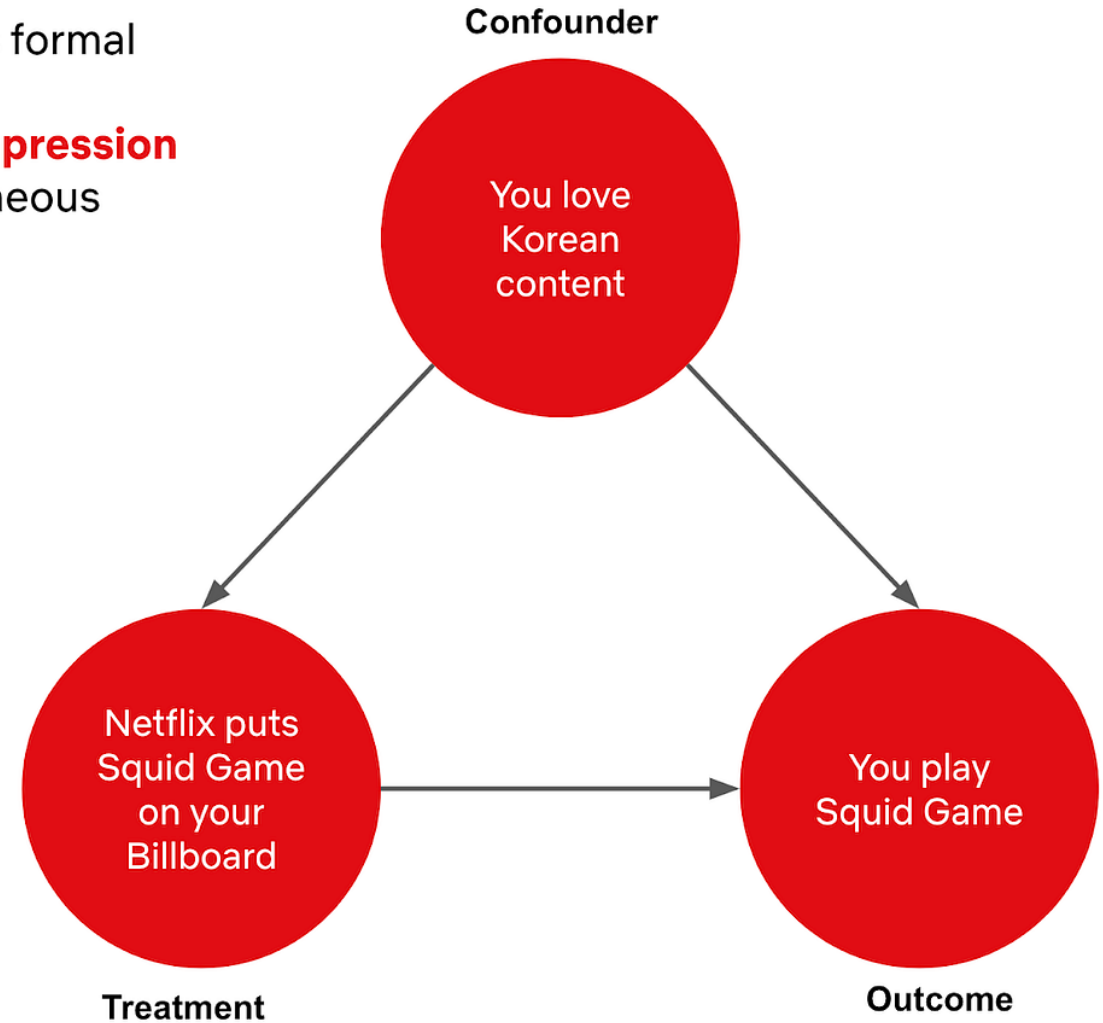
```
x<-c(2, 7, 7, 4, 5, 1, 3)
var(x)
```

```
## [1] 5.47619
```

What problems are applied statisticians/data scientists working on?

Causal Inference provides formal tools to tease out the true **incremental** value of an **impression** for each profile: Heterogeneous Treatment Effect (**HTE**)



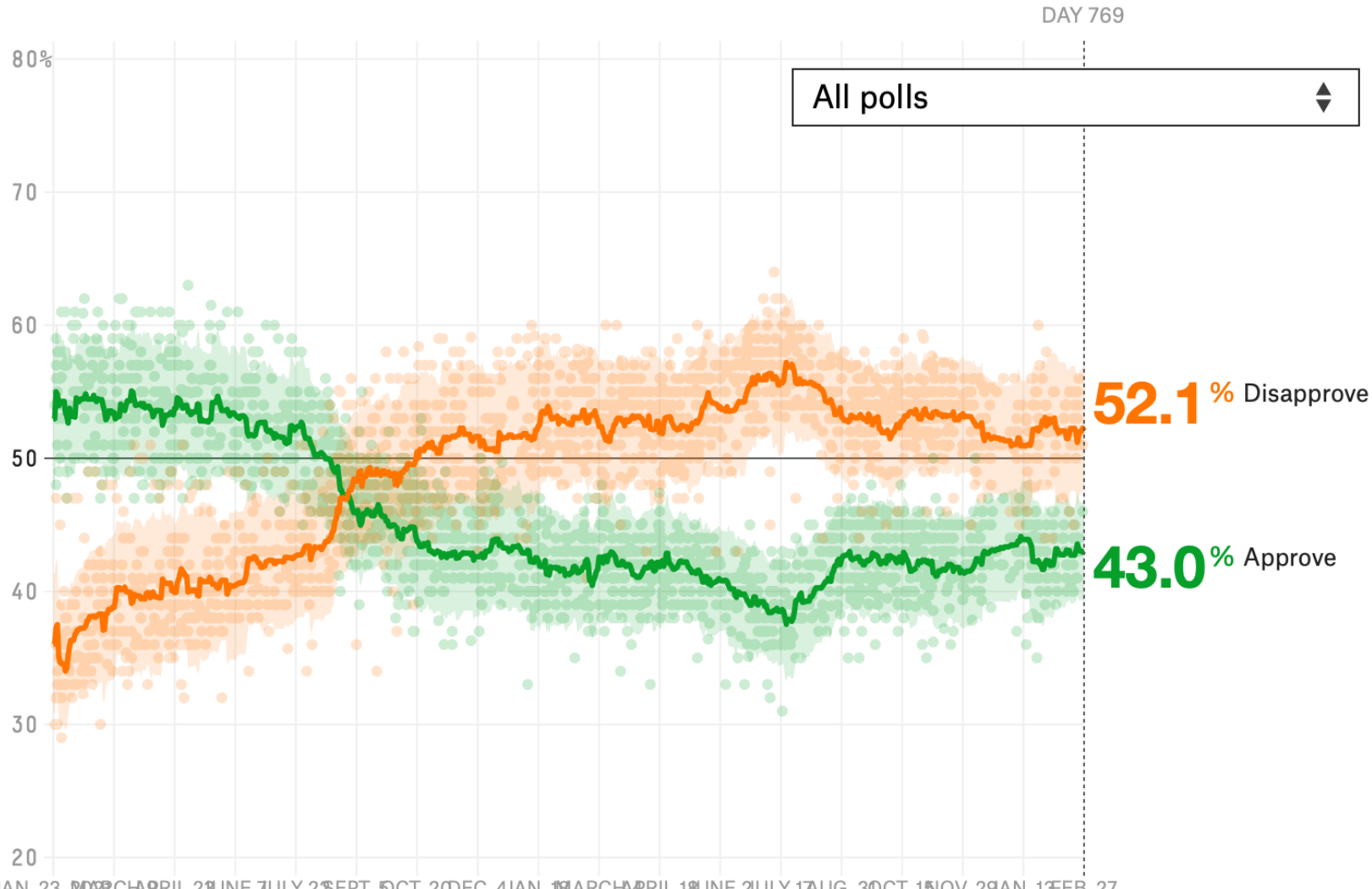


A new computer model uses publicly available data to predict crime accurately in eight U.S. cities, while revealing increased police response in wealthy neighborhoods at the expense of less advantaged areas.

June 30, 2022

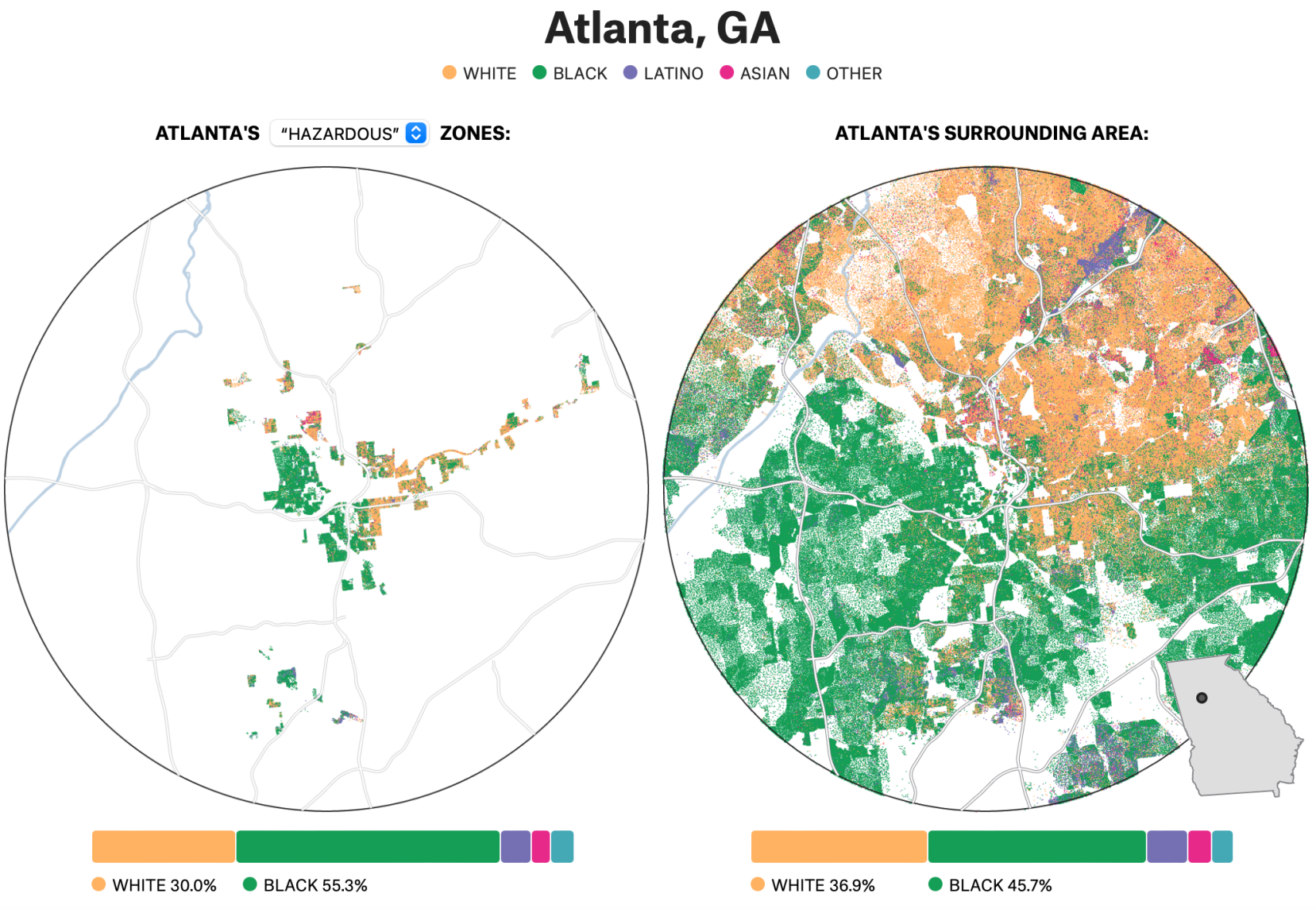
How popular is Joe Biden?

An updating calculation of the president's approval rating, accounting for each poll's quality, recency, sample size and partisan lean. How this works »



Visualization

Understanding how the past matters



2. Course details

Name, N of previous stat/programming classes, main statistical program/package (if any), etc.

What will you learn in this class?

- Summerize and visualize data

What will you learn in this class?

- Summerize and visualize data
- Wrangle messy data into tidy forms

What will you learn in this class?

- Summerize and visualize data
- Wrangle messy data into tidy forms
- Evaluate claims about causality

What will you learn in this class?

- Summerize and visualize data
- Wrangle messy data into tidy forms
- Evaluate claims about causality
- Be able to use linear regression to analyze data

What will you learn in this class?

- Summerize and visualize data
- Wrangle messy data into tidy forms
- Evaluate claims about causality
- Be able to use linear regression to analyze data
- Understand uncertainty in data analysis and how to quantify it

What will you learn in this class?

- Summerize and visualize data
- Wrangle messy data into tidy forms
- Evaluate claims about causality
- Be able to use linear regression to analyze data
- Understand uncertainty in data analysis and how to quantify it
- Use professional tools like R and RStudio

- Deliberate pacing and **tons** of support

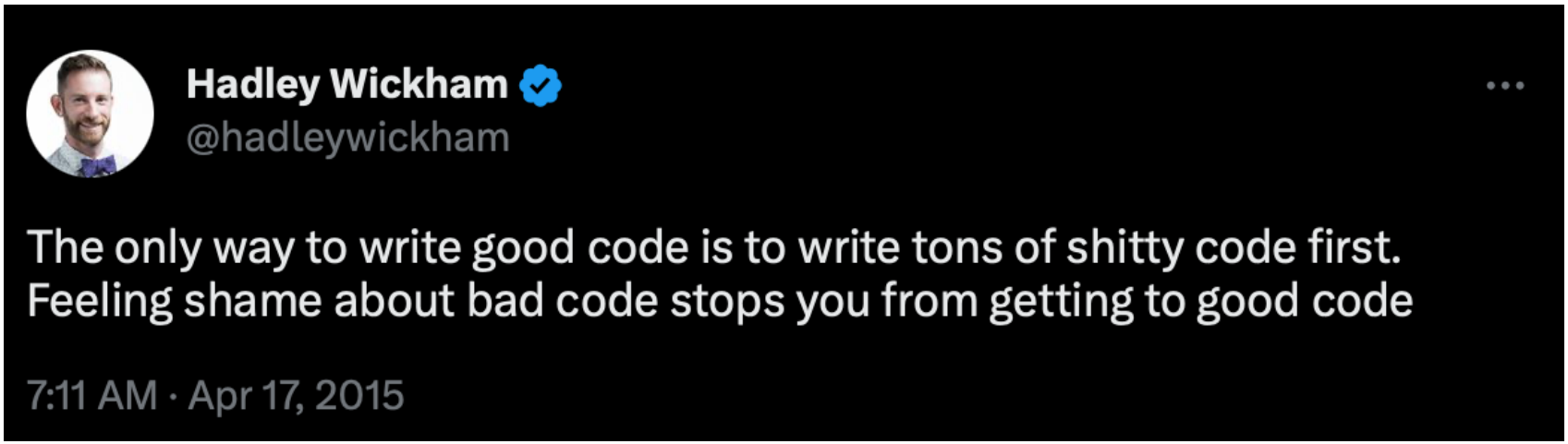
Teaching philosophy

- Deliberate pacing and **tons** of support
- Emphasize intuition and computational approaches over mathematical equations

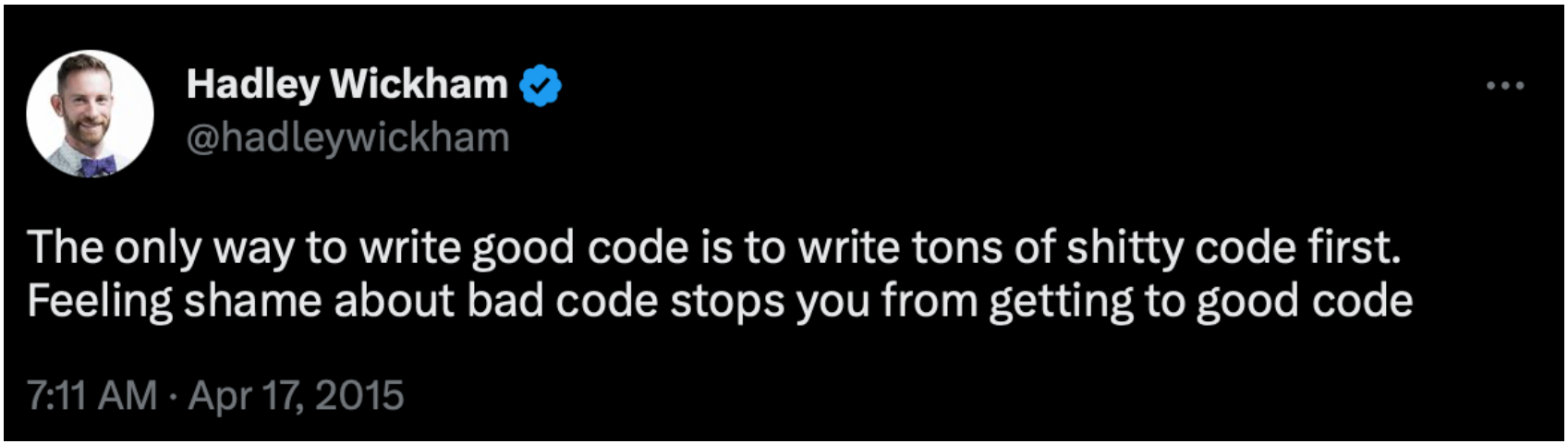
Teaching philosophy

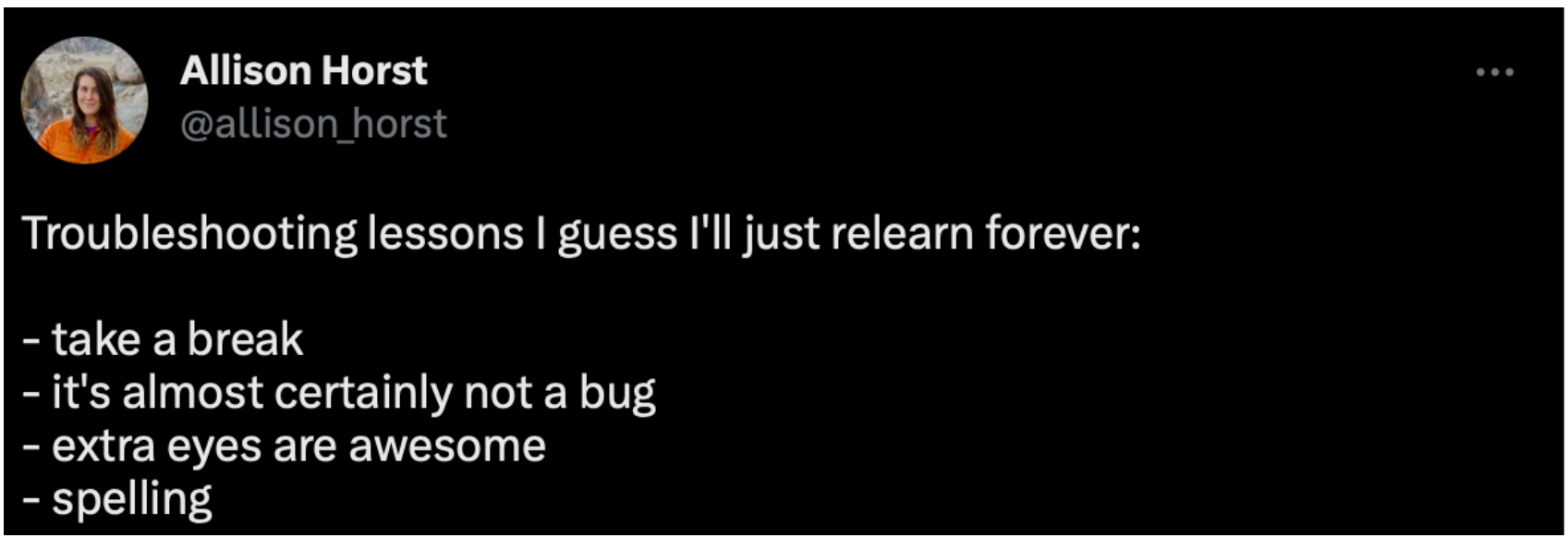
- Deliberate pacing and **tons** of support
- Emphasize intuition and computational approaches over mathematical equations
- Practice, practice, practice

How to become a better programmer



How to become a better programmer





Prerequisites of the course

- **None** (no prior programming or statistics knowledge)
  - I am excited to teach this class with a room full of individuals who are just enthusiastic about data and analyses as much as I am!

- Three text book:
  - Modern Dive (MD) (free online)
  - Quantitative Social Science (QSS): An Introduction in tidyverse by Kosuke Imai (not free)
  - Introduction to Modern Statistics (IDS) (free online)
- Sometimes same materials in two/three different books. Choose which helps most.

- Three text book:
  - Modern Dive (MD) (free online)
  - Quantitative Social Science (QSS): An Introduction in tidyverse by Kosuke Imai (not free)
  - Introduction to Modern Statistics (IDS) (free online)
- Sometimes same materials in two/three different books. Choose which helps most.
- Lectures:
  - Broad coverage of the course materials
  - Coding demonstrations (follow along with your laptop) (if any)
  - Slides will be posted to the website shortly (or the night) before lecture

Assignments

- Roughly weekly homework
  - It will be posted on Thursday morning, and due following Wednesday
  - Your assignments will not be graded but highly encourage finishing those!
- Final project (optional), though I highly encourage everyone to do so!
  - Especially if you find assignments too easy, this would be an opportunity to practice your data/analysis skills (of course with tons of my support/assistance)! At the end, I will provide brief comments on your project.

Assignments

- Roughly weekly homework
  - It will be posted on Thursday morning, and due following Wednesday
  - Your assignments will not be graded but highly encourage finishing those!
- Final project (optional), though I highly encourage everyone to do so!
  - Especially if you find assignments too easy, this would be an opportunity to practice your data/analysis skills (of course with tons of my support/assistance)! At the end, I will provide brief comments on your project.
- Another option for advanced individuals: after a couple of assignments, if the assignments are still easy, let me know; I will assign relevant exercises from the QSS book.

- We will use the R statistical environment to analyze data
  - It's free
  - A popular option for data analysis
  - Academics, 538, NYT, Facebook, Google, Twitter, nonprofits, governments all use R
- Interface with R via a program called R studio

3. R and Rmarkdown introduction

Two computer revolutions

The frontier of computing

Where statistical computing lives

- Touch-based interfaces

Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time

Where statistical computing lives

Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps

Where statistical computing lives

Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system

Where statistical computing lives

- Windows and pointers

Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system

Where statistical computing lives

- Windows and pointers
- Multi-tsking, multiple windows

Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system

Where statistical computing lives

- Windows and pointers
- Multi-tsking, multiple windows
- Works heavily with the file system

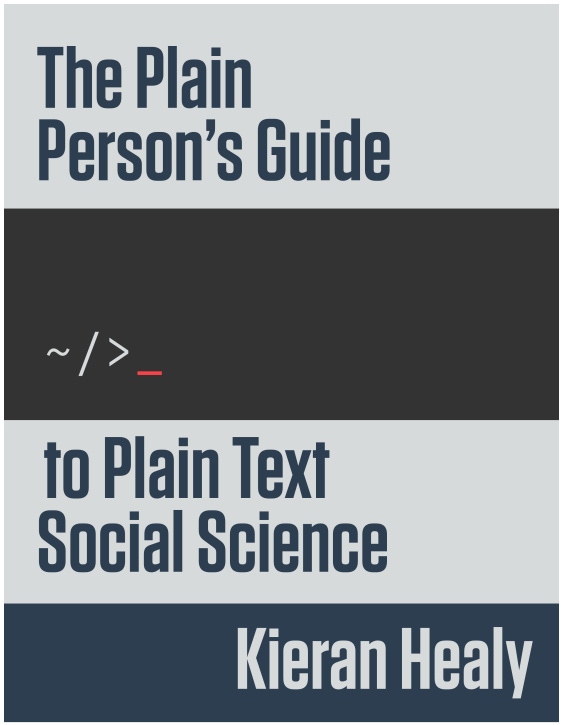
Two computer revolutions

The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system

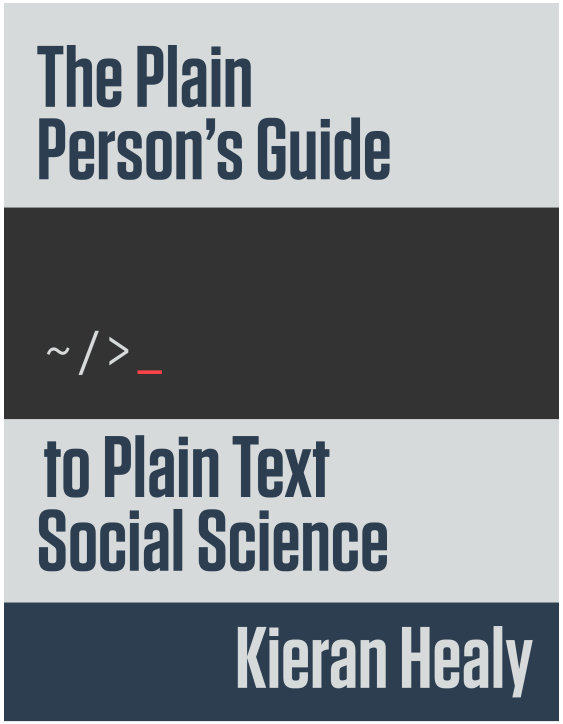
Where statistical computing lives

- Windows and pointers
- Multi-tsking, multiple windows
- Works heavily with the file system
- Underneath it's UNIX and the command line



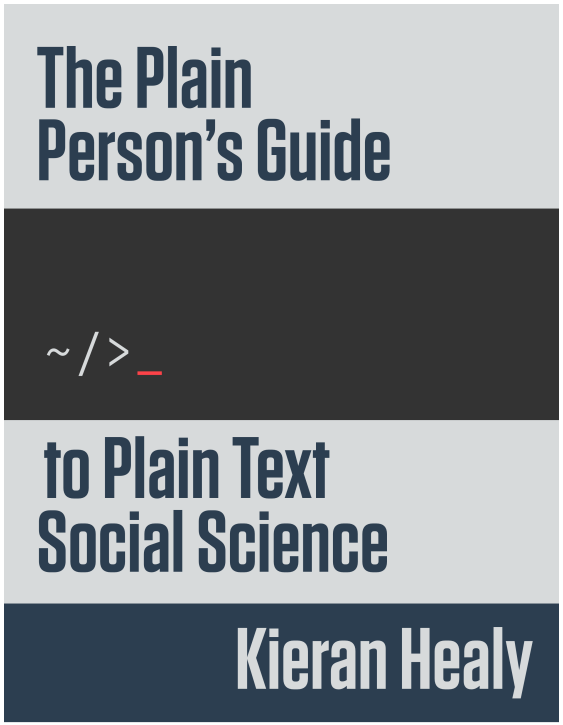
For more info, visit
https://plain-text.co/

- often free, open-sourced, and powerful



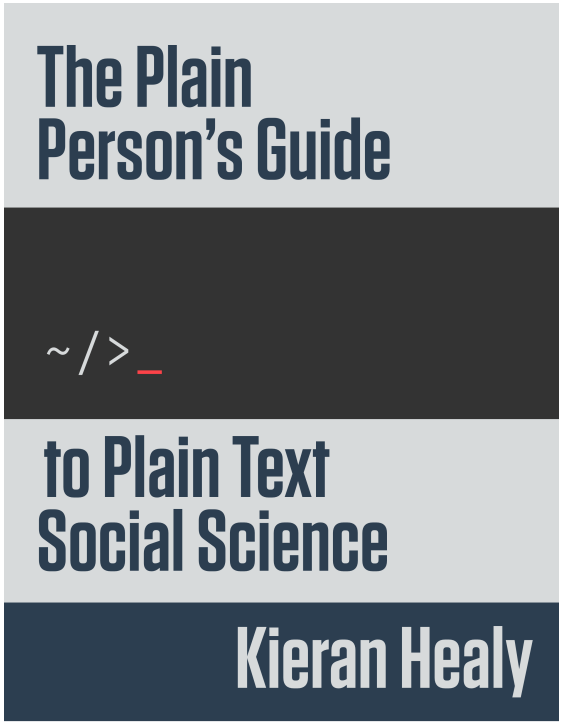
For more info, visit
https://plain-text.co/

- often free, open-sourced, and powerful
- Large, friendly communities around them



For more info, visit
https://plain-text.co/

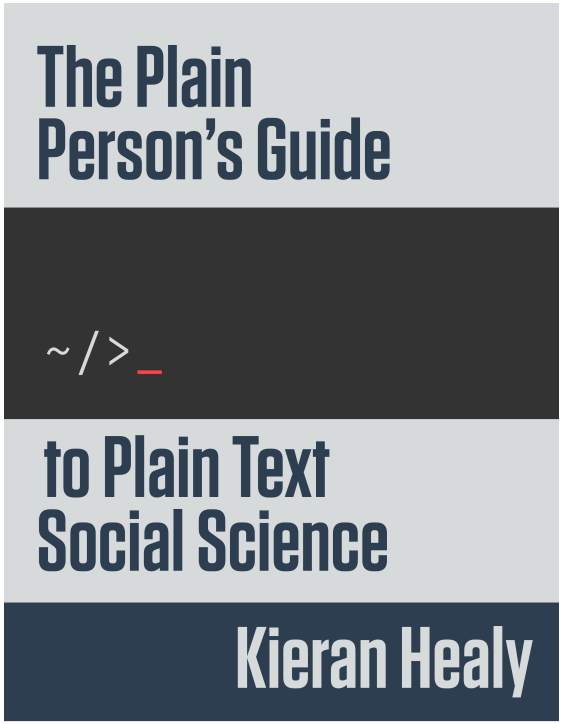
- often free, open-sourced, and powerful
- Large, friendly communities around them
- Tons of resources



For more info, visit
https://plain-text.co/

- often free, open-sourced, and powerful
- Large, friendly communities around them
- Tons of resources
- But... far from the touch-based paradigm of modern computing

Plain-text tools for data analysis



For more info, visit
https://plain-text.co/

- often free, open-sourced, and powerful
- Large, friendly communities around them
- Tons of resources
- But... far from the touch-based paradigm of modern computing
- So why use them?

The process of applied statistics/data science is intrinsically messy

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real

In the Engineering model

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)

In the Engineering model

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files

In the Engineering model

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files
- Final output is the file you are working on (e.g., Word doc or maybe converted to a PDF).

In the Engineering model

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files
- Final output is the file you are working on (e.g., Word doc or maybe converted to a PDF).

In the Engineering model

- Plain-text files are real

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files
- Final output is the file you are working on (e.g., Word doc or maybe converted to a PDF).

In the Engineering model

- Plain-text files are real
- Intermediate outputs are produced via code, often inside documents

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files
- Final output is the file you are working on (e.g., Word doc or maybe converted to a PDF).

In the Engineering model

- Plain-text files are real
- Intermediate outputs are produced via code, often inside documents
- Changes are tracked outside files

Office vs. engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real
- Intermediate outputs (copy/pasted into documents)
- Changes are tracked inside files
- Final output is the file you are working on (e.g., Word doc or maybe converted to a PDF).

In the Engineering model

- Plain-text files are real
- Intermediate outputs are produced via code, often inside documents
- Changes are tracked outside files
- Final outputs are assembled programatically and converted to desired output format

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - “Track Changes” is powerful and easy

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - “Track Changes” is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:
  - Plain text is universally portable

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:
  - Plain text is universally portable
  - Push button, recreate analysis

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:
  - Plain text is universally portable
  - Push button, recreate analysis
  - Why won't R just do what I want!

Pros and cons to each approach

- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:
  - Plain text is universally portable
  - Push button, recreate analysis
  - Why won't R just do what I want!
  - Version control is a pain

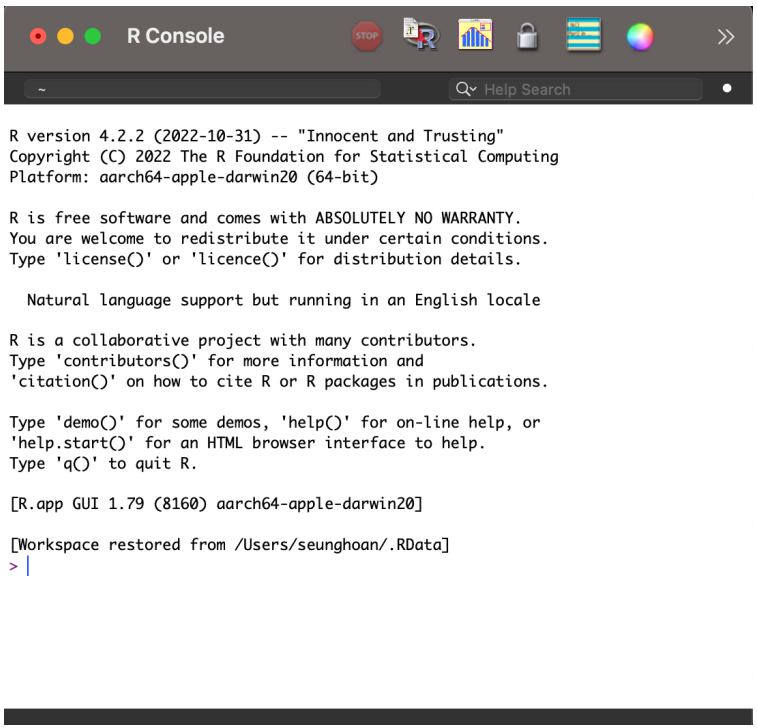
Pros and cons to each approach

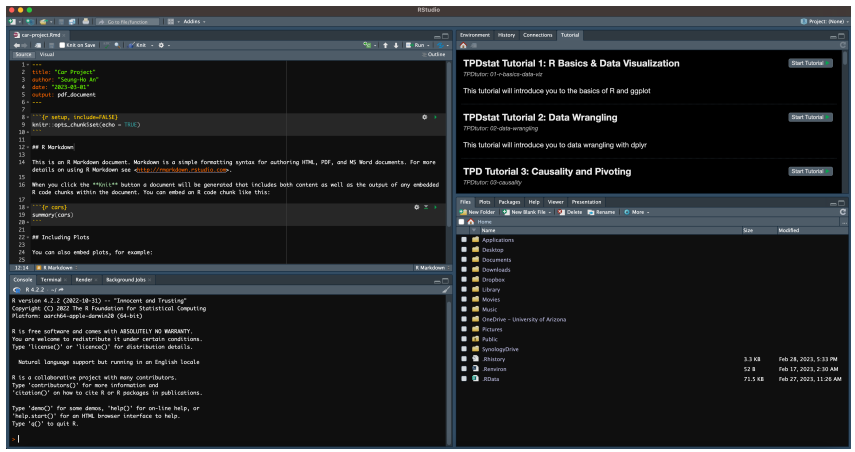
- Office model:
  - Everyone knows Word, Excel, Google Docs, etc.
  - "Track Changes" is powerful and easy
  - Wait, how did I make this figure and table?
  - Ah, I found the codes but, which one generated this figure and table?
  - An_final_final_real_final_lastedits_v57.docx
- Engineering model:
  - Plain text is universally portable
  - Push button, recreate analysis
  - Why won't R just do what I want!
  - Version control is a pain

We'll tend toward the Engineering model because it's better suited to keep the mess in check

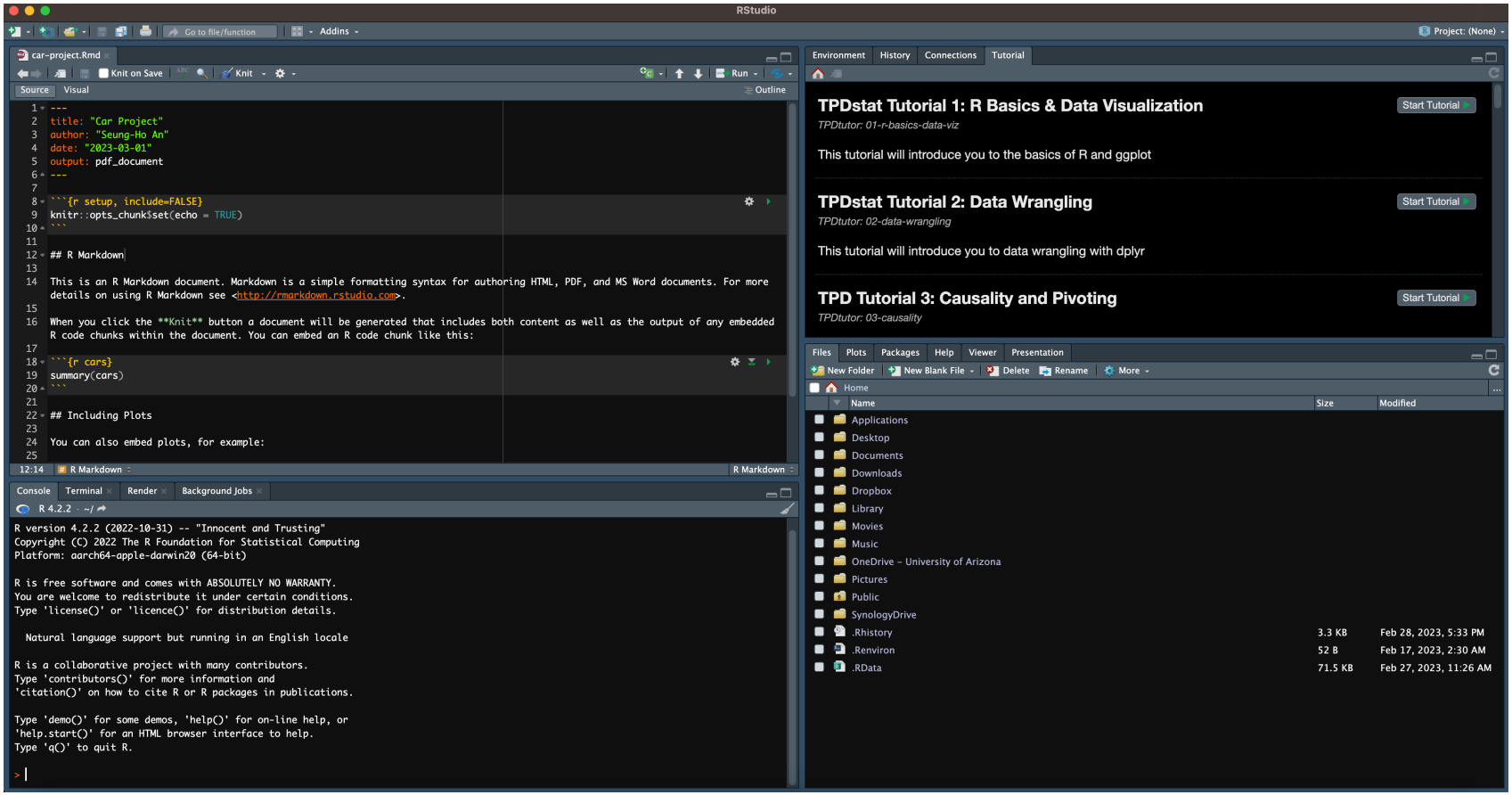
Let's take a tou**R**

R vs. RStudio

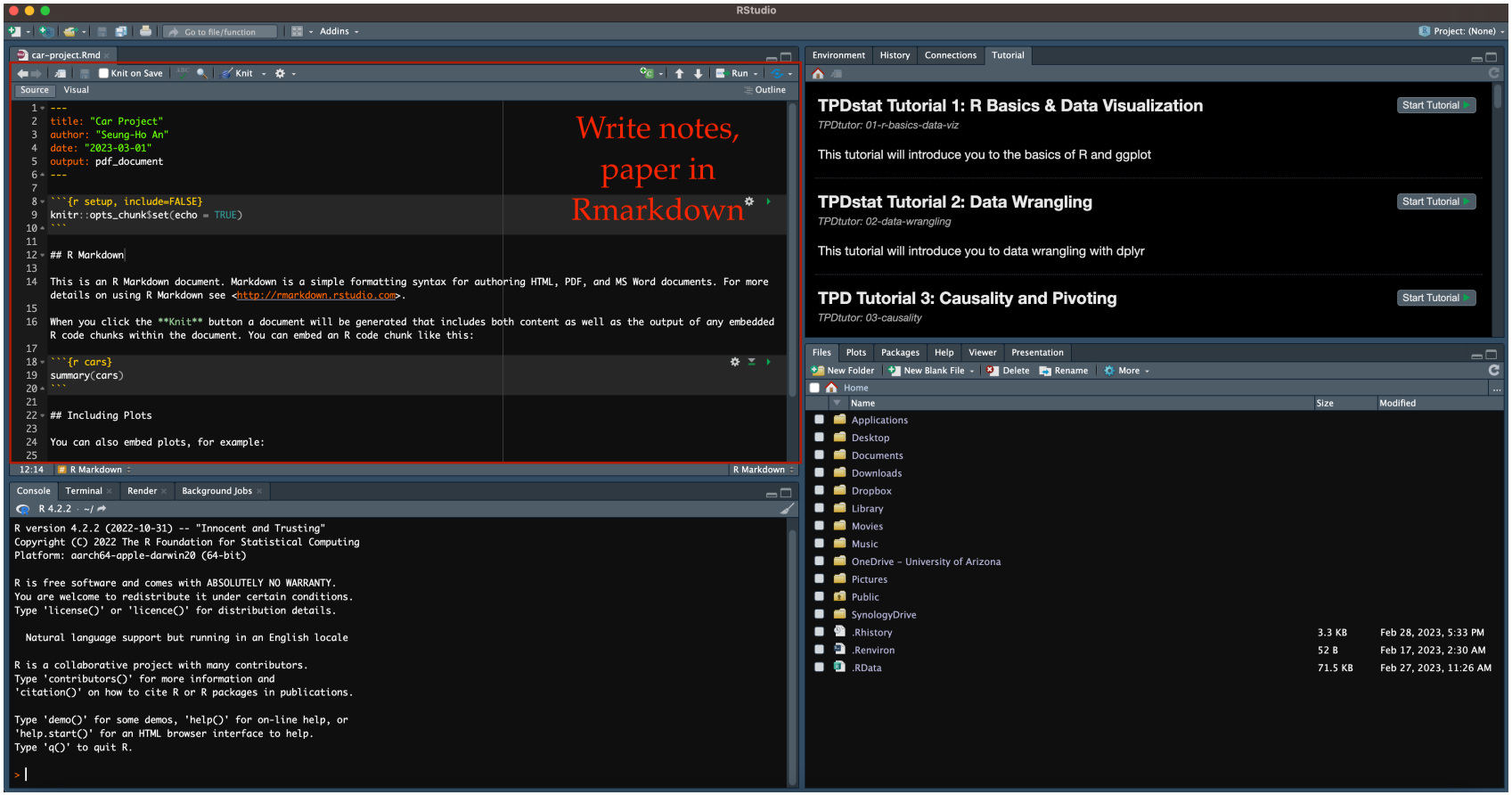

R Console
Help Search
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[R.app GUI 1.79 (8160) aarch64-apple-darwin20]
[Workspace restored from /Users/seunghoan/.RData]
>

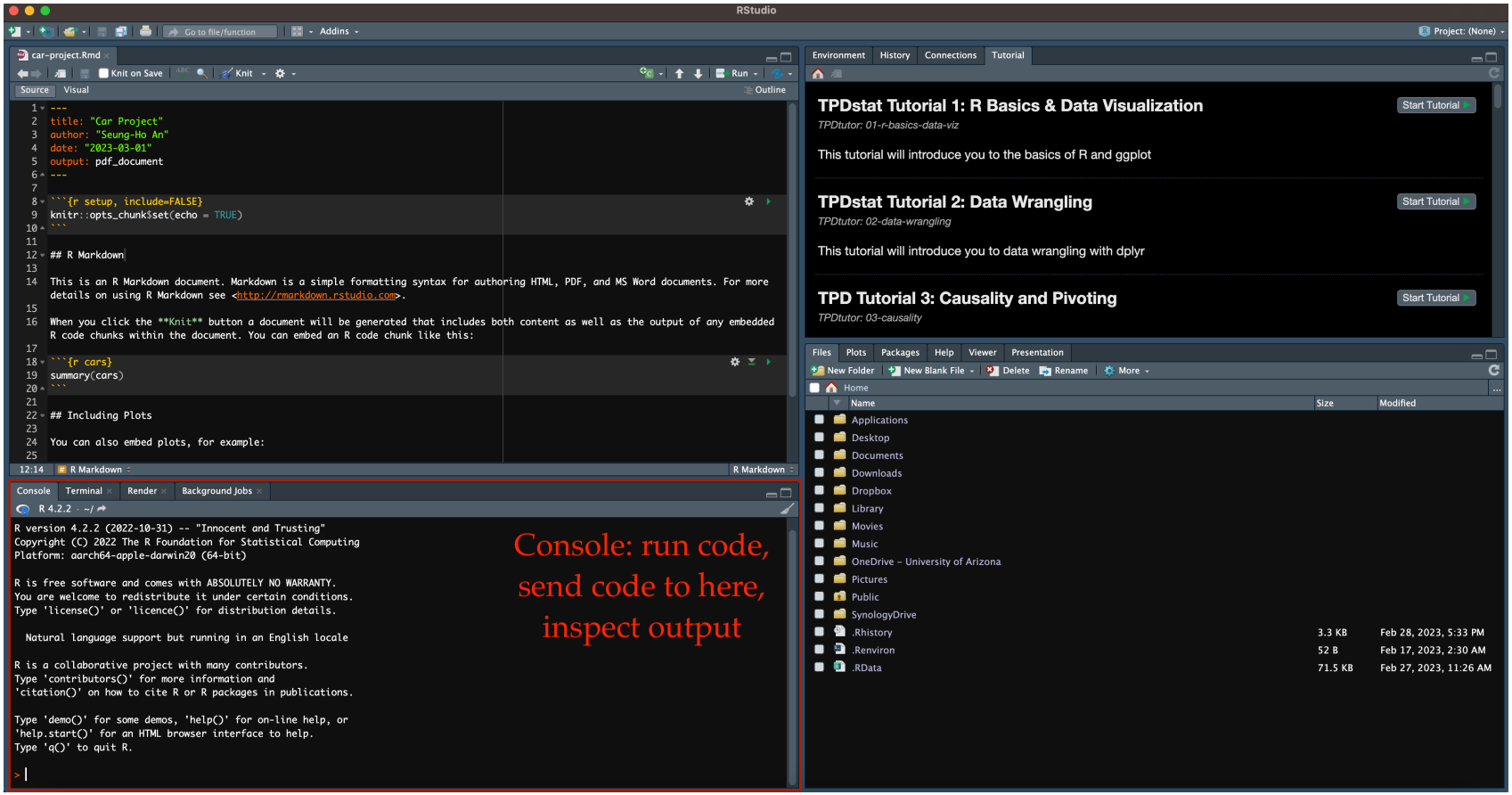

TPDstat Tutorial 1: R Basics & Data Visualization
This tutorial will introduce you to the basics of R and ggplot
TPDstat Tutorial 2: Data Wrangling
This tutorial will introduce you to data wrangling with dplyr
TPD Tutorial 3: Causality and Pivoting

R vs. RStudio (cont)

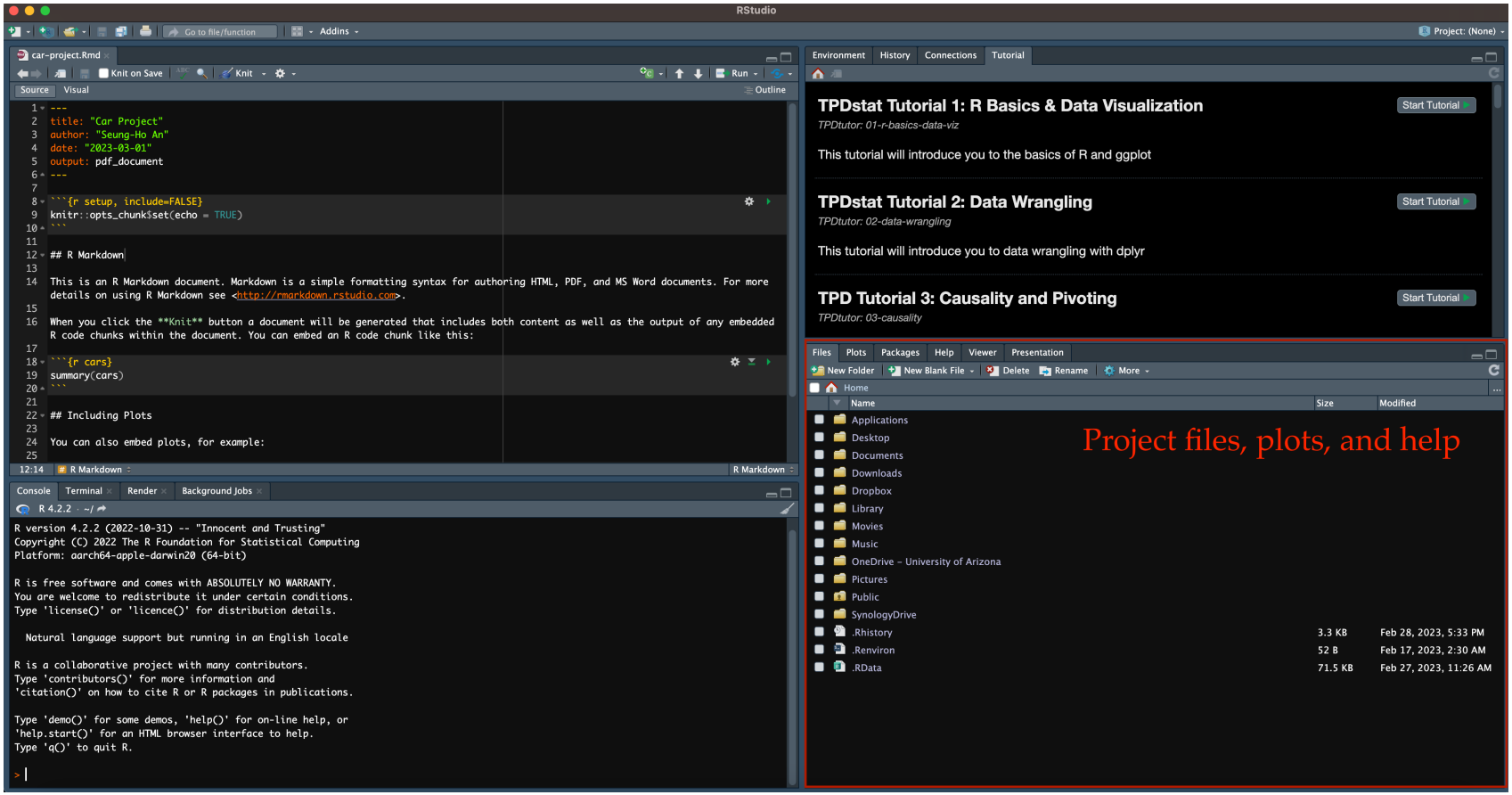

RStudio
Addins
Project: (None)
car-project.Rmd
Knit on Save
Knit
Run
Source
Visual
Outline
---
title: "Car Project"
author: "Seung-Ho An"
date: "2023-03-01"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Including Plots
You can also embed plots, for example:
12:14
R Markdown
R Markdown
Console
Terminal
Render
Background Jobs
R 4.2.2 · ~/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment
History
Connections
Tutorial
TPDstat Tutorial 1: R Basics & Data Visualization
Start Tutorial
TPDtutor: 01-r-basics-data-viz
This tutorial will introduce you to the basics of R and ggplot
TPDstat Tutorial 2: Data Wrangling
Start Tutorial
TPDtutor: 02-data-wrangling
This tutorial will introduce you to data wrangling with dplyr
TPD Tutorial 3: Causality and Pivoting
Start Tutorial
TPDtutor: 03-causality
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Delete
Rename
More
Home
Name
Size
Modified
Applications
Desktop
Documents
Downloads
Dropbox
Library
Movies
Music
OneDrive - University of Arizona
Pictures
Public
SynologyDrive
.Rhistory 3.3 KB Feb 28, 2023, 5:33 PM
.Renviron 52 B Feb 17, 2023, 2:30 AM
.RData 71.5 KB Feb 27, 2023, 11:26 AM

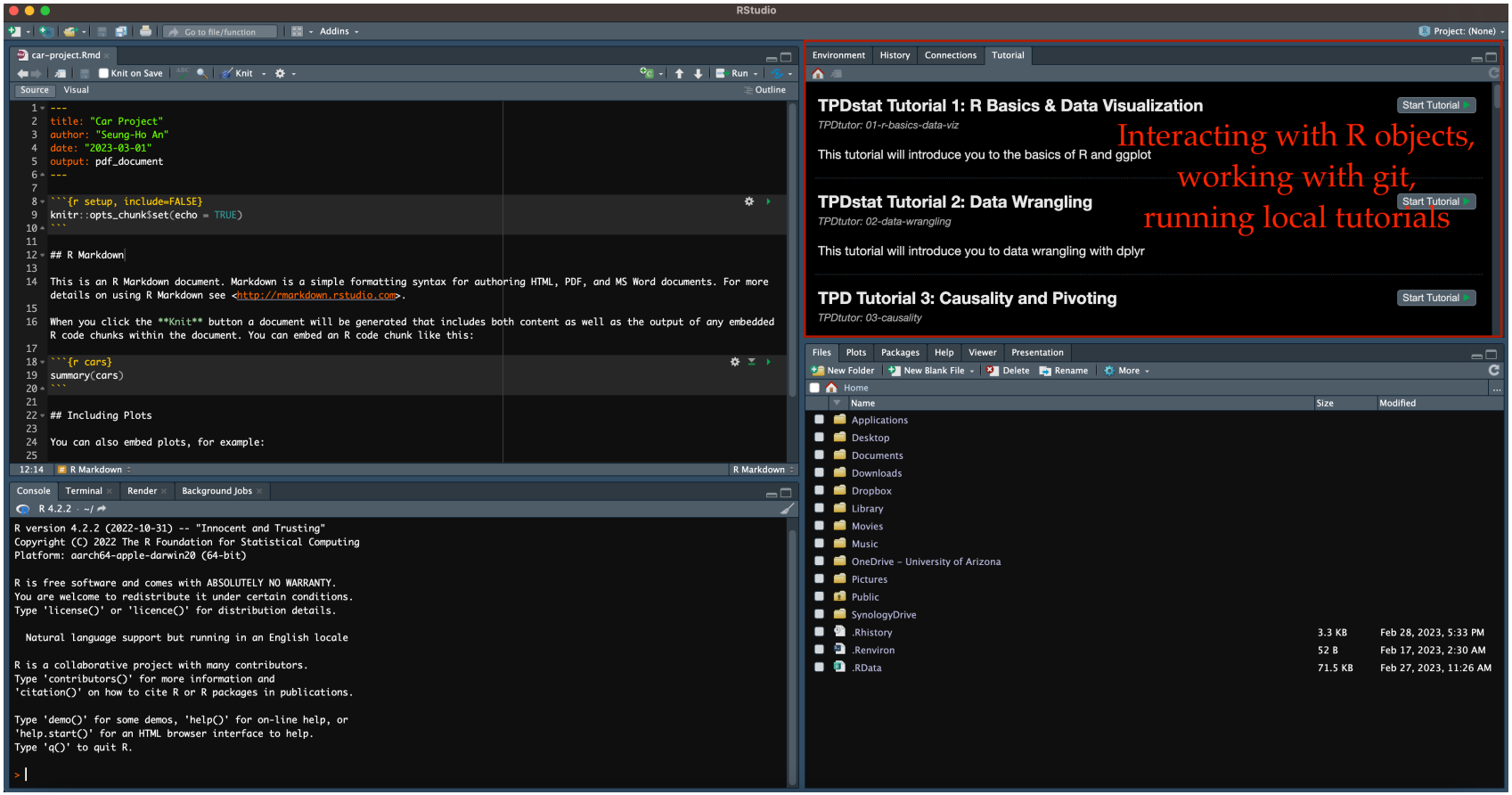

RStudio
Write notes, paper in Rmarkdown
TPDstat Tutorial 1: R Basics & Data Visualization
TPDtutor: 01-r-basics-data-viz
This tutorial will introduce you to the basics of R and ggplot
TPDstat Tutorial 2: Data Wrangling
TPDtutor: 02-data-wrangling
This tutorial will introduce you to data wrangling with dplyr
TPD Tutorial 3: Causality and Pivoting
TPDtutor: 03-causality


RStudio
car-project.Rmd
Knit on Save
Knit
Run
Source
Visual
Outline
---
title: "Car Project"
author: "Seung-Ho An"
date: "2023-03-01"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Including Plots
You can also embed plots, for example:
12:14
R Markdown
R Markdown
Console
Terminal
Render
Background Jobs
R 4.2.2 · ~/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Console: run code, send code to here, inspect output
Environment
History
Connections
Tutorial
TPDstat Tutorial 1: R Basics & Data Visualization
TPDtutor: 01-r-basics-data-viz
This tutorial will introduce you to the basics of R and ggplot
Start Tutorial
TPDstat Tutorial 2: Data Wrangling
TPDtutor: 02-data-wrangling
This tutorial will introduce you to data wrangling with dplyr
Start Tutorial
TPD Tutorial 3: Causality and Pivoting
TPDtutor: 03-causality
Start Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Delete
Rename
More
Home
Name
Size
Modified
Applications
Desktop
Documents
Downloads
Dropbox
Library
Movies
Music
OneDrive - University of Arizona
Pictures
Public
SynologyDrive
.Rhistory 3.3 KB Feb 28, 2023, 5:33 PM
.Renviron 52 B Feb 17, 2023, 2:30 AM
.RData 71.5 KB Feb 27, 2023, 11:26 AM


RStudio
car-project.Rmd
Knit on Save
Knit
Run
Source
Visual
Outline
---
title: "Car Project"
author: "Seung-Ho An"
date: "2023-03-01"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Including Plots
You can also embed plots, for example:
12:14
R Markdown
Console
Terminal
Render
Background Jobs
R 4.2.2 · ~/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment
History
Connections
Tutorial
TPDstat Tutorial 1: R Basics & Data Visualization
TPDtutor: 01-r-basics-data-viz
This tutorial will introduce you to the basics of R and ggplot
Start Tutorial
TPDstat Tutorial 2: Data Wrangling
TPDtutor: 02-data-wrangling
This tutorial will introduce you to data wrangling with dplyr
Start Tutorial
TPD Tutorial 3: Causality and Pivoting
TPDtutor: 03-causality
Start Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Delete
Rename
More
Home
Name
Size
Modified
Applications
Desktop
Documents
Downloads
Dropbox
Library
Movies
Music
OneDrive - University of Arizona
Pictures
Public
SynologyDrive
.Rhistory 3.3 KB Feb 28, 2023, 5:33 PM
.Renviron 52 B Feb 17, 2023, 2:30 AM
.RData 71.5 KB Feb 27, 2023, 11:26 AM
Project files, plots, and help


RStudio
car-project.Rmd
Knit on Save
Knit
Run
Source
Visual
Outline
---
title: "Car Project"
author: "Seung-Ho An"
date: "2023-03-01"
output: pdf_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Including Plots
You can also embed plots, for example:
12:14
R Markdown
Console
Terminal
Render
Background Jobs
R 4.2.2 · ~/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Addins
Project: (None)
Environment
History
Connections
Tutorial
TPDstat Tutorial 1: R Basics & Data Visualization
TPDtutor: 01-r-basics-data-viz
This tutorial will introduce you to the basics of R and ggplot
Start Tutorial
TPDstat Tutorial 2: Data Wrangling
TPDtutor: 02-data-wrangling
This tutorial will introduce you to data wrangling with dplyr
Start Tutorial
TPD Tutorial 3: Causality and Pivoting
TPDtutor: 03-causality
Start Tutorial
Interacting with R objects,
working with git,
running local tutorials
Files
Plots
Packages
Help
Viewer
Presentation
New Folder
New Blank File
Delete
Rename
More
Home
Name
Size
Modified
Applications
Desktop
Documents
Downloads
Dropbox
Library
Movies
Music
OneDrive - University of Arizona
Pictures
Public
SynologyDrive
.Rhistory 3.3 KB Feb 28, 2023, 5:33 PM
.Renviron 52 B Feb 17, 2023, 2:30 AM
.RData 71.5 KB Feb 27, 2023, 11:26 AM

3. Using RMarkdown

```
1  library(ggplot2)
2  ggplot(mtcars, aes(x = wt, y = mpg)) +
3    geom_point()
```

Figure 2: Writing codes

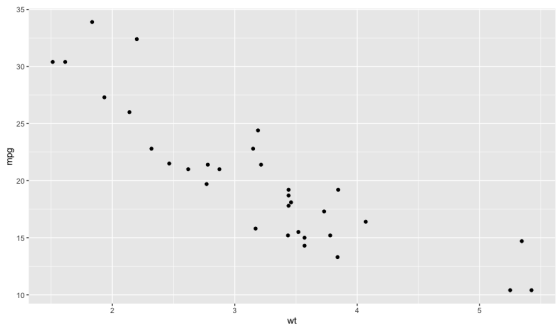


Figure 3: Looking at output

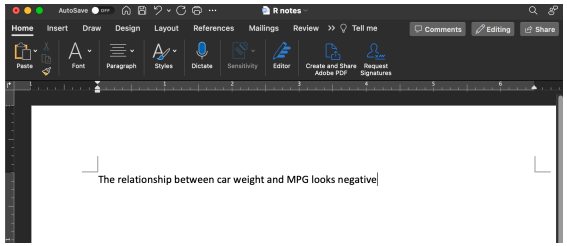


Figure 4: Taking notes

Rmarkdown files to the rescue

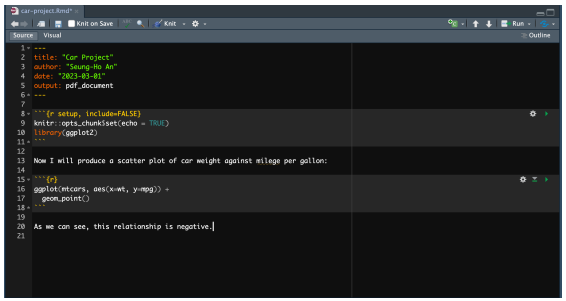


Figure 5: Rmarkdown file

Keep code and notes together in plain text

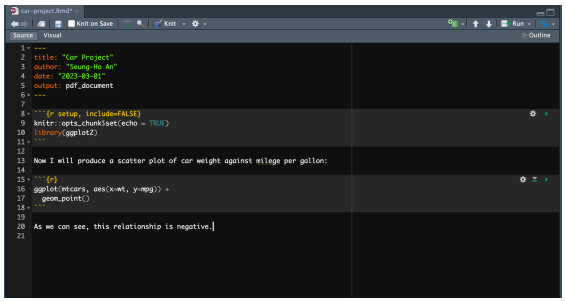


Figure 5: Rmarkdown file

Keep code and notes together in plain text

Figure 6: Knit in R

Rmarkdown files to the rescue

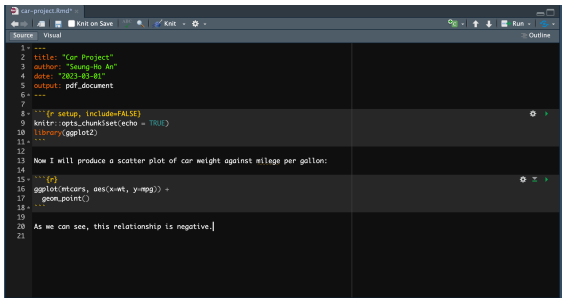


Figure 5: Rmarkdown file

Keep code and notes together in plain text

Figure 6: Knit in R

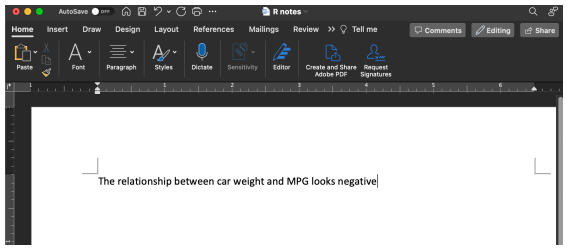


Figure 7: PDF output

Produce nice-looking outputs in different formats

Markdown: formatting in plain text

Non-code text in Rmd files is plain text with formatting instructions

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1

Header 2

Header 3

Header 4

Header 5

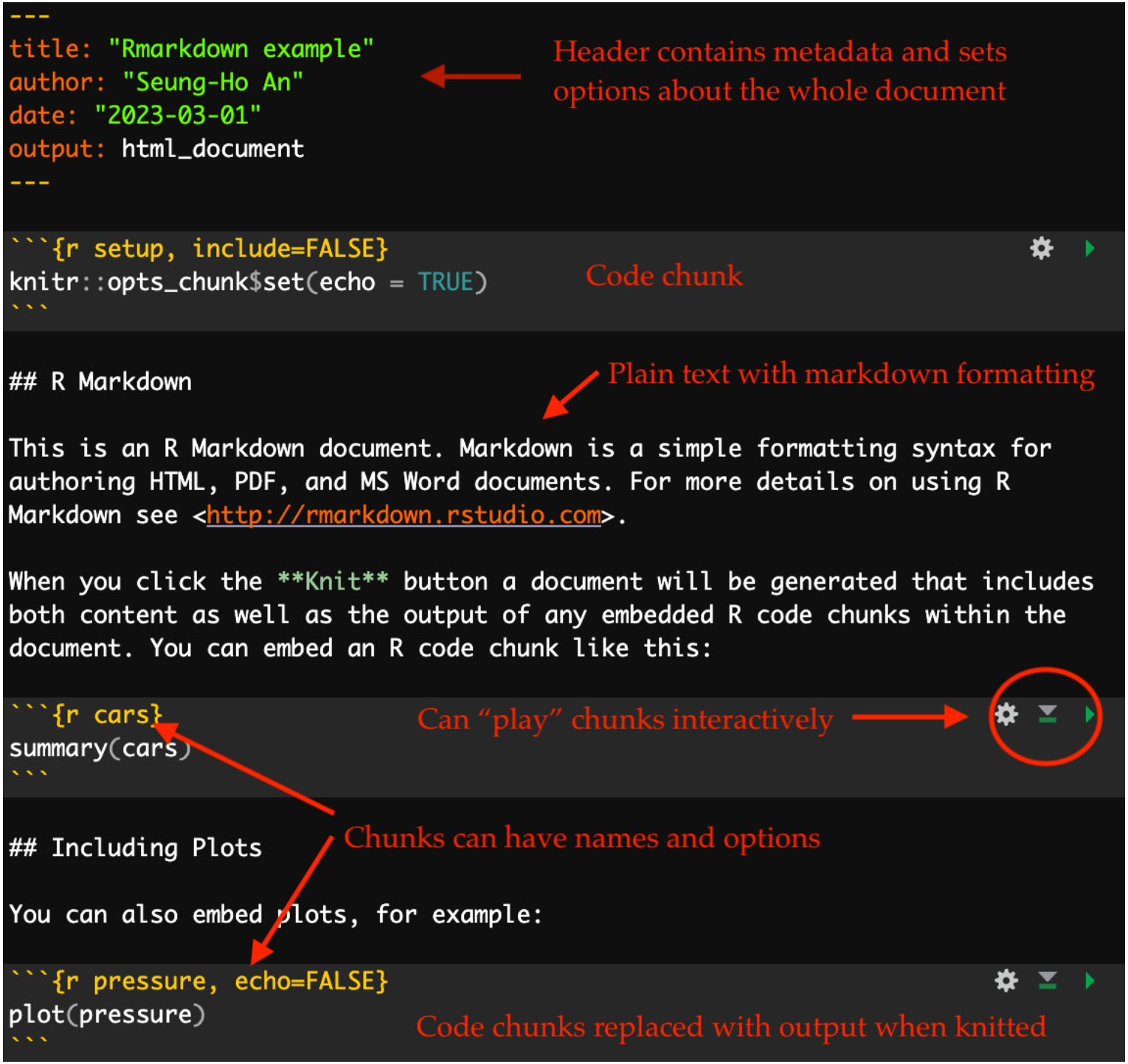
Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image: 
horizontal rule (or slide break):

> block quote

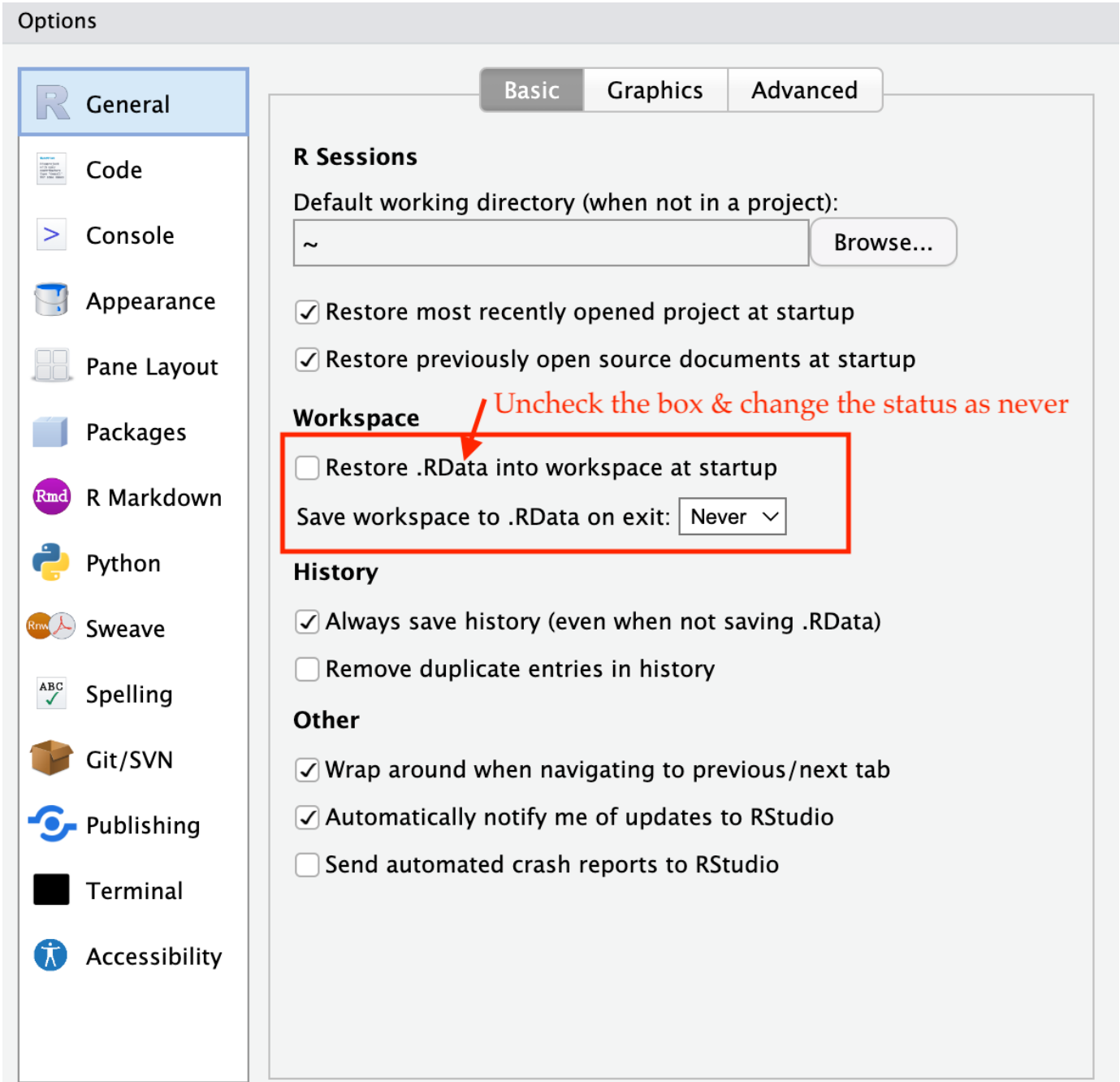
- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2


---
title: "Rmarkdown example"
author: "Seung-Ho An"
date: "2023-03-01"
output: html_document

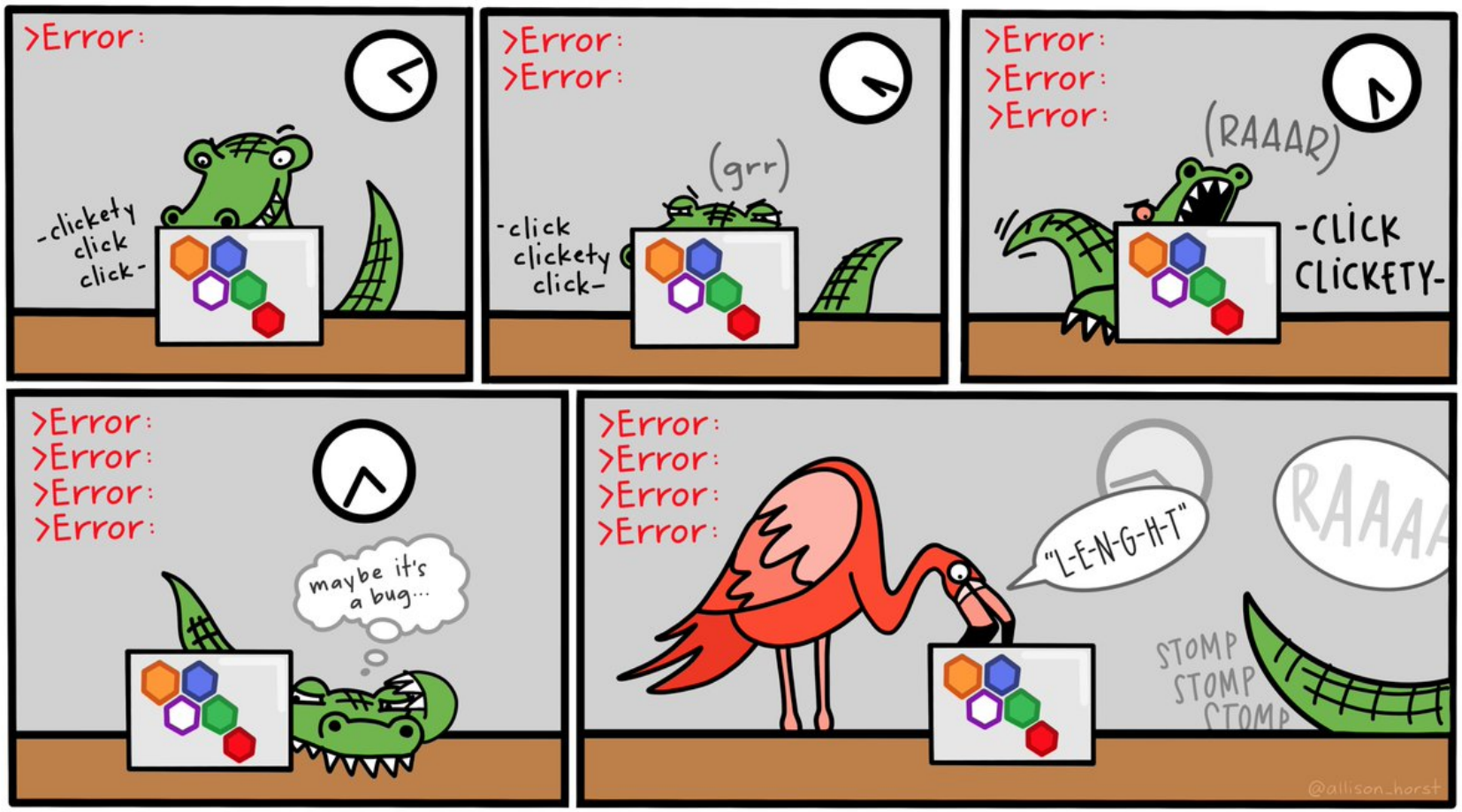
Header contains metadata and sets options about the whole document
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
Code chunk
R Markdown
Plain text with markdown formatting
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Can "play" chunks interactively
Including Plots
Chunks can have names and options
You can also embed plots, for example:
```{r pressure, echo=FALSE}
plot(pressure)
```
Code chunks replaced with output when knitted

Remember what's real



Try to type your code by hand

Typing speeds up the try-fail cycle



Physically typing the code is best way to familiarize yourself with R and the try-fail-try-fail-try-succeed cycle

Code that you can type and run:

```
## Any R code that begins with the # character is a comment
## Comments are ignored by R

my_numbers <- c(4, 8, 15, 16, 23, 42) # Anything after # is also a comment
```

Code that you can type and run:

```
## Any R code that begins with the # character is a comment
## Comments are ignored by R

my_numbers <- c(4, 8, 15, 16, 23, 42) # Anything after # is also a comment
```

Output from code prefixed by ## by convention:

```
my_numbers
```

```
## [1]  4  8 15 16 23 42
```

Code that you can type and run:

```
## Any R code that begins with the # character is a comment
## Comments are ignored by R

my_numbers <- c(4, 8, 15, 16, 23, 42) # Anything after # is also a comment
```

Output from code prefixed by ## by convention:

```
my_numbers
```

```
## [1]  4  8 15 16 23 42
```

Output also has a counter in brackets when over one line:

```
options(width = 45)
letters
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
## [11] "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [21] "u" "v" "w" "x" "y" "z"
```

Everything in R has a name

```
my_numbers # just created this
```

```
## [1]  4  8 15 16 23 42
```

```
letters # this is build into R
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
## [11] "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [21] "u" "v" "w" "x" "y" "z"
```

```
pi # also built in
```

```
## [1] 3.141593
```

Everything in R has a name

```
my_numbers # just created this
```

```
## [1]  4  8 15 16 23 42
```

```
letters # this is build into R
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
## [11] "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [21] "u" "v" "w" "x" "y" "z"
```

```
pi # also built in
```

```
## [1] 3.141593
```

Some names are forbidden (NA, TRUE, FALSE, etc) or strongly not recommended (c, mean, table)

We do things in R with functions

Functions take in objects, perform actions, and return outputs:

```
mean(x = my_numbers)
```

```
## [1] 18
```

- x is the argument name,

We do things in R with functions

Functions take in objects, perform actions, and return outputs:

```
mean(x = my_numbers)
```

```
## [1] 18
```

- x is the argument name,
- my_numbers is what we're passing to that argument

We do things in R with functions

Functions take in objects, perform actions, and return outputs:

```
mean(x = my_numbers)
```

```
## [1] 18
```

- x is the argument name,
- `my_numbers` is what we're passing to that argument

If you omit the argument name, R will assume the default argument order:

```
mean(my_numbers)
```

```
## [1] 18
```

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

- Sometimes inscrutable, so look elsewhere:

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

- Sometimes inscrutable, so look elsewhere:
  - Google, StackOverflow, Twitter, RStudio Community

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

- Sometimes inscrutable, so look elsewhere:
  - Google, StackOverflow, Twitter, RStudio Community
  - Ask on the Teams Chat

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

- Sometimes inscrutable, so look elsewhere:
  - Google, StackOverflow, Twitter, RStudio Community
  - Ask on the Teams Chat
- Get help **early** before becoming too frustrated!

How do we know the default argument order? Look to help files:

```
help(mean)
?mean #shorter
```

- Sometimes inscrutable, so look elsewhere:
  - Google, StackOverflow, Twitter, RStudio Community
  - Ask on the Teams Chat
- Get help **early** before becoming too frustrated! -Easy to overlook small issues like missing commas, typos, etc.

Functions live in packages

Packages are bundles of functions written by other uses that we can use

Functions live in packages

Packages are bundles of functions written by other uses that we can use

Install packages using `install.packages()` to have them on your machine:

```
install.packages("ggplot2")
```

Functions live in packages

Packages are bundles of functions written by other uses that we can use

Install packages using `install.packages()` to have them on your machine:

```
install.packages("ggplot2")
```

Load them into your R session with `library()`:

```
library(ggplot2)
```

Functions live in packages

Packages are bundles of functions written by other uses that we can use

Install packages using `install.packages()` to have them on your machine:

```
install.packages("ggplot2")
```

Load them into your R session with `library()`:

```
library(ggplot2)
```

Now we can use any function provided by ggplot2

We can also use the `mypackage::` prefix to access package functions without loading:

```
knitr::kable(head(mtcars))
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

R tutorials & course packages

```
# installing course packages
my_packages <- c("tidyverse", "usethis", "devtools", "learnr",
                 "tinytex", "gitcreds", "broom")
install.packages(my_packages, repos = "http://cran.rstudio.com")

remotes::install_github("kosukeimai/qss-package", build_vignettes = TRUE)

# installing R tutorials locally
remotes::install_github("rstudio/learnr")

remotes::install_github("rstudio-education/gradethis")

remotes::install_github("seungho-an/TPDtutor")

# installing tiny tex
tinytex::install_tinytex()
```

4.

Data visualizations

4.1

Building plots by layers

Midwest data

```
midwest
```

```
## # A tibble: 437 x 28
##      PID county   state  area popto~1 popde~2
##    <int> <chr>    <chr> <dbl>   <int>   <dbl>
##  1   561 ADAMS    IL    0.052   66090   1271.
##  2   562 ALEXAND~ IL    0.014   10626    759 
##  3   563 BOND     IL    0.022   14991    681.
##  4   564 BOONE    IL    0.017   30806   1812.
##  5   565 BROWN    IL    0.018    5836    324.
##  6   566 BUREAU   IL    0.05    35688    714.
##  7   567 CALHOUN  IL    0.017    5322    313.
##  8   568 CARROLL  IL    0.027   16805    622.
##  9   569 CASS     IL    0.024   13437    560.
## 10   570 CHAMPAI~ IL    0.058  173025   2983.
## # ... with 427 more rows, 22 more variables:
## #   popwhite <int>, popblack <int>,
## #   popamerindian <int>, popasian <int>,
## #   popother <int>, percwhite <dbl>,
## #   percblack <dbl>, percamerindan <dbl>,
## #   percasian <dbl>, percother <dbl>,
## #   popadults <int>, perchsd <dbl>, ...
```

Create ggplot object and direct it to the correct data:

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

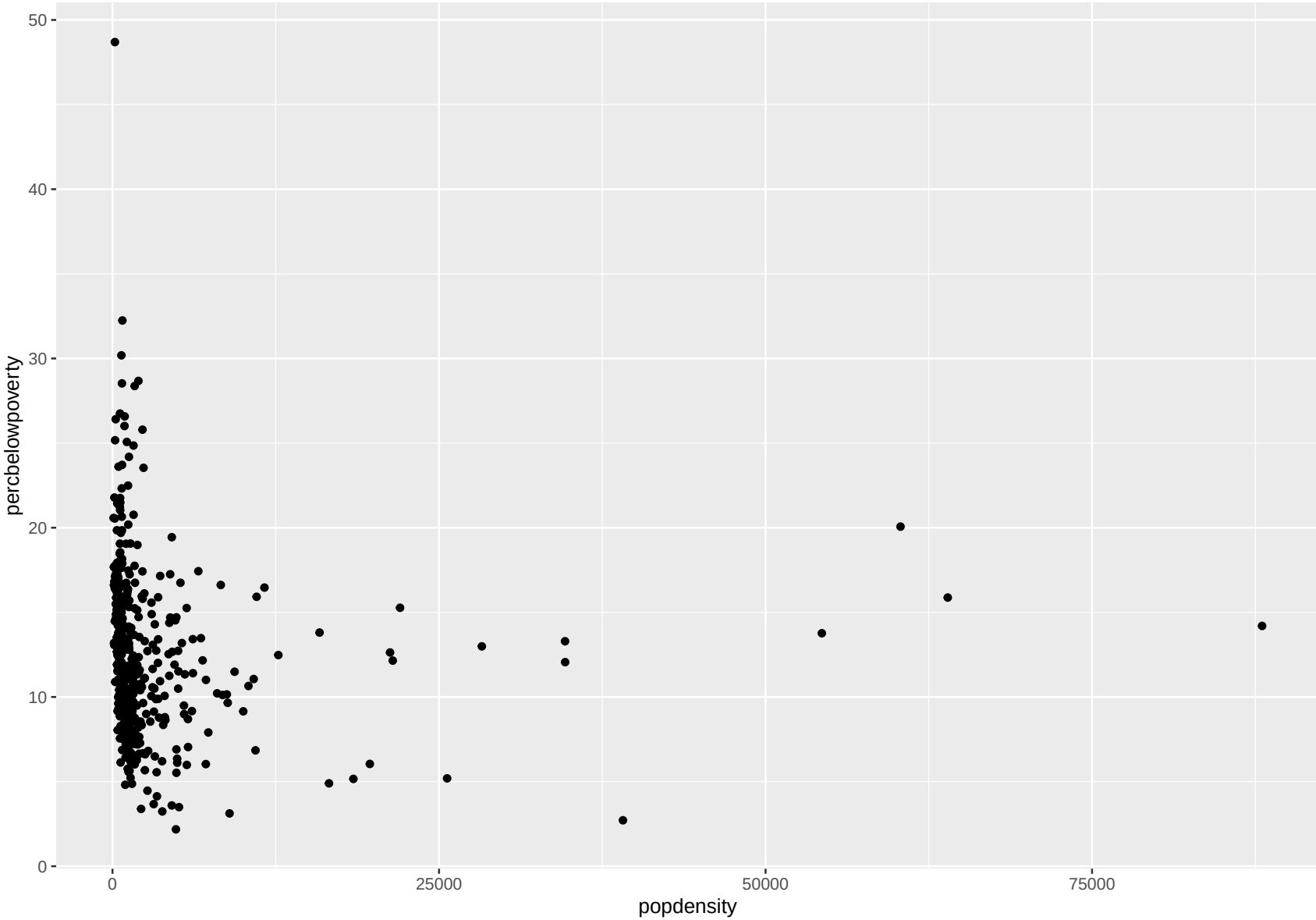
Mapping: tell ggplot what visual aesthetics correspond to which variables

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

Other aesthetic mappings: `color`, `shape`, `size`, `etc.`

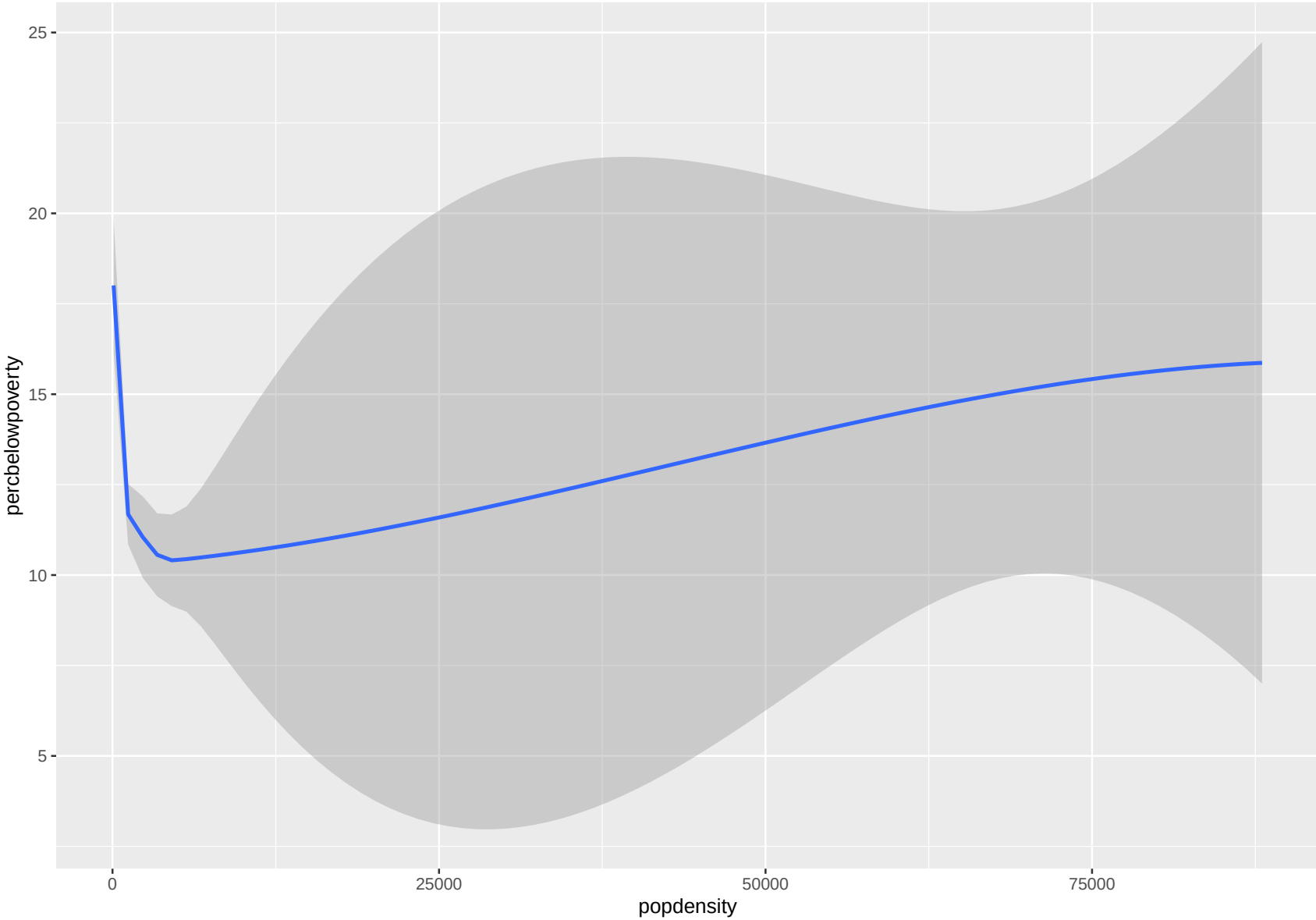
Adding a geom layer

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


50
40
30
20
10
0
percbelowpoverty
0
25000
50000
75000
popdensity

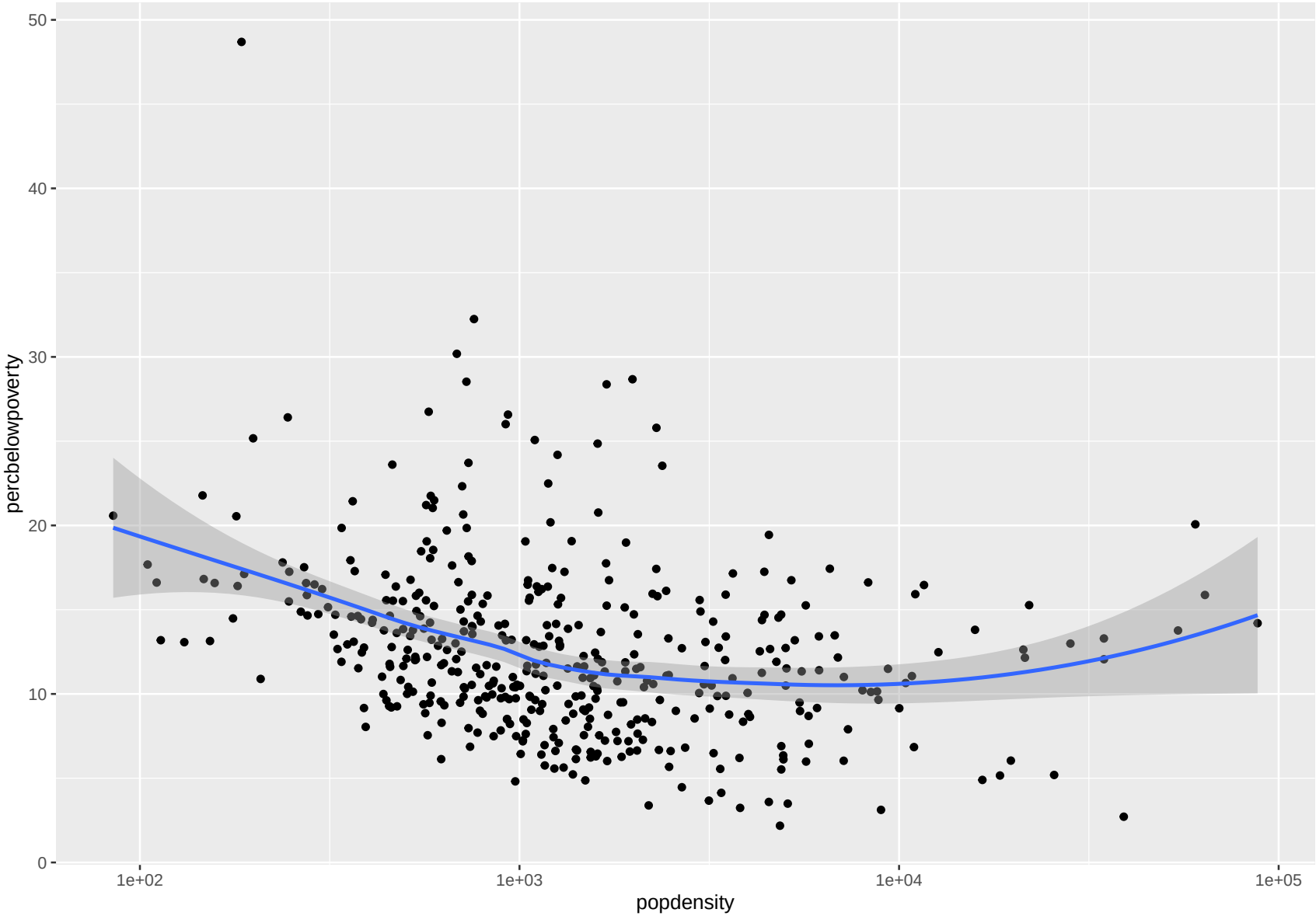
Trying a new geom

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


25
20
15
10
5
percbelowpoverty
0
25000
50000
75000
popdensity

Layering geoms is additive

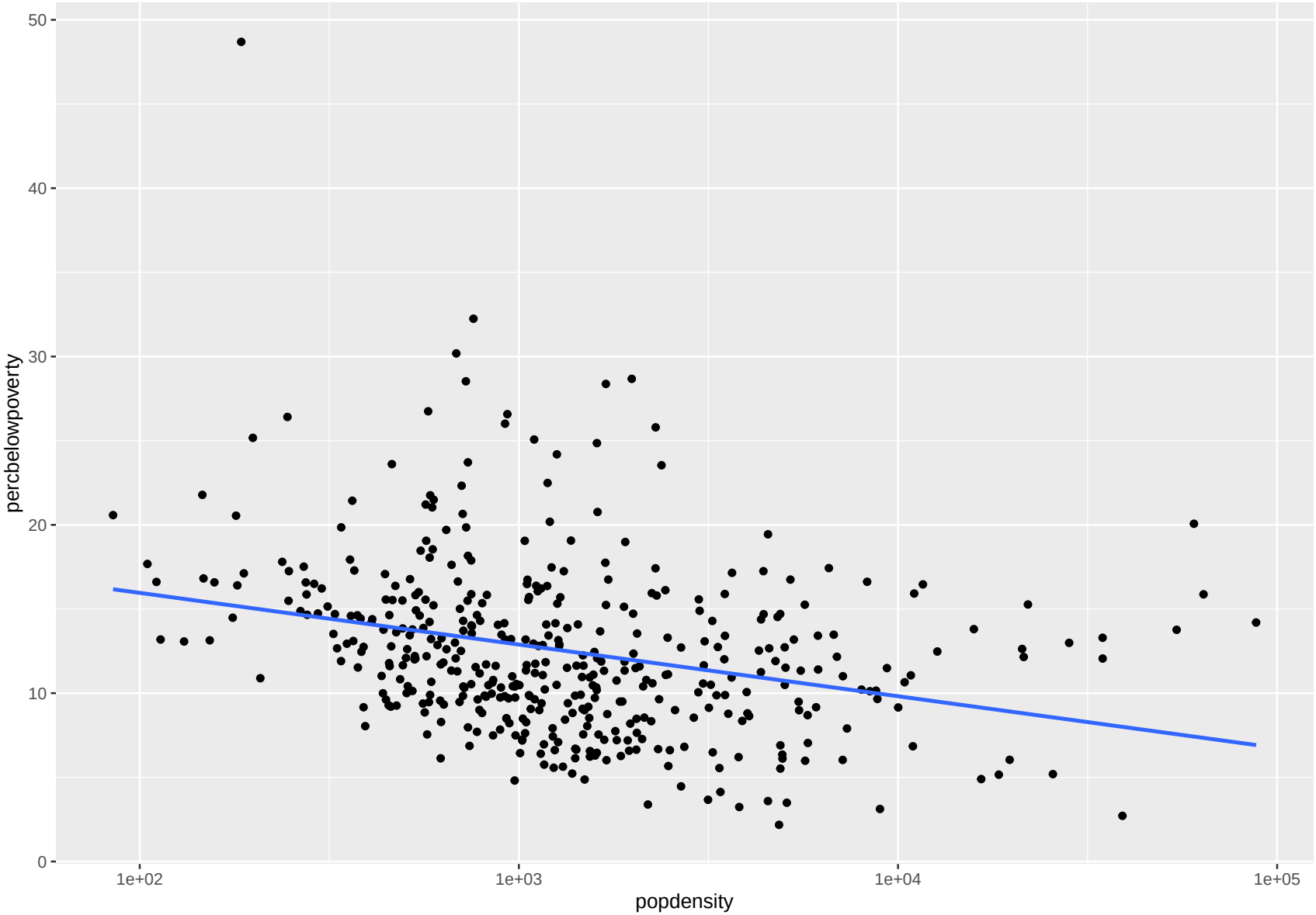
```
ggplot(data = midwest,
               mapping = aes(x = popdensity,
                             y = percbelowpoverty)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10()
```


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

Geoms can take arguments:

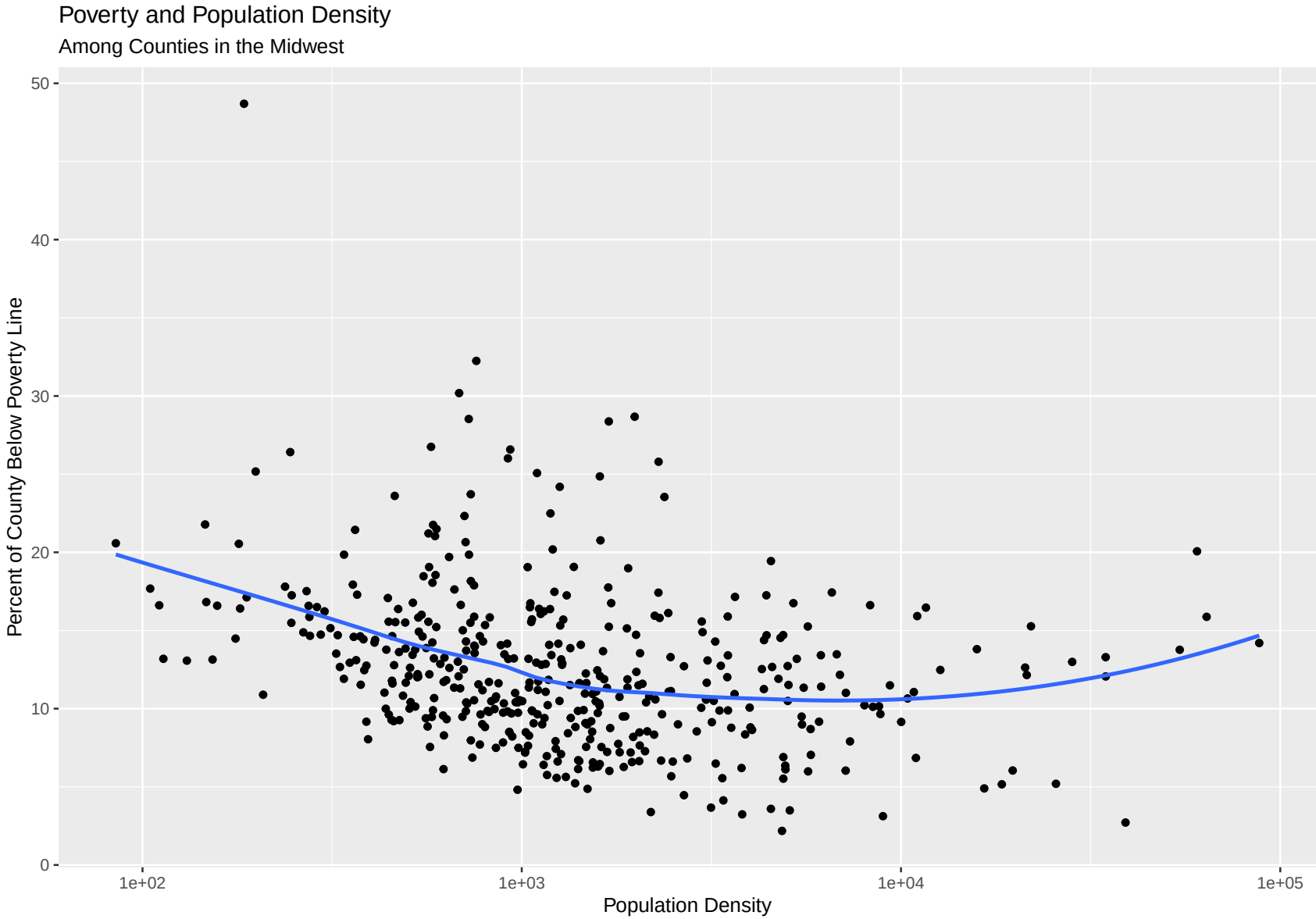
```
ggplot(data = midwest,
               mapping = aes(x = popdensity,
                             y = percbelowpoverty)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_log10()
```

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

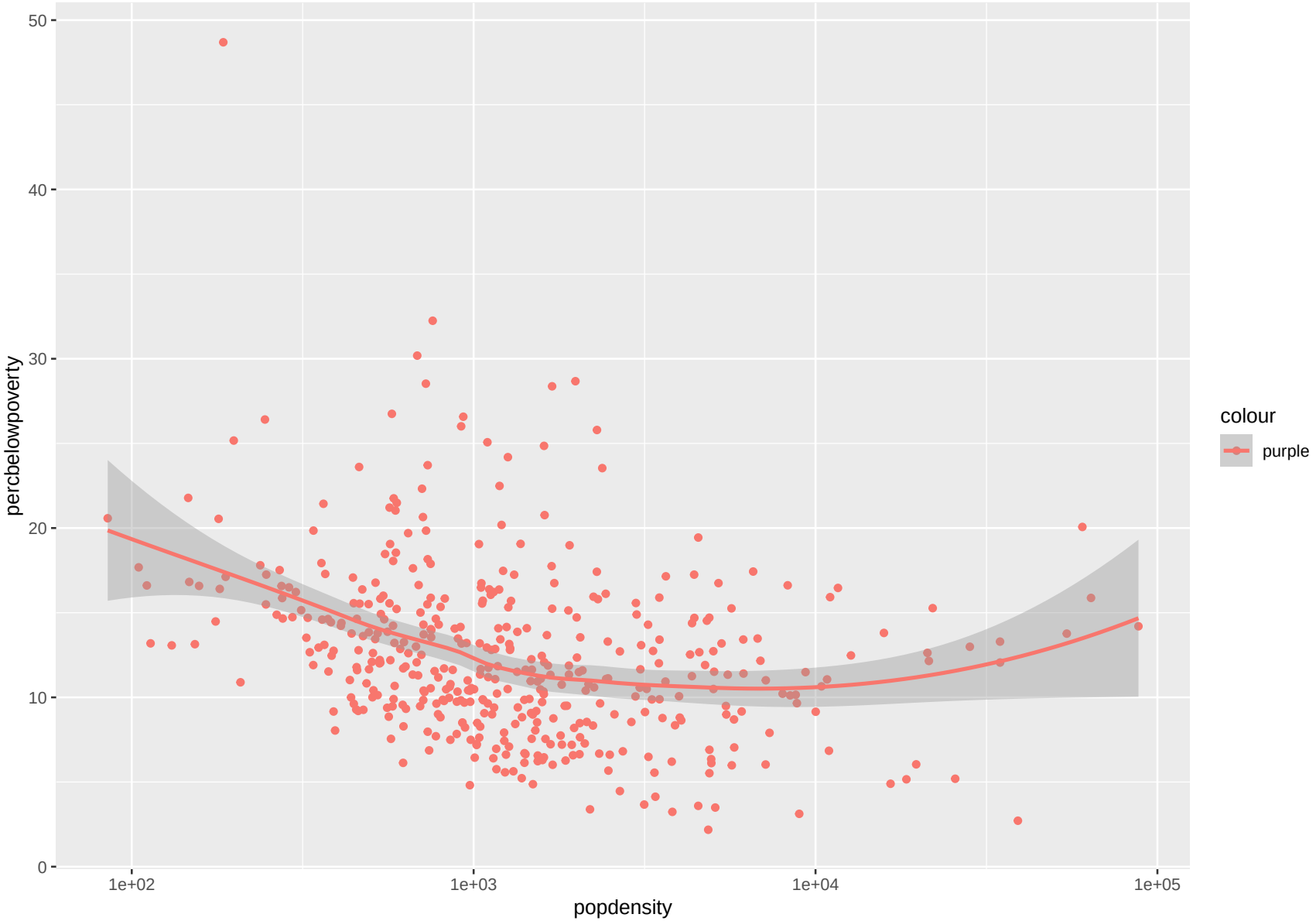
```
ggplot(data = midwest,
       mapping = aes(x = popdensity,
                     y = percbelowpoverty)) +
geom_point() +
geom_smooth(method = "loess", se = FALSE) + scale_x_log10()
labs(x = "Population Density",
       y = "Percent of County Below Poverty Line",
       title = "Poverty and Population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

Mapping vs. setting asthetics

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

Wait what?


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

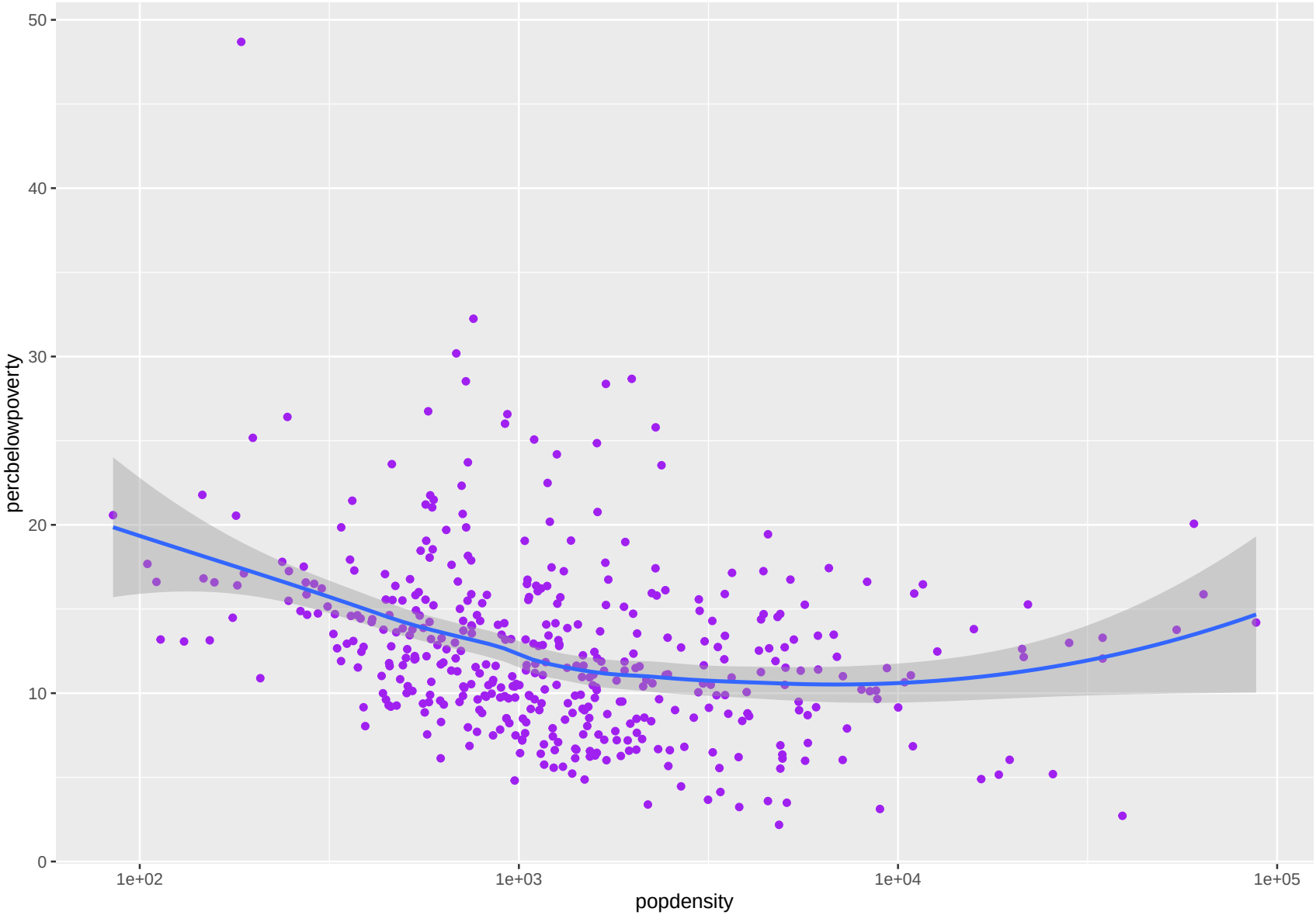
Mapping always refers to variables

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)

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

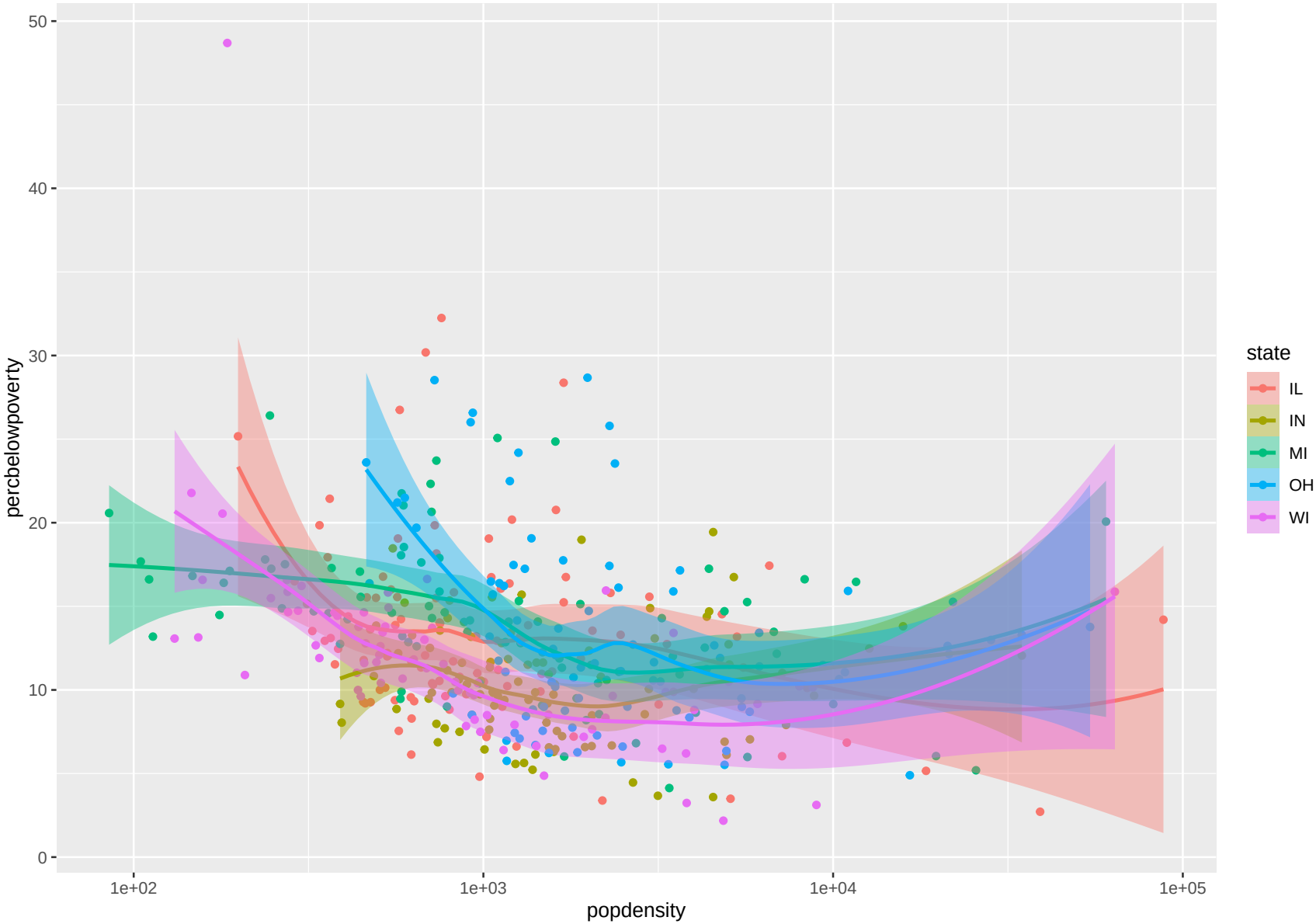
Set the color outside the mapping = aes() format

```
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

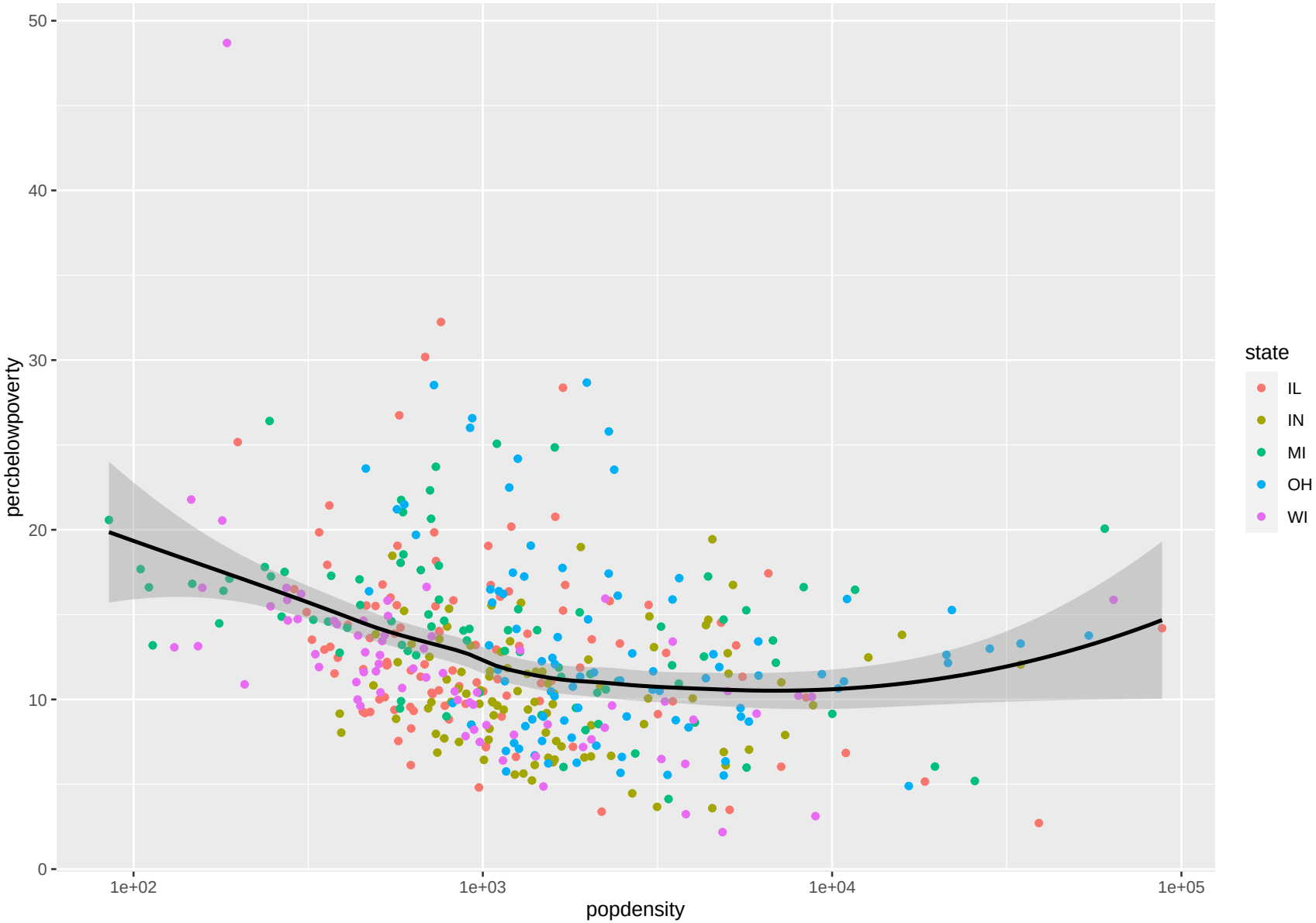
Mapping more aesthetics

```
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

Mappings 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

4.2

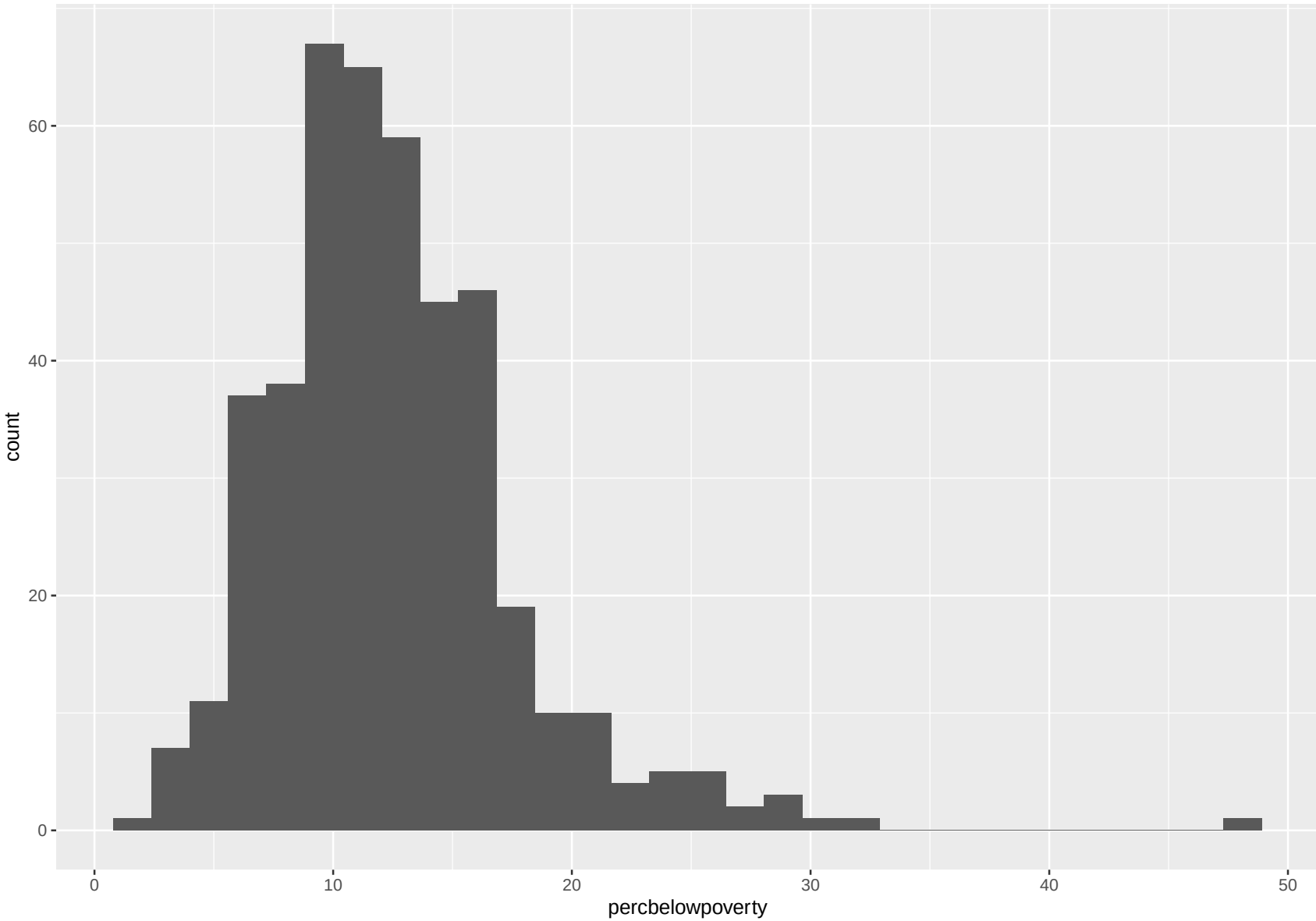
Histograms and boxplots

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

```
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

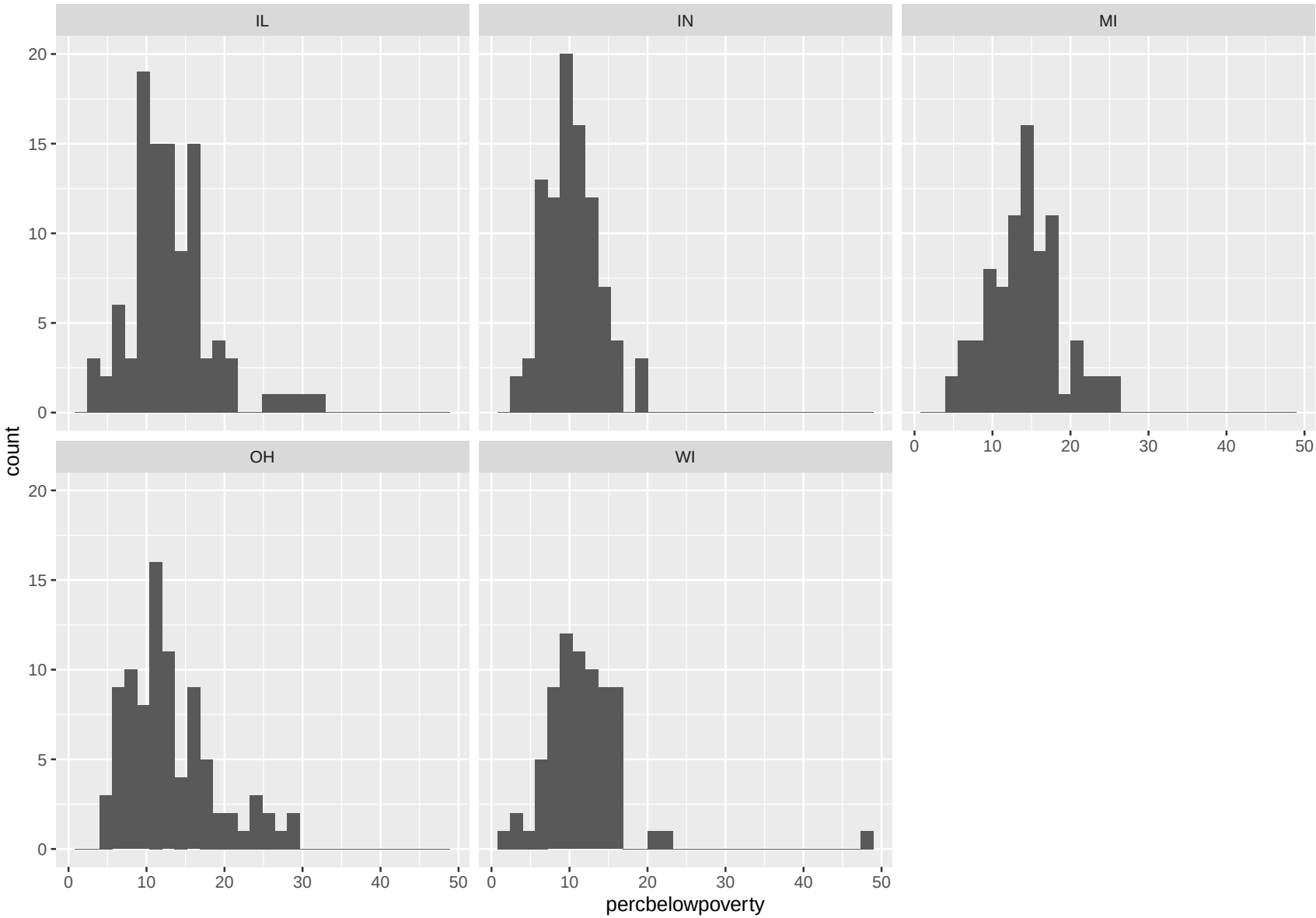

count
60
40
20
0
0
10
20
30
40
50
percbelowpoverty

Creating small multiples with facets

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

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

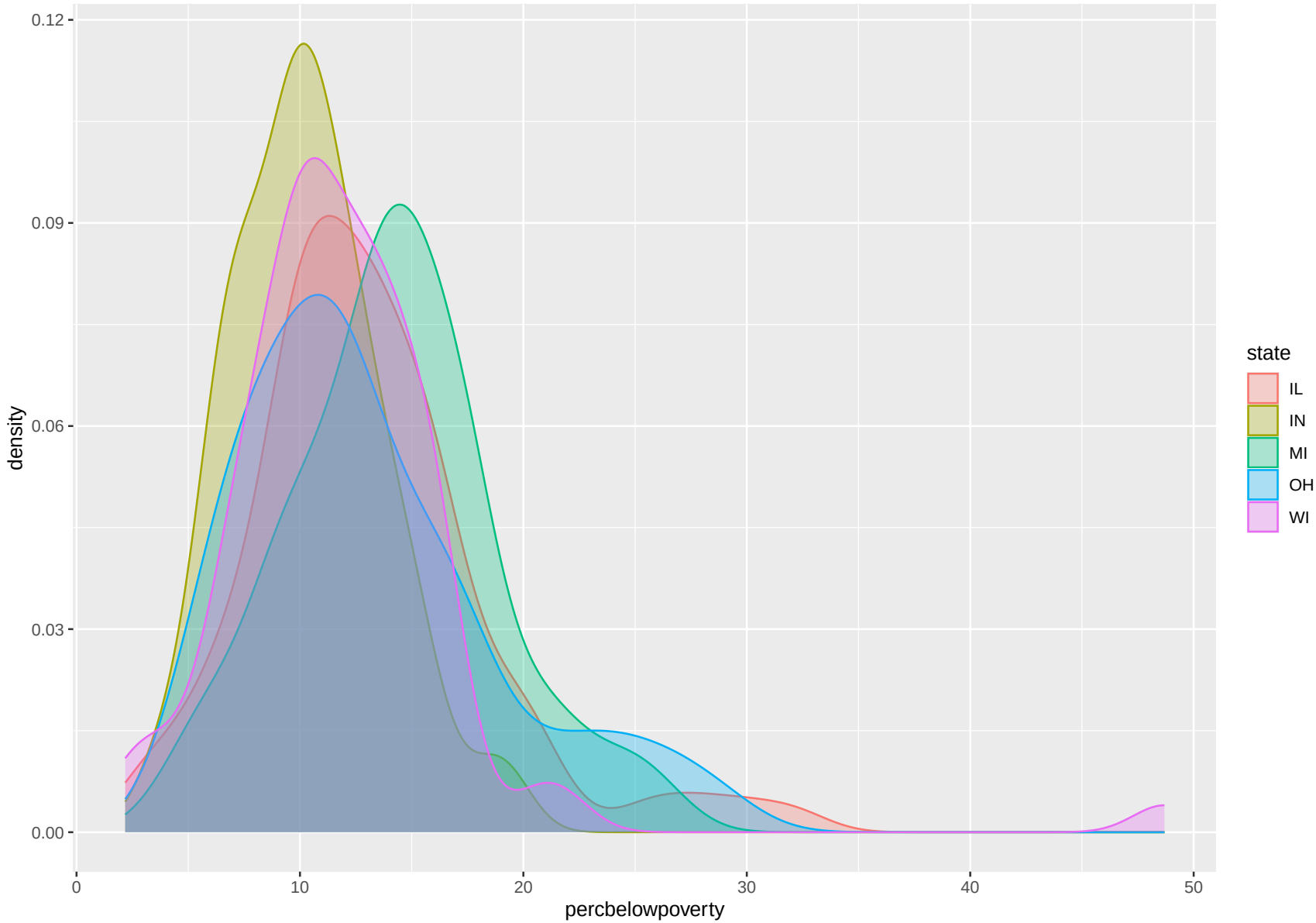
We'll see more of the ~ variable syntax (called a formula)


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

Density as alternative to histograms

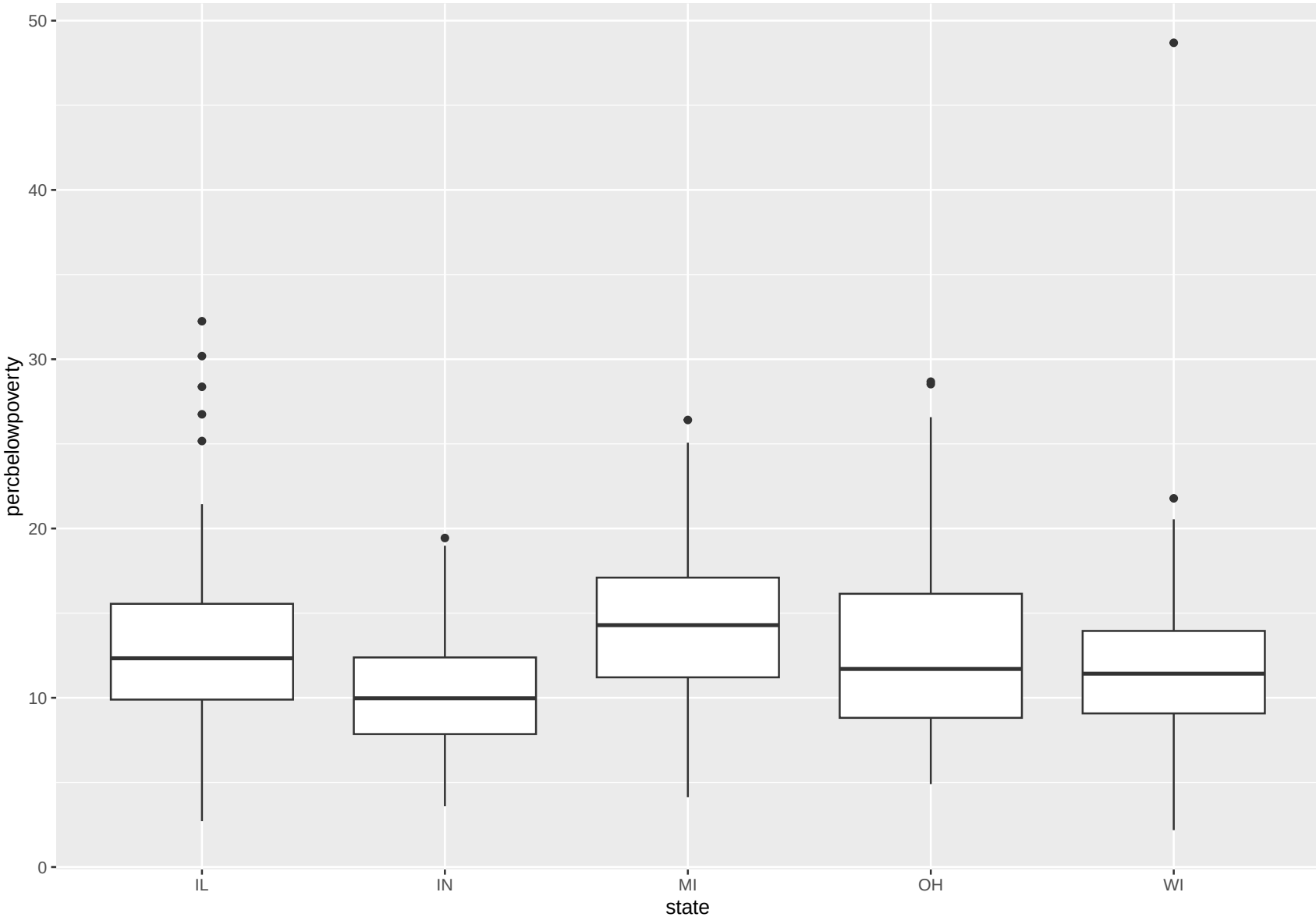
A **kernel density** plot is a smoothed version of a histogram and slightly easier to overlay

```
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 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

- "Box" represents middle 50% of the data

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width of the box is called the inter quartile range (IQR)

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width of the box is called the inter quartile range (IQR)
- Horizontal line in the box is the median

Boxplots in R

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width 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

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width 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" represents either:

Boxplots in R

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width 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" represents either:
  - 1.5 $\times$ IQR or max/min of the data, whichever is smaller

Boxplots in R

- "Box" represents middle 50% of the data
  - 25% of the data above the box, 25% below
  - Width 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" represents either:
  - $1.5 \times$ IQR or max/min of the data, whichever is smaller
  - Points beyond whiskers are outliers

4.3

Grouped data

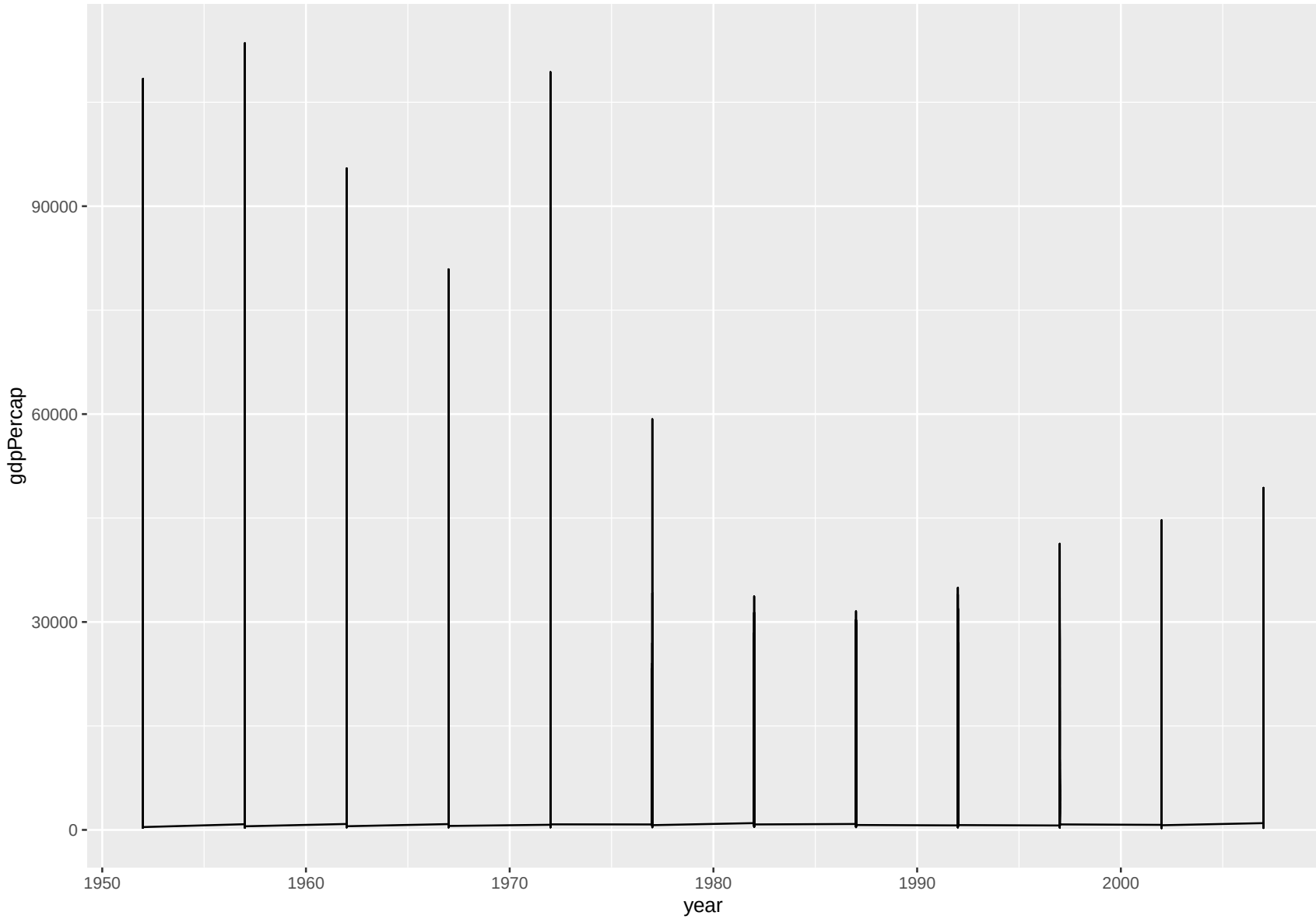
Gapminder data

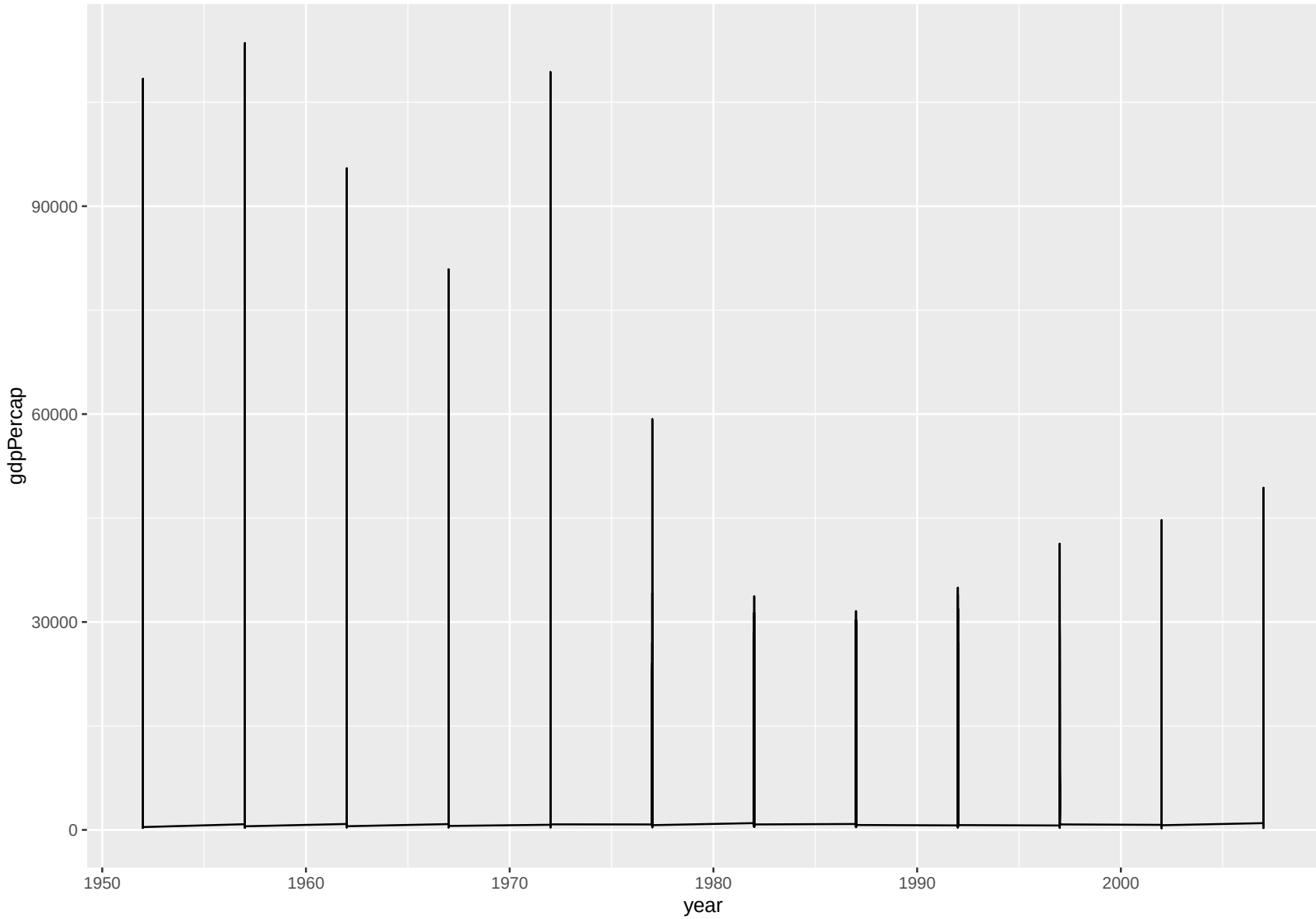
```
library(dplyr)
library(gapminder)
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanist~
## $ continent <fct> Asia, Asia, Asia, Asia, A~
## $ year      <int> 1952, 1957, 1962, 1967, 1~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 3~
## $ pop       <int> 8425333, 9240934, 1026708~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1~
```

Let's plot the trend in income

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

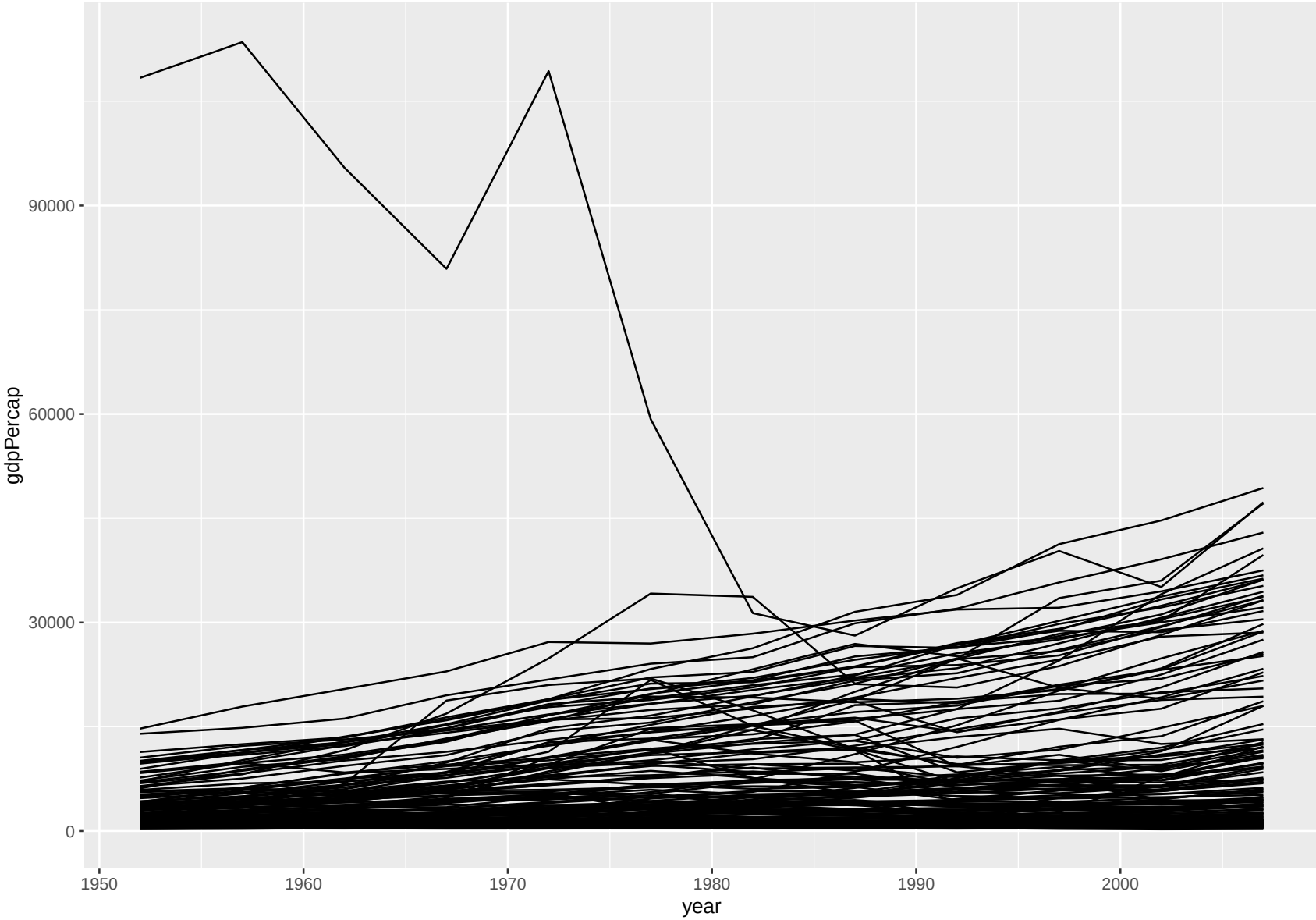

90000
60000
30000
0
gdpPercap
1950
1960
1970
1980
1990
2000
year



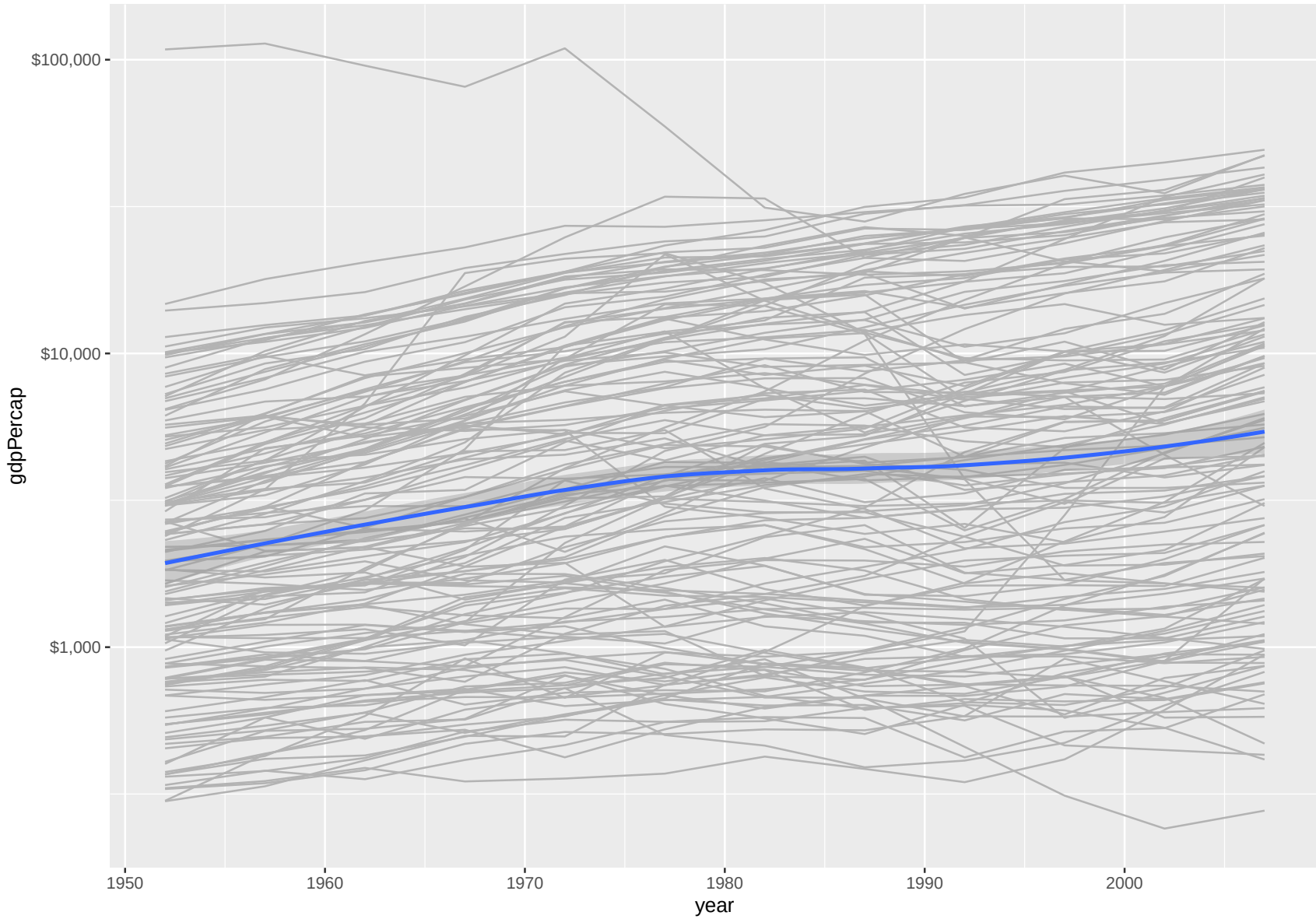
`geom_line` connects points from different countries in the same year.

Tell geom_line how to group the lines

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


90000
60000
30000
0
gdpPercap
1950
1960
1970
1980
1990
2000
year

```
ggplot(data = gapminder,
       mapping = aes(x = year,
                     y = gdpPercap)) +
  geom_line(mapping = aes(group = country), color = "grey7(
  geom_smooth(method = "loess") +
  scale_y_log10(labels = scales::dollar)
```


$100,000
$10,000
$1,000
gdpPercap
1950
1960
1970
1980
1990
2000
year