

# Introduction to the course (and R) and data visualization

Seung-Ho An, University of Arizona

- 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?

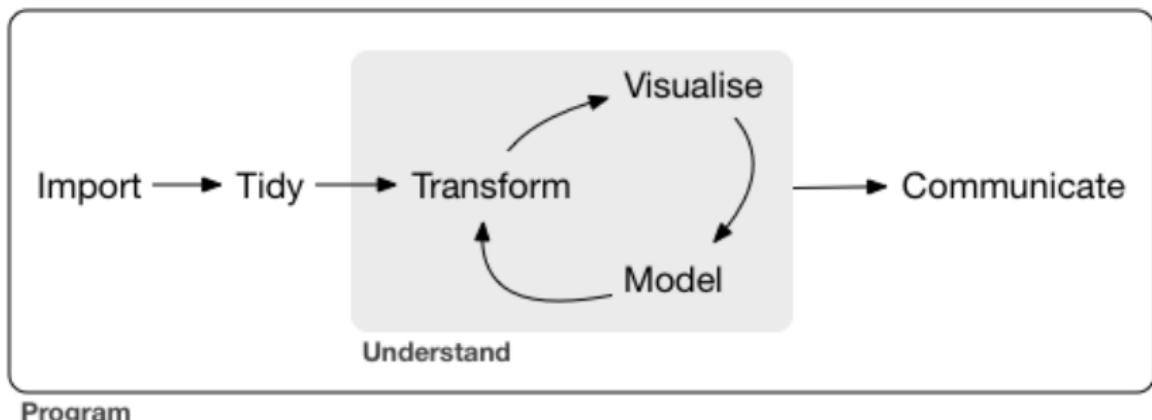


Figure 1: Data science process (Hadley)

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

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,

$$\begin{aligned}s^2 &= \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} \\&= \frac{\sum_{i=1}^n x^2 - \frac{(\sum_{i=1}^n x)^2}{n}}{n-1} \\&= \frac{(4+49+49+16+25+1+9) - \frac{(2+7+7+4+5+1+3)^2}{7}}{7-1} = \frac{153 - \frac{841}{7}}{6} \\&= 5.47619\end{aligned}$$

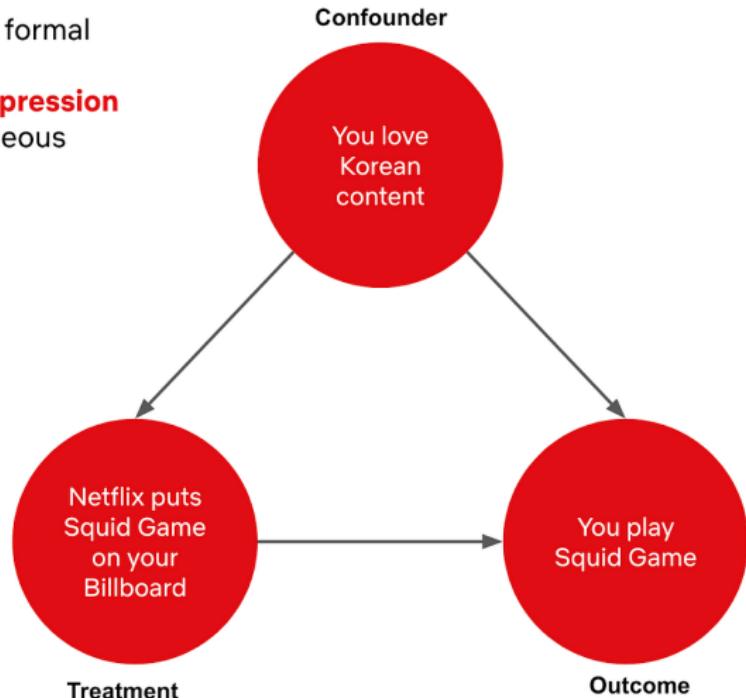
We will focus more on

```
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**)



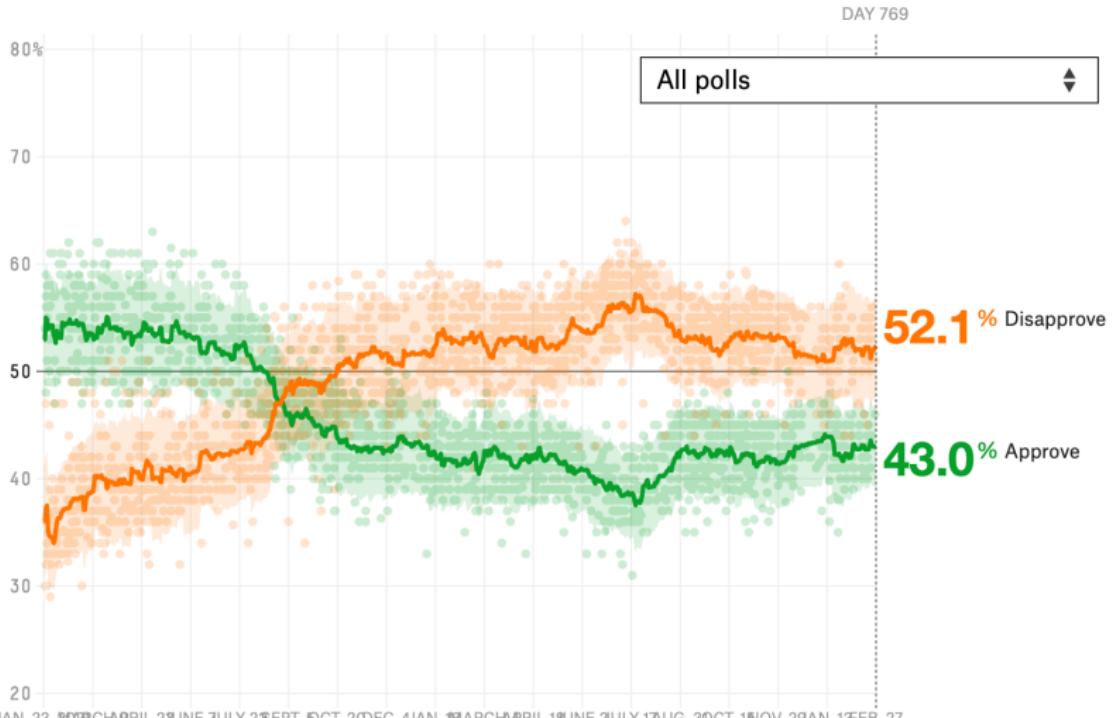
A new computer model uses publicly available data to predict crime accurately in eight U.S. cities, while revealing increased police response in wealthy neighborhoods at the expense of less advantaged areas.

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

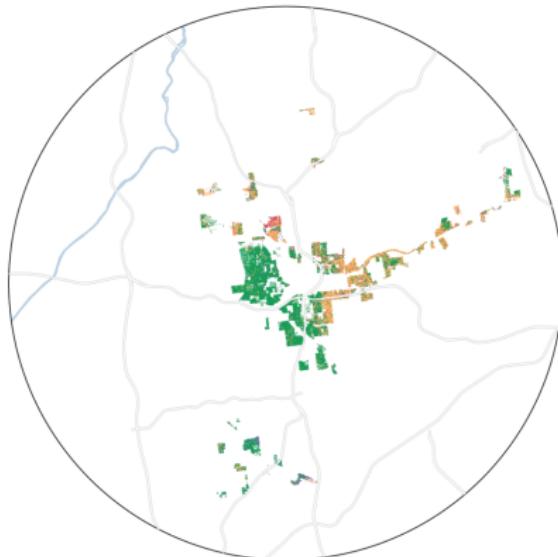


# Understanding how the past matters

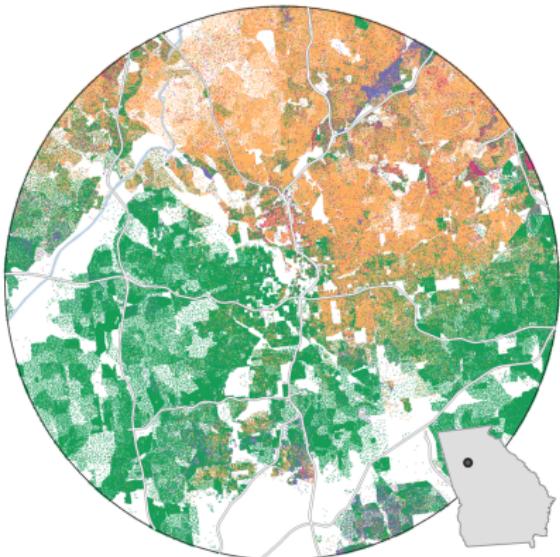
## Atlanta, GA

● WHITE ● BLACK ● LATINO ● ASIAN ● OTHER

ATLANTA'S "HAZARDOUS" 🚧 ZONES:



ATLANTA'S SURROUNDING AREA:



## 2. Course details

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

## What will you learn in this class?

- Summarize and visualize data

## What will you learn in this class?

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

## What will you learn in this class?

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

## What will you learn in this class?

- Summarize 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?

- Summarize 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?

- Summarize 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

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

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

# How to become a better programmer



**Hadley Wickham** @hadleywickham

...

The only way to write good code is to write tons of shitty code first.  
Feeling shame about bad code stops you from getting to good code

7:11 AM · Apr 17, 2015

# How to become a better programmer



**Hadley Wickham**   
@hadleywickham

...

The only way to write good code is to write tons of shitty code first.  
Feeling shame about bad code stops you from getting to good code

7:11 AM · Apr 17, 2015



**Allison Horst**  
@allison\_horst

...

Troubleshooting lessons I guess I'll just relearn forever:

- take a break
- it's almost certainly not a bug
- extra eyes are awesome
- spelling

## 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

- 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.

- 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**



**Where statistical computing lives**

- Touch-based interfaces
- Single app at a time

## Two computer revolutions



**The frontier of computing**



**Where statistical computing lives**

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

## 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-tasking, 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-tasking, 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-tasking, multiple windows
- Works heavily with the file system
- Underneath it's UNIX and the command line

## The Plain Person's Guide

~/>\_

## to Plain Text Social Science

Kieran Healy

- often free, open-sourced, and powerful

For more info, visit

<https://plain-text.co/>

## The Plain Person's Guide

~/>\_

## to Plain Text Social Science

Kieran Healy

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

For more info, visit

<https://plain-text.co/>

## The Plain Person's Guide

~/>\_

## to Plain Text Social Science

Kieran Healy

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

For more info, visit

<https://plain-text.co/>

## The Plain Person's Guide

~/>\_

## to Plain Text Social Science

Kieran Healy

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

For more info, visit

<https://plain-text.co/>

## The Plain Person's Guide

~/>\_

## to Plain Text Social Science

Kieran Healy

- 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?

For more info, visit

<https://plain-text.co/>

**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 touR

# R vs. RStudio

```
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)

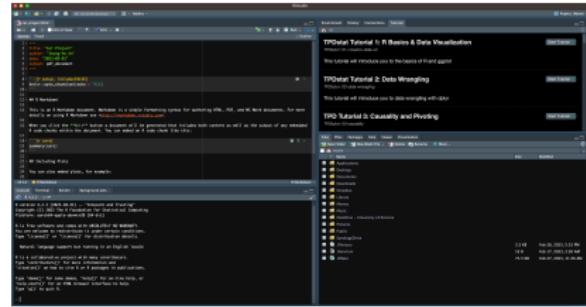
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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]
> |
```



## R vs. RStudio (cont)



```

RStudio
car-project.Rmd - Add file
Source | Visual
1: ---
2: title: "Car Project"
3: author: "Seung-Beom Kim"
4: date: "2023-02-27"
5: output: pdf_document
6: ---
7:
8: ----[r setup, include=FALSE]
9: knitr::opts_chunk$set(echo = TRUE)
10: ---
11:
12: ## R Markdown
13:
14: 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.
15:
16: 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:
17:
18: ----[r cars]
19: summary(cars)
20: -
21:
22: ## Including Plots
23:
24: You can also embed plots, for example:
25:
26: ----[r cars]
27: 
28: 
29: 
30: 
31: 
32: 
33: 
34: 
35: 
36: 
37: 
38: 
39: 
40: 
41: 
42: 
43: 
44: 
45: 
46: 
47: 
48: 
49: 
50: 
51: 
52: 
53: 
54: 
55: 
56: 
57: 
58: 
59: 
60: 
61: 
62: 
63: 
64: 
65: 
66: 
67: 
68: 
69: 
70: 
71: 
72: 
73: 
74: 
75: 
76: 
77: 
78: 
79: 
80: 
81: 
82: 
83: 
84: 
85: 
86: 
87: 
88: 
89: 
90: 
91: 
92: 
93: 
94: 
95: 
96: 
97: 
98: 
99: 
100: 
101: 
102: 
103: 
104: 
105: 
106: 
107: 
108: 
109: 
110: 
111: 
112: 
113: 
114: 
115: 
116: 
117: 
118: 
119: 
120: 
121: 
122: 
123: 
124: 
125: 
126: 
127: 
128: 
129: 
130: 
131: 
132: 
133: 
134: 
135: 
136: 
137: 
138: 
139: 
140: 
141: 
142: 
143: 
144: 
145: 
146: 
147: 
148: 
149: 
150: 
151: 
152: 
153: 
154: 
155: 
156: 
157: 
158: 
159: 
160: 
161: 
162: 
163: 
164: 
165: 
166: 
167: 
168: 
169: 
170: 
171: 
172: 
173: 
174: 
175: 
176: 
177: 
178: 
179: 
180: 
181: 
182: 
183: 
184: 
185: 
186: 
187: 
188: 
189: 
190: 
191: 
192: 
193: 
194: 
195: 
196: 
197: 
198: 
199: 
200: 
201: 
202: 
203: 
204: 
205: 
206: 
207: 
208: 
209: 
210: 
211: 
212: 
213: 
214: 
215: 
216: 
217: 
218: 
219: 
220: 
221: 
222: 
223: 
224: 
225: 
226: 
227: 
228: 
229: 
230: 
231: 
232: 
233: 
234: 
235: 
236: 
237: 
238: 
239: 
240: 
241: 
242: 
243: 
244: 
245: 
246: 
247: 
248: 
249: 
250: 
251: 
252: 
253: 
254: 
255: 
256: 
257: 
258: 
259: 
260: 
261: 
262: 
263: 
264: 
265: 
266: 
267: 
268: 
269: 
270: 
271: 
272: 
273: 
274: 
275: 
276: 
277: 
278: 
279: 
280: 
281: 
282: 
283: 
284: 
285: 
286: 
287: 
288: 
289: 
290: 
291: 
292: 
293: 
294: 
295: 
296: 
297: 
298: 
299: 
300: 
301: 
302: 
303: 
304: 
305: 
306: 
307: 
308: 
309: 
310: 
311: 
312: 
313: 
314: 
315: 
316: 
317: 
318: 
319: 
320: 
321: 
322: 
323: 
324: 
325: 
326: 
327: 
328: 
329: 
330: 
331: 
332: 
333: 
334: 
335: 
336: 
337: 
338: 
339: 
340: 
341: 
342: 
343: 
344: 
345: 
346: 
347: 
348: 
349: 
350: 
351: 
352: 
353: 
354: 
355: 
356: 
357: 
358: 
359: 
360: 
361: 
362: 
363: 
364: 
365: 
366: 
367: 
368: 
369: 
370: 
371: 
372: 
373: 
374: 
375: 
376: 
377: 
378: 
379: 
380: 
381: 
382: 
383: 
384: 
385: 
386: 
387: 
388: 
389: 
390: 
391: 
392: 
393: 
394: 
395: 
396: 
397: 
398: 
399: 
400: 
401: 
402: 
403: 
404: 
405: 
406: 
407: 
408: 
409: 
410: 
411: 
412: 
413: 
414: 
415: 
416: 
417: 
418: 
419: 
420: 
421: 
422: 
423: 
424: 
425: 
426: 
427: 
428: 
429: 
430: 
431: 
432: 
433: 
434: 
435: 
436: 
437: 
438: 
439: 
440: 
441: 
442: 
443: 
444: 
445: 
446: 
447: 
448: 
449: 
450: 
451: 
452: 
453: 
454: 
455: 
456: 
457: 
458: 
459: 
460: 
461: 
462: 
463: 
464: 
465: 
466: 
467: 
468: 
469: 
470: 
471: 
472: 
473: 
474: 
475: 
476: 
477: 
478: 
479: 
480: 
481: 
482: 
483: 
484: 
485: 
486: 
487: 
488: 
489: 
490: 
491: 
492: 
493: 
494: 
495: 
496: 
497: 
498: 
499: 
500: 
501: 
502: 
503: 
504: 
505: 
506: 
507: 
508: 
509: 
510: 
511: 
512: 
513: 
514: 
515: 
516: 
517: 
518: 
519: 
520: 
521: 
522: 
523: 
524: 
525: 
526: 
527: 
528: 
529: 
530: 
531: 
532: 
533: 
534: 
535: 
536: 
537: 
538: 
539: 
540: 
541: 
542: 
543: 
544: 
545: 
546: 
547: 
548: 
549: 
550: 
551: 
552: 
553: 
554: 
555: 
556: 
557: 
558: 
559: 
559: 
560: 
561: 
562: 
563: 
564: 
565: 
566: 
567: 
568: 
569: 
569: 
570: 
571: 
572: 
573: 
574: 
575: 
576: 
577: 
578: 
579: 
579: 
580: 
581: 
582: 
583: 
584: 
585: 
586: 
587: 
588: 
589: 
589: 
590: 
591: 
592: 
593: 
594: 
595: 
596: 
597: 
598: 
599: 
599: 
600: 
601: 
602: 
603: 
604: 
605: 
606: 
607: 
608: 
609: 
609: 
610: 
611: 
612: 
613: 
614: 
615: 
616: 
617: 
617: 
618: 
619: 
619: 
620: 
621: 
622: 
623: 
624: 
625: 
626: 
627: 
627: 
628: 
629: 
629: 
630: 
631: 
632: 
633: 
634: 
635: 
636: 
637: 
637: 
638: 
639: 
639: 
640: 
641: 
642: 
643: 
644: 
645: 
646: 
646: 
647: 
648: 
648: 
649: 
650: 
651: 
652: 
653: 
654: 
655: 
656: 
657: 
657: 
658: 
659: 
659: 
660: 
661: 
662: 
663: 
664: 
665: 
666: 
667: 
667: 
668: 
669: 
669: 
670: 
671: 
672: 
673: 
674: 
675: 
676: 
677: 
677: 
678: 
679: 
679: 
680: 
681: 
682: 
683: 
684: 
685: 
686: 
687: 
687: 
688: 
689: 
689: 
690: 
691: 
692: 
693: 
694: 
695: 
696: 
697: 
697: 
698: 
699: 
699: 
700: 
701: 
702: 
703: 
704: 
705: 
706: 
707: 
707: 
708: 
709: 
709: 
710: 
711: 
712: 
713: 
714: 
715: 
716: 
717: 
717: 
718: 
719: 
719: 
720: 
721: 
722: 
723: 
724: 
725: 
726: 
727: 
727: 
728: 
729: 
729: 
730: 
731: 
732: 
733: 
734: 
735: 
736: 
737: 
737: 
738: 
739: 
739: 
740: 
741: 
742: 
743: 
744: 
745: 
746: 
746: 
747: 
748: 
748: 
749: 
750: 
751: 
752: 
753: 
754: 
755: 
756: 
757: 
757: 
758: 
759: 
759: 
760: 
761: 
762: 
763: 
764: 
765: 
766: 
767: 
767: 
768: 
769: 
769: 
770: 
771: 
772: 
773: 
774: 
775: 
776: 
777: 
777: 
778: 
779: 
779: 
780: 
781: 
782: 
783: 
784: 
785: 
786: 
787: 
787: 
788: 
789: 
789: 
790: 
791: 
792: 
793: 
794: 
795: 
796: 
797: 
797: 
798: 
799: 
799: 
800: 
801: 
802: 
803: 
804: 
805: 
806: 
807: 
807: 
808: 
809: 
809: 
810: 
811: 
812: 
813: 
814: 
815: 
816: 
817: 
817: 
818: 
819: 
819: 
820: 
821: 
822: 
823: 
824: 
825: 
826: 
827: 
827: 
828: 
829: 
829: 
830: 
831: 
832: 
833: 
834: 
835: 
836: 
837: 
837: 
838: 
839: 
839: 
840: 
841: 
842: 
843: 
844: 
845: 
846: 
847: 
847: 
848: 
849: 
849: 
850: 
851: 
852: 
853: 
854: 
855: 
856: 
857: 
857: 
858: 
859: 
859: 
860: 
861: 
862: 
863: 
864: 
865: 
866: 
867: 
867: 
868: 
869: 
869: 
870: 
871: 
872: 
873: 
874: 
875: 
876: 
877: 
877: 
878: 
879: 
879: 
880: 
881: 
882: 
883: 
884: 
885: 
886: 
887: 
887: 
888: 
889: 
889: 
890: 
891: 
892: 
893: 
894: 
895: 
896: 
897: 
897: 
898: 
899: 
899: 
900: 
901: 
902: 
903: 
904: 
905: 
906: 
907: 
907: 
908: 
909: 
909: 
910: 
911: 
912: 
913: 
914: 
915: 
916: 
916: 
917: 
918: 
918: 
919: 
920: 
921: 
922: 
923: 
924: 
925: 
926: 
926: 
927: 
928: 
928: 
929: 
930: 
931: 
932: 
933: 
934: 
935: 
936: 
937: 
937: 
938: 
939: 
939: 
940: 
941: 
942: 
943: 
944: 
945: 
946: 
946: 
947: 
948: 
948: 
949: 
950: 
951: 
952: 
953: 
954: 
955: 
956: 
957: 
957: 
958: 
959: 
959: 
960: 
961: 
962: 
963: 
964: 
965: 
966: 
967: 
967: 
968: 
969: 
969: 
970: 
971: 
972: 
973: 
974: 
975: 
976: 
977: 
977: 
978: 
979: 
979: 
980: 
981: 
982: 
983: 
984: 
985: 
986: 
987: 
987: 
988: 
989: 
989: 
990: 
991: 
992: 
993: 
994: 
995: 
996: 
997: 
998: 
999: 
999: 
1000: 
1001: 
1002: 
1003: 
1004: 
1005: 
1006: 
1007: 
1008: 
1009: 
10010: 
10011: 
10012: 
10013: 
10014: 
10015: 
10016: 
10017: 
10018: 
10019: 
10020: 
10021: 
10022: 
10023: 
10024: 
10025: 
10026: 
10027: 
10028: 
10029: 
10030: 
10031: 
10032: 
10033: 
10034: 
10035: 
10036: 
10037: 
10038: 
10039: 
10040: 
10041: 
10042: 
10043: 
10044: 
10045: 
10046: 
10047: 
10048: 
10049: 
10050: 
10051: 
10052: 
10053: 
10054: 
10055: 
10056: 
10057: 
10058: 
10059: 
10060: 
10061: 
10062: 
10063: 
10064: 
10065: 
10066: 
10067: 
10068: 
10069: 
10070: 
10071: 
10072: 
10073: 
10074: 
10075: 
10076: 
10077: 
10078: 
10079: 
10080: 
10081: 
10082: 
10083: 
10084: 
10085: 
10086: 
10087: 
10088: 
10089: 
10090: 
10091: 
10092: 
10093: 
10094: 
10095: 
10096: 
10097: 
10098: 
10099: 
100100: 
100101: 
100102: 
100103: 
100104: 
100105: 
100106: 
100107: 
100108: 
100109: 
100110: 
100111: 
100112: 
100113: 
100114: 
100115: 
100116: 
100117: 
100118: 
100119: 
100120: 
100121: 
100122: 
100123: 
100124: 
100125: 
100126: 
100127: 
100128: 
100129: 
100130: 
100131: 
100132: 
100133: 
100134: 
100135: 
100136: 
100137: 
100138: 
100139: 
100140: 
100141: 
100142: 
100143: 
100144: 
100145: 
100146: 
100147: 
100148: 
100149: 
100150: 
100151: 
100152: 
100153: 
100154: 
100155: 
100156: 
100157: 
100158: 
100159: 
100160: 
100161: 
100162: 
100163: 
100164: 
100165: 
100166: 
100167: 
100168: 
100169: 
100170: 
100171: 
100172: 
100173: 
100174: 
100175: 
100176: 
100177: 
100178: 
100179: 
100180: 
100181: 
100182: 
100183: 
100184: 
100185: 
100186: 
100187: 
100188: 
100189: 
100190: 
100191: 
100192: 
100193: 
100194: 
100195: 
100196: 
100197: 
100198: 
100199: 
100200: 
100201: 
100202: 
100203: 
100204: 
100205: 
100206: 
100207: 
100208: 
100209: 
100210: 
100211: 
100212: 
100213: 
100214: 
100215: 
100216: 
100217: 
100218: 
100219: 
100220: 
100221: 
100222: 
100223: 
100224: 
100225: 
100226: 
100227: 
100228: 
100229: 
100230: 
100231: 
100232: 
100233: 
100234: 
100235: 
100236: 
100237: 
100238: 
100239: 
100240: 
100241: 
100242: 
100243: 
100244: 
100245: 
100246: 
100247: 
100248: 
100249: 
100250: 
100251: 
100252: 
100253: 
100254: 
100255: 
100256: 
100257: 
100258: 
100259: 
100260: 
100261: 
100262: 
100263: 
100264: 
100265: 
100266: 
100267: 
100268: 
100269: 
100270: 
100271: 
100272: 
100273: 
100274: 
100275: 
100276: 
100277: 
100278: 
100279: 
100280: 
100281: 
100282: 
100283: 
100284: 
100285: 
100286: 
100287: 
100288: 
100289: 
100290: 
100291: 
100292: 
100293: 
100294: 
100295: 
100296: 
100297: 
100298: 
100299: 
100300: 
100301: 
100302: 
100303: 
100304: 
100305: 
100306: 
100307: 
100308: 
100309: 
100310: 
100311: 
100312: 
100313: 
100314: 
100315: 
100316: 
100317: 
100318: 
100319: 
100320: 
100321: 
100322: 
100323: 
100324: 
100325: 
100326: 
100327: 
100328: 
100329: 
100330: 
100331: 
100332: 
100333: 
100334: 
100335: 
100336: 
100337: 
100338: 
100339: 
100340: 
100341: 
100342: 
100343: 
100344: 
100345: 
100346: 
100347: 
100348: 
100349: 
100350: 
100351: 
100352: 
100353: 
100354: 
100355: 
100356: 
100357: 
100358: 
100359: 
100360: 
100361: 
100362: 
100363: 
100364: 
100365: 
100366: 
100367: 
100368: 
100369: 
100370: 
100371: 
100372: 
100373: 
100374: 
100375: 
100376: 
100377: 
100378: 
100379: 
100380: 
100381: 
100382: 
100383: 
100384: 
100385: 
100386: 
100387: 
100388: 
100389: 
100390: 
100391: 
100392: 
100393: 
100394: 
100395: 
100396: 
100397: 
100398: 
100399: 
100400: 
100401: 
100402: 
100403: 
100404: 
100405: 
100406: 
100407: 
100408: 
100409: 
100410: 
100411: 
100412: 
100413: 
100414: 
100415: 
100416: 
100417: 
100418: 
100419: 
100420: 
100421: 
100422: 
100423: 
100424: 
100425: 
100426: 
100427: 
100428: 
100429: 
100430: 
100431: 
100432: 
100433: 
100434: 
100435: 
100436: 
100437: 
100438: 
100439: 
100440: 
100441: 
100442: 
100443: 
100444: 
100445: 
100446: 
100447: 
100448: 
100449: 
100450: 
100451: 
100452: 
100453: 
100454: 
100455: 
100456: 
100457: 
100458: 
100459: 
100460: 
100461: 
100462: 
100463: 
100464: 
100465: 
100466: 
100467: 
100468: 
100469: 
100470: 
100471: 
100472: 
100473: 
100474: 
100475: 
100476: 
100477: 
100478: 
100479: 
100480: 
100481: 
100482: 
100483: 
100484: 
100485: 
100486: 
100487: 
100488: 
100489: 
100490: 
100491: 
100492: 
100493: 
100494: 
100495: 
100496: 
100497: 
100498: 
100499: 
100500: 
100501: 
100502: 
100503: 
100504: 
100505: 
100506: 
100507: 
100508: 
100509: 
100510: 
100511: 
100512: 
100513: 
100514: 
100515: 
100516: 
100517: 
100518: 
100519: 
100520: 
100521: 
100522: 
100523: 
100524: 
100525: 
100526: 
100527: 
100528: 
100529: 
100530: 
100531: 
100532: 
100533: 
100534: 
100535: 
100536: 
100537: 
100538: 
100539: 
100540: 
100541: 
100542: 
100543: 
100544: 
100545: 
100546: 
100547: 
100548: 
100549: 
100550: 
100551: 
100552: 
100553: 
100554: 
100555: 
100556: 
100557: 
100558: 
100559: 
100560: 
100561: 
100562: 
100563: 
100564: 
100565: 
100566: 
100567: 
100568: 
100569: 
100570: 
100571: 
100572: 
100573: 
100574: 
100575: 
100576: 
100577: 
100578: 
100579: 
100580: 
100581: 
100582: 
100583: 
100584: 
100585: 
100586: 
100587: 
100588: 
100589: 
100590: 
100591: 
100592: 
100593: 
100594: 
100595: 
100596: 
100597: 
100598: 
100599: 
100600: 
100601: 
100602: 
100603: 
100604: 
100605: 
100606: 
100607: 
100608: 
100609: 
100610: 
100611: 
100612: 
100613: 
100614: 
100615: 
100616: 
100617: 
100618: 
100619: 
100620: 
100621: 
100622: 
100623: 
100624: 
100625: 
100626: 
100627: 
100628: 
100629: 
100630: 
100631: 
100632: 
100633: 
100634: 
100635: 
100636: 
100637: 
100638: 
100639: 
100640: 
100641: 
100642: 
100643: 
100644: 
100645: 
100646: 
100647: 
100648: 
100649: 
100650: 
100651: 
100652: 
100653: 
100654: 
100655: 
100656: 
100657: 
100658: 
100659: 
100660: 
100661: 
100662: 
100663: 
100664: 
100665: 
100666: 
100667: 
100668: 
100669: 
100670: 
100671: 
100672: 
100673: 
100674: 
100675: 
100676: 
100677: 
100678: 
100679: 
100680: 
100681: 
100682: 
100683: 
100684: 
100685: 
100686: 
100687: 
100688: 
100689: 
100690: 
100691: 
100692: 
100693: 
100694: 
100695: 
100696: 
100697: 
100698: 
100699: 
100700: 
100701: 
100702: 
100703: 
100704: 
100705: 
100706: 
100707: 
100708: 
100709: 
100710: 
100711: 
100712: 
100713: 
100714: 
100715: 
100716: 
100717: 
100718: 
100719: 
100720: 
100721: 
100722: 
100723: 
100724: 
100725: 
100726: 
100727: 
100728: 
100729: 
100730: 
100731: 
100732: 
100733: 
100734: 
100735: 
100736: 
100737: 
100738: 
100739: 
100740: 
100741: 
100742: 
100743: 
100744: 
100745: 
100746: 
100747: 
100748: 
100749: 
100750: 
100751: 
100752: 
100753: 
100754: 
100755: 
100756: 
100757: 
100758: 
100759: 
100760: 
100761: 
100762: 
100763: 
100764: 
100765: 
100766: 
100767: 
100768: 
100769: 
100770: 
100771: 
100772: 
100773: 
100774: 
100775: 
100776: 
100777: 
100778: 
100779: 
100780: 
100781: 
100782: 
100783: 
100784: 
100785: 
100786: 
100787: 
100788: 
100789: 
100790: 
100791: 
100792: 
100793: 
100794: 
100795: 
100796: 
100797: 
100798: 
100799: 
100800: 
100801: 
100802: 
100803: 
100804: 
100805: 
100806: 
100807: 
100808: 
100809: 
100810: 
100811: 
100812: 
100813: 
100814: 
100815: 
100816: 
100817: 
100818: 
100819: 
100820: 
100821: 
100822: 
100823: 
100824: 
100825: 
100826: 
100827: 
100828: 
100829: 
100830: 
100831: 
100832: 
100833: 
100834: 
100835: 
100836: 
100837: 
100838: 
100839: 
100840: 
100841: 
100842: 
100843: 
100844: 
100845: 
100846: 
100847: 
100848: 
100849: 
100850: 
100851: 
100852: 
100853: 
100854: 
100855: 
100856: 
100
```

The screenshot shows the RStudio interface. On the left, a R Markdown file titled "car-project.Rmd" is open. The code includes a header, a section on R Markdown, and a section on including plots. The R console at the bottom shows the R version 4.2.2 startup message. On the right, there is a sidebar with three tutorial sections: "TPDstat Tutorial 1: R Basics & Data Visualization", "TPDstat Tutorial 2: Data Wrangling", and "TPDstat Tutorial 3: Causality and Pivoting".

```

1: ---
2: title: "Car Project"
3: author: "Seung-Jo Ahn"
4: output: pdf_document
5: output: pdf_document
6: ---
7:
8: ----[r setup, include=FALSE]
9: knitr::opts_chunk$set(echo = TRUE)
10: ---
11:
12: ## R Markdown
13:
14: 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.
15:
16: 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:
17:
18: ----[r cars]
19: summary(cars)
20: ---
21:
22: ## Including Plots
23:
24: You can also embed plots, for example:
25:
26: ----[r cars]
27: library(ggplot2)
28: ggplot(cars, aes(x = .))
29: 
```

R version 4.2.2 (2022-10-31) -- "Tessenderlo Trysting"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: arm64-pc-apple-darwin16 (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.

>

**TPDstat Tutorial 1: R Basics & Data Visualization**  
TPDstat: 01-basics-data-init [Start Tutorial](#)

This tutorial will introduce you to the basics of R and ggplot

**TPDstat Tutorial 2: Data Wrangling**  
TPDstat: 02-data-wrangling [Start Tutorial](#)

This tutorial will introduce you to data wrangling with dplyr

**TPDstat Tutorial 3: Causality and Pivoting**  
TPDstat: 03-causality [Start Tutorial](#)

The screenshot shows the RStudio interface with several windows open:

- Code Editor:** Displays an R Markdown script. The code includes sections for setting up the environment, including packages, and creating plots. It also shows how to embed R code chunks directly into the document.
- Browser:** Shows three R-related tutorials:
  - TPDstat Tutorial 1: R Basics & Data Visualization**
  - TPDstat Tutorial 2: Data Wrangling**
  - TPDstat Tutorial 3: Causality and Pivoting**
- File Browser:** Shows the local file system structure under "Home".
- Console:** Displays the R command-line interface with version information and a help message about the R software.

**Console: run code, send code to here, inspect output**

```

R version 4.3.3 (2023-10-30) -- "Tessons and Truffles"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: arm64-pc-apple-darwin18 (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,
"html.start()" for an HTML browser interface to help.
Type "??" to quit R.

>
  
```

The image shows a composite screenshot of the RStudio interface. On the left, the RStudio code editor displays an R Markdown file named "car-project.Rmd". The code includes R code chunks, a summary of the file's purpose, and a note about embedding plots. Below the code editor is the R console window, which outputs the standard R startup message, information about the version (R 4.2.2), and a welcome message for new users.

On the right side of the interface, there is a "Project Files" sidebar. This sidebar lists several files and folders: Applications, Desktop, Documents, Downloads, Dockerfile, Library, Movies, Music, OneDrive - University of Arizona, Pictures, Public, SyncedDrive, Rhistory, Renviron, and XData. Each item has a "Size" and "Modified" column next to it, showing details like "3.3 KB Feb 28, 2023, 5:33 PM".

At the top right, there are three tabs for "TPDstat Tutorial": "1: R Basics & Data Visualization", "2: Data Wrangling", and "3: Causality and Pivoting". Each tab has a "Start Tutorial" button. The first tab is currently active.

Project files, plots, and help

The screenshot shows the RStudio interface with two main panes. The left pane displays an R Markdown document titled "car-project.Rmd". The code includes R code chunks, a summary of the document, and a note about embedding plots. The right pane shows three Tutorials:

- TPDstat Tutorial 1: R Basics & Data Visualization**: A brief introduction to R and ggplot.
- TPDstat Tutorial 2: Data Wrangling**: An introduction to data wrangling with dplyr.
- TPDstat Tutorial 3: Causality and Pivoting**: An introduction to causal inference and pivoting.

Below the tutorials is a file browser showing the contents of the user's Home directory.

```

1: ---
2: title: "Car Project"
3: author: "Seung-Beom Kim"
4: output: pdf_document
5: output: pdf_document
6: ---
7:
8: ----[r setup, include=FALSE]
9: knitr::opts_chunk$set(codetext = TRUE)
10: ---
11:
12: ## R Markdown
13:
14: 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.
15:
16: 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:
17:
18: ----[r cars]
19: summary(cars)
20: -
21:
22: ## Including Plots
23:
24: You can also embed plots, for example:
25:
26:
27: 12.14 [R] Markdown
28:
29: R version 4.2.2 (2022-10-31) -- "Tessenderlo Trysting"
30: Copyright (C) 2022 The R Foundation for Statistical Computing
31: Platform: arm64-pc-apple-darwin16 (64-bit)
32:
33: R is free software and comes with ABSOLUTELY NO WARRANTY.
34: You are welcome to redistribute it under certain conditions.
35: Type "license()" or "licence()" for distribution details.
36:
37: Natural language support but running in an English locale
38:
39: R is a collaborative project with many contributors.
40: Type "contributors()" for more information and
41: "citation()" on how to cite R or R packages in publications.
42:
43: Type "demo()" for some demos, "help()" for on-line help, or
44: "help.start()" for an HTML browser interface to help.
45: Type "q()" to quit R.
46:
47: >

```

### 3. Using RMarkdown

# The acts of coding

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

Figure 2: Writing codes

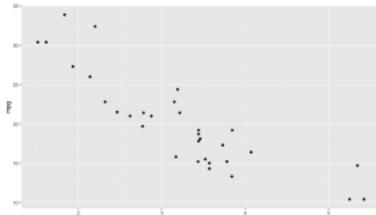
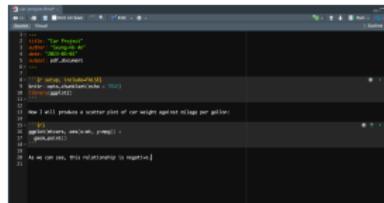


Figure 3: Looking at output



Figure 4: Taking notes

# Rmarkdown files to the rescue



A screenshot of the RStudio interface showing an RMarkdown file. The code editor contains the following R code:

```
1 title("Car Prices")
2
3 library(ggplot2)
4
5 #> #> #> #> #>
6
7 p <- ggplot(mtcars, aes(mpg))
8 p + geom_point()
9
10 #> #> #> #> #>
11
12 #> #> #> #> #>
13 #> #> #> #> #>
14 #> #> #> #> #>
```

The RStudio status bar at the bottom shows "Untitled.Rmd" and "100%".

Figure 5: Rmarkdown file

Keep code and notes  
together in plain text

## Rmarkdown files to the rescue

```
1 //title: "Top Projects"
2 //author: "John Doe"
3 //version: "1.0.0"
4 //category: "Data Science"
5 //tags: ["Machine Learning"]
6
7 //import statements
8 import tensorflow as tf
9 import numpy as np
10 import pandas as pd
11
12 # How will produce a scatter plot of our weight against mileage per gallon
13 # vs mpg
14 # We can see, there is a negative relationship between weight and mpg
15 # As we can see, this relationship is negative
16
```

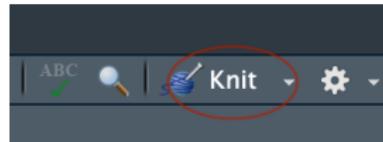


Figure 6: Knit in R

Figure 5: Rmarkdown file

Keep code and notes  
together in plain text

## Rmarkdown files to the rescue

```
1 // "Top Project"
2 // "ProjectA"
3 // "ProjectB"
4 // "ProjectC"
5
6 // [100, 100, 100]
7 // [100, 100, 100]
8 // [100, 100, 100]
9
10 // [100, 100, 100]
11 // [100, 100, 100]
12 // [100, 100, 100]
13
14 // How it will produce a scatter plot of our weight against miles per gallon
15 // [100, 100, 100]
16 // [100, 100, 100]
17 // [100, 100, 100]
18 // [100, 100, 100]
19
20 As we can see, this relationship is negative.
21
```

Figure 5: Rmarkdown file

Keep code and notes  
together in plain text

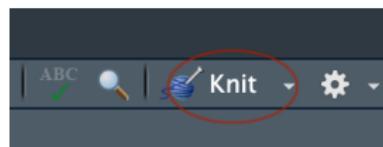


Figure 6: Knit in R

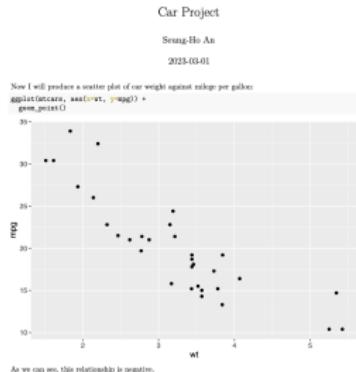


Figure 7: PDF output

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: 
```

```
horizontal rule (or slide break):
```

```
***
```

```
> block quote
```

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

```
1. ordered list
2. item 2
  + sub-item 1
```

## becomes

```
Plain text
End a line with two spaces to start a new paragraph.
italics and italics
bold and bold
superscript2
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
  - o sub-item 1
  - o sub-item 2

1. ordered list
2. item 2
  - o sub-item 1

```
---
```

```
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
```

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

Options

General

Code

Console

Appearance

Pane Layout

Packages

R Markdown

Python

Sweave

Spelling

Git/SVN

Publishing

Terminal

Accessibility

Basic   Graphics   Advanced

**R Sessions**

Default working directory (when not in a project): ~

Restore most recently opened project at startup

Restore previously open source documents at startup

**Workspace** Uncheck the box & change the status as never

Restore .RData into workspace at startup

Save workspace to .RData on exit:

**History**

Always save history (even when not saving .RData)

Remove duplicate entries in history

**Other**

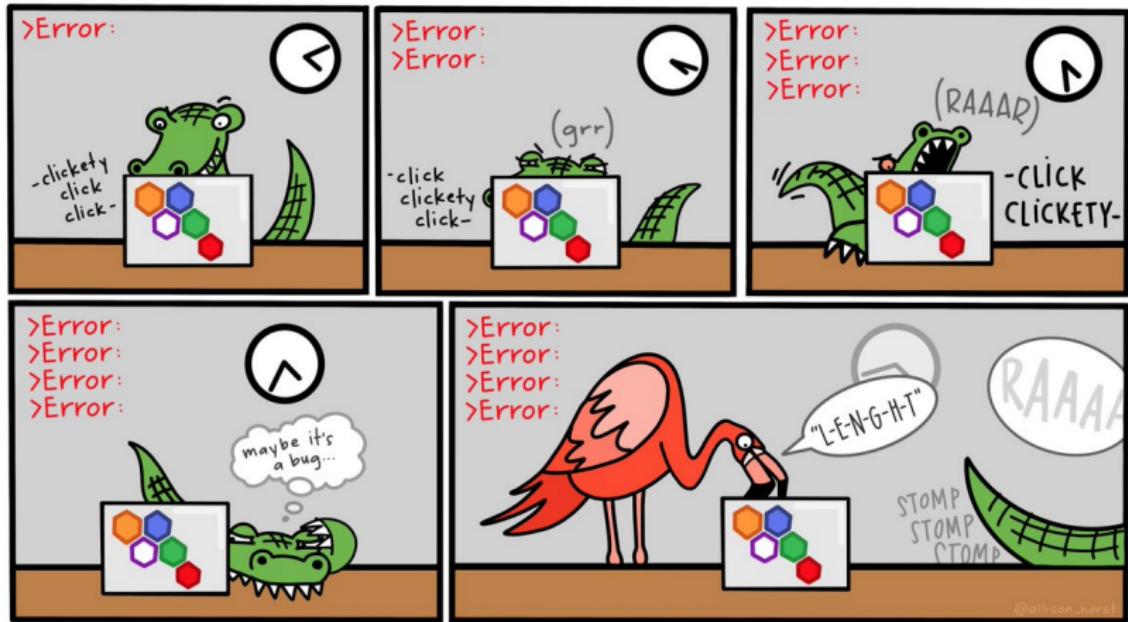
Wrap around when navigating to previous/next tab

Automatically notify me of updates to RStudio

Send automated crash reports to RStudio

**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
```

# What R looks like

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 users that we can use

## Functions live in packages

Packages are bundles of functions written by other users 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 users 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 users 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

# Functions live in packages

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
```

```
## # 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>,
## #   percasiain <dbl>, percother <dbl>,
## #   popadults <int>, perchsd <dbl>, ...
```

## Building up a graph in pieces

Create ggplot object and direct it to the correct data:

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

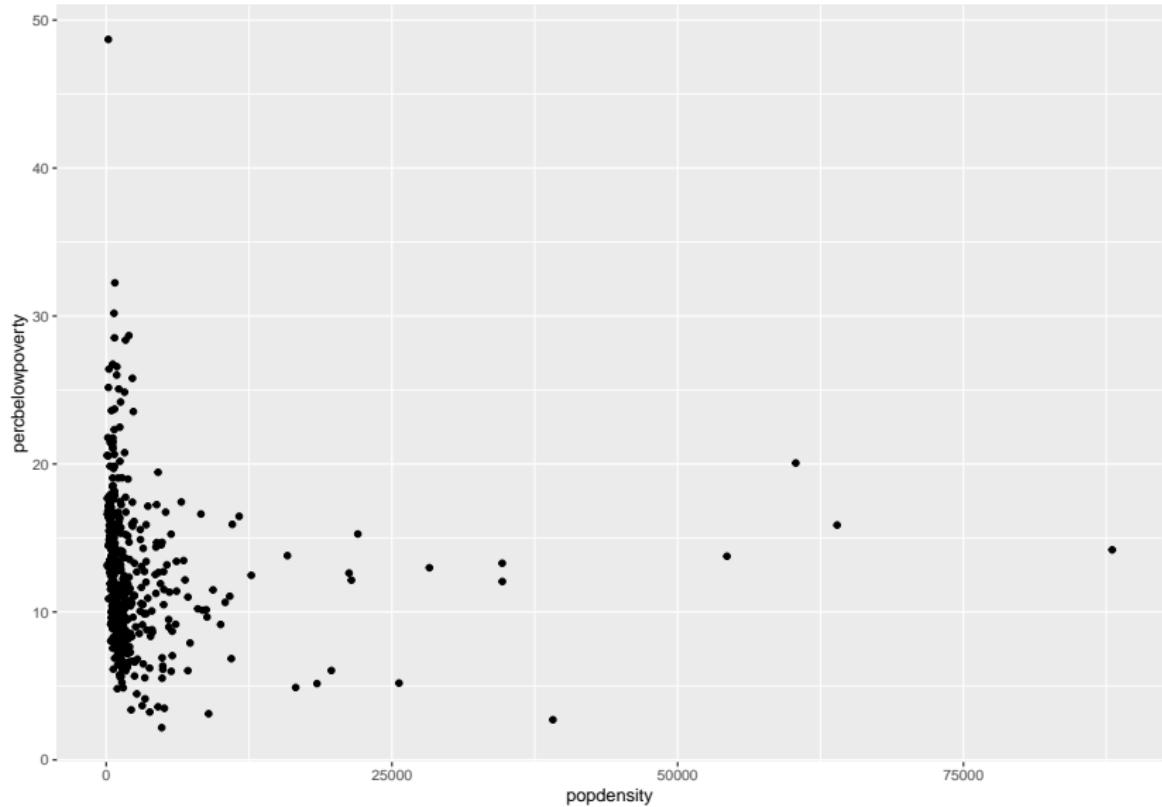
**Mapping**: tell ggplot what visual aesthetics correspond to which variables

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

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

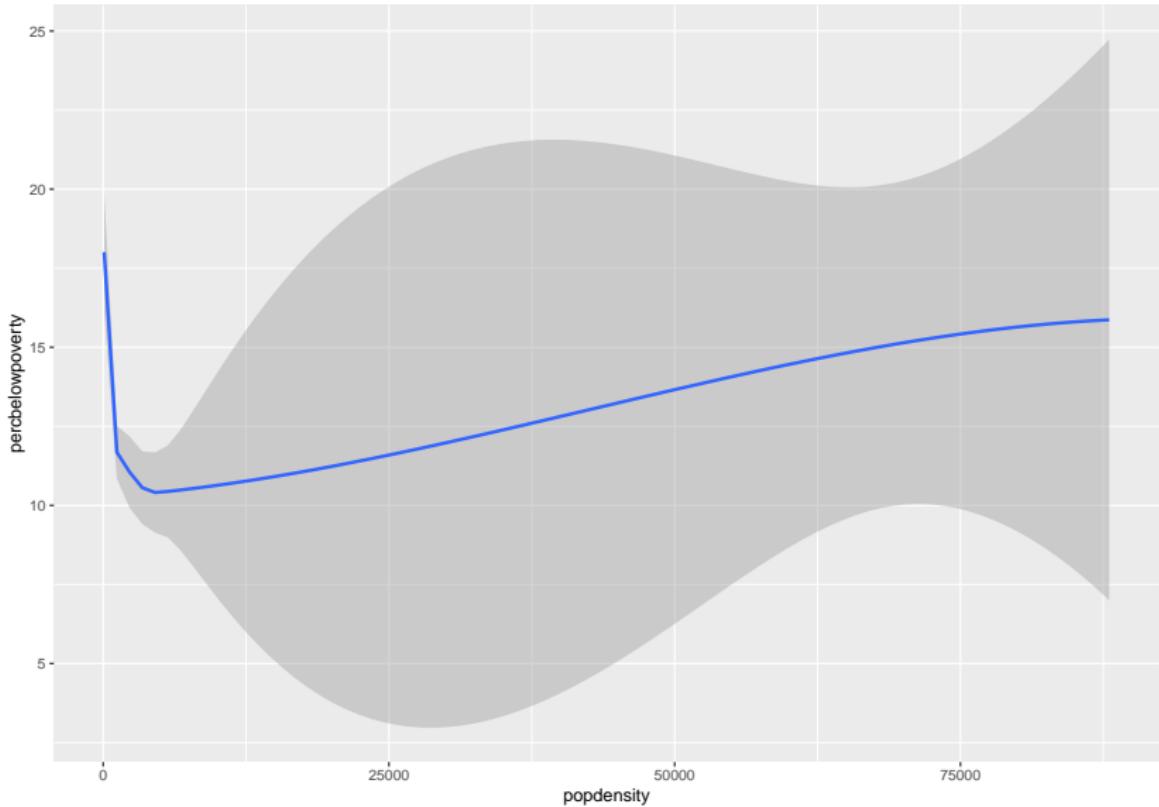
## Adding a geom layer

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



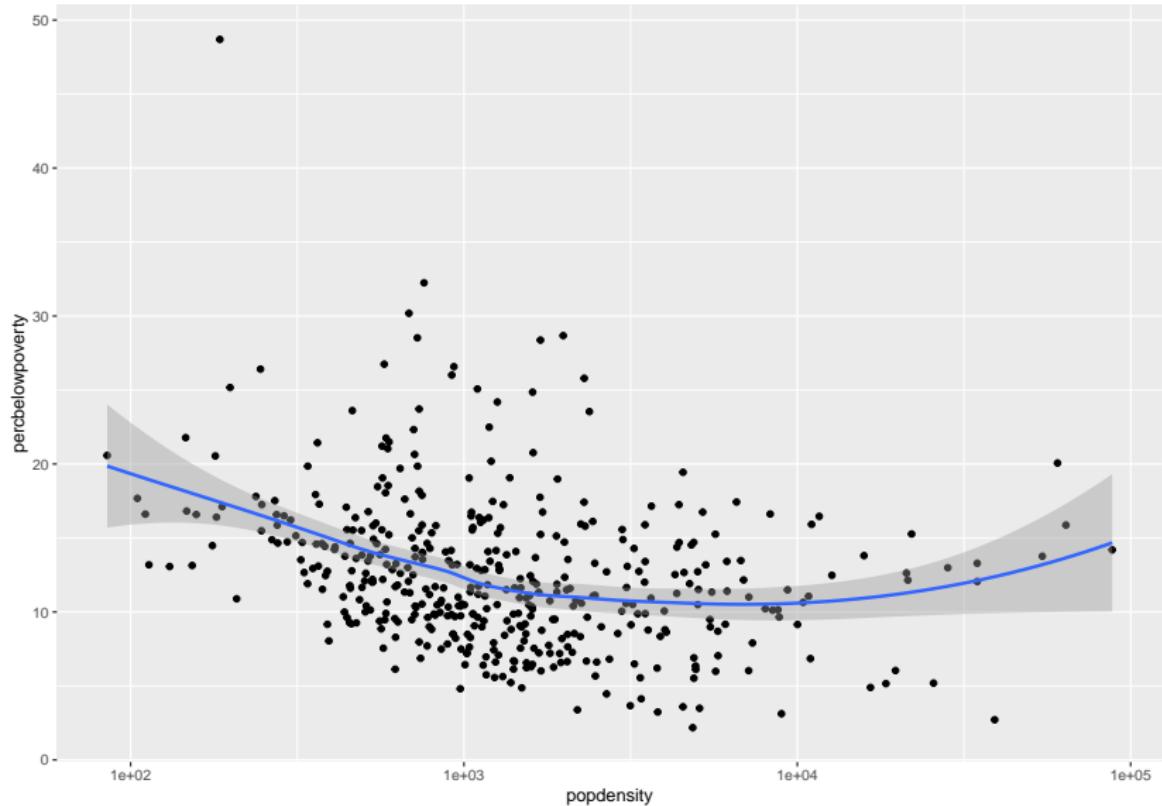
## Trying a new geom

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



## Layering geoms is additive

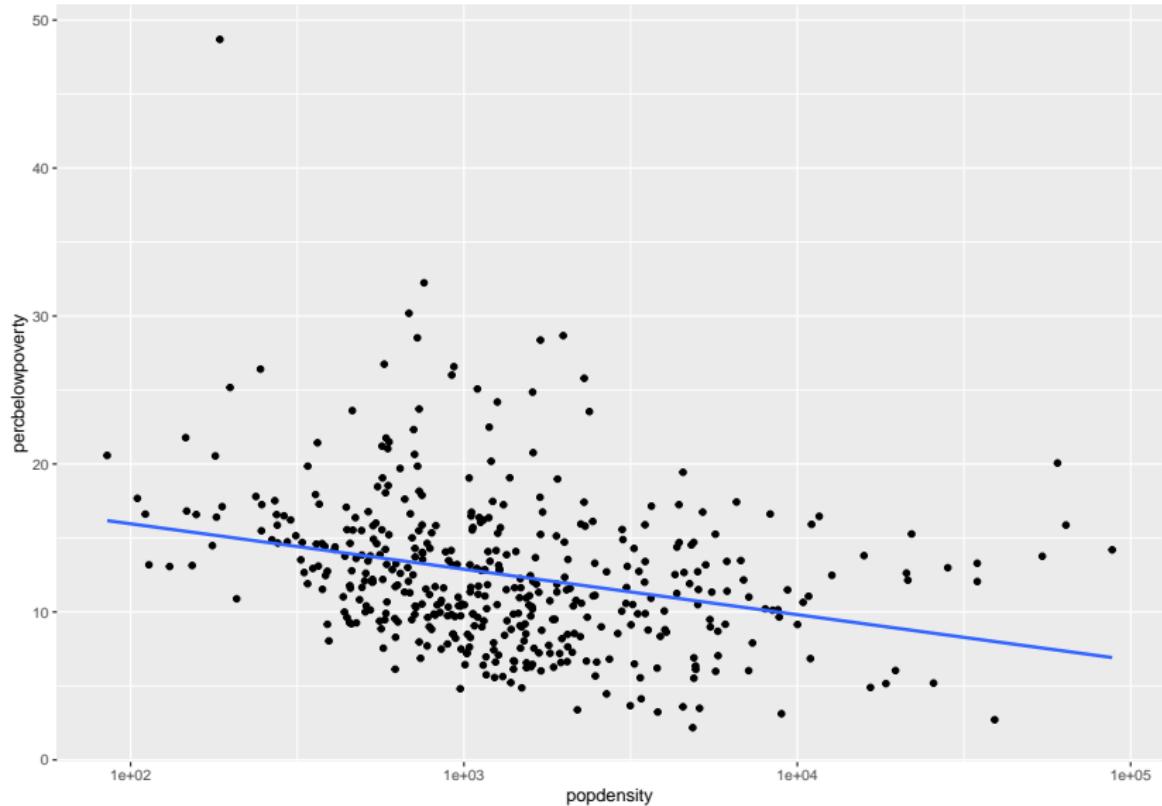
```
ggplot(data = midwest,  
        mapping = aes(x = popdensity,  
                      y = percbelowpoverty)) +  
  geom_point() +  
  geom_smooth() +  
  scale_x_log10()
```



Geoms can take arguments:

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

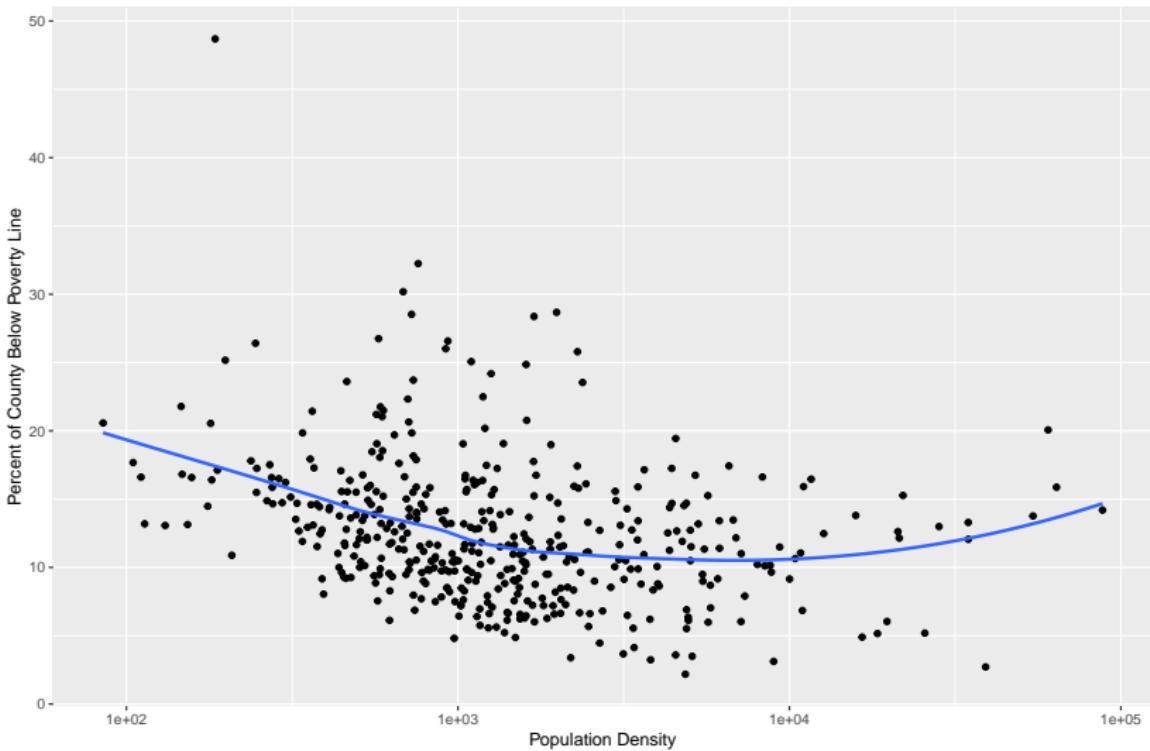
Tells `geom_smooth` to do a linear fit with no error region



## 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")
```

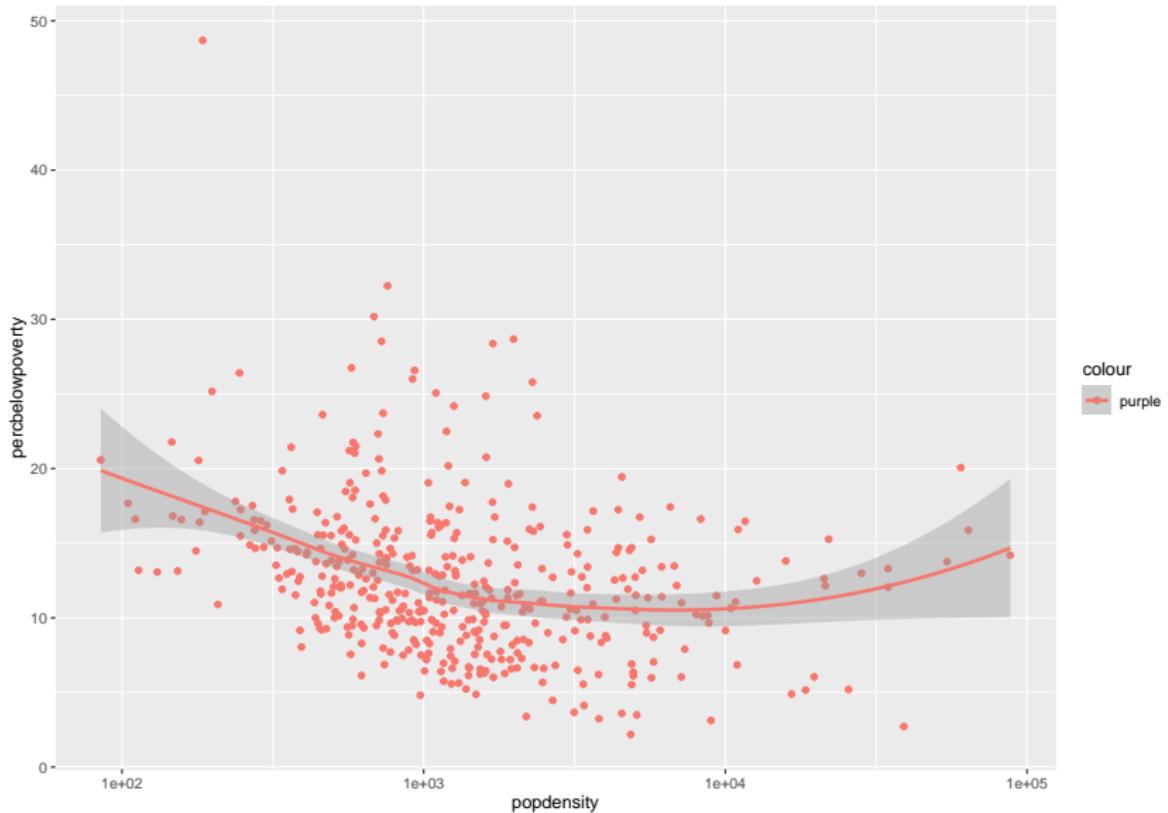
Poverty and Population Density  
Among Counties in the Midwest



## Mapping vs. setting aesthetics

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

Wait what?



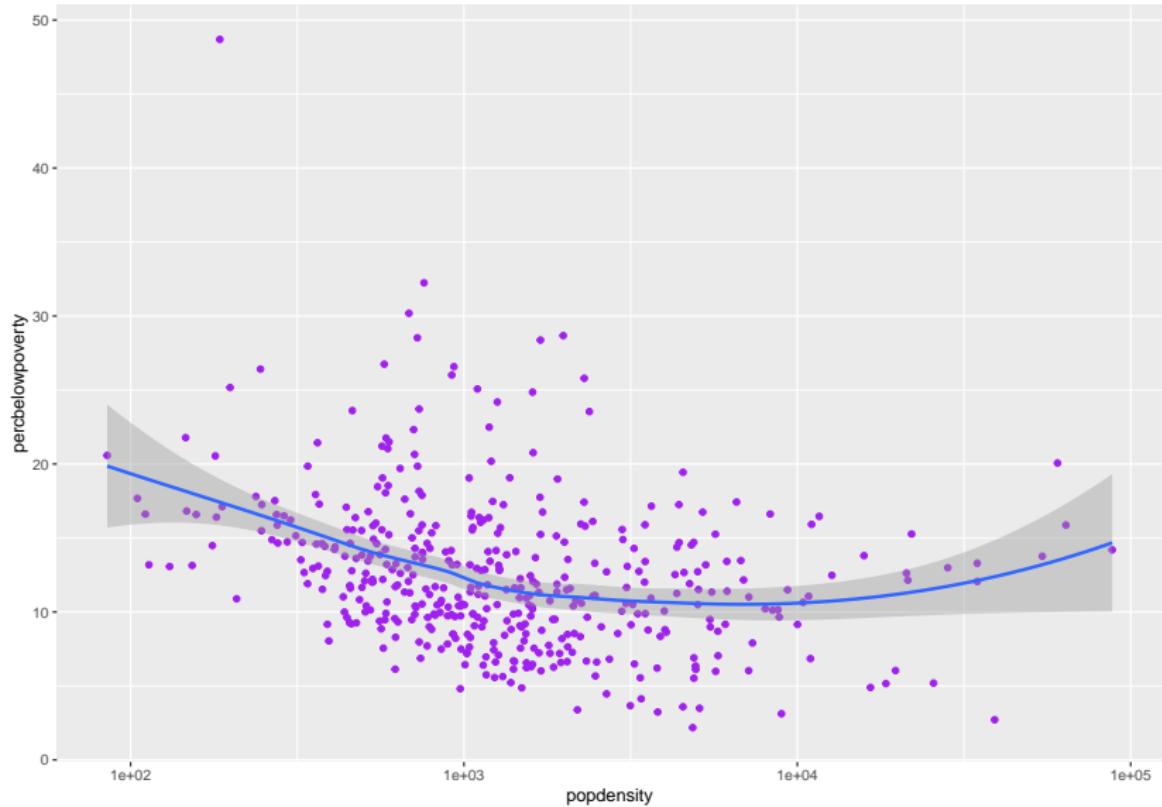
## Mapping always refers to variables

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()
```

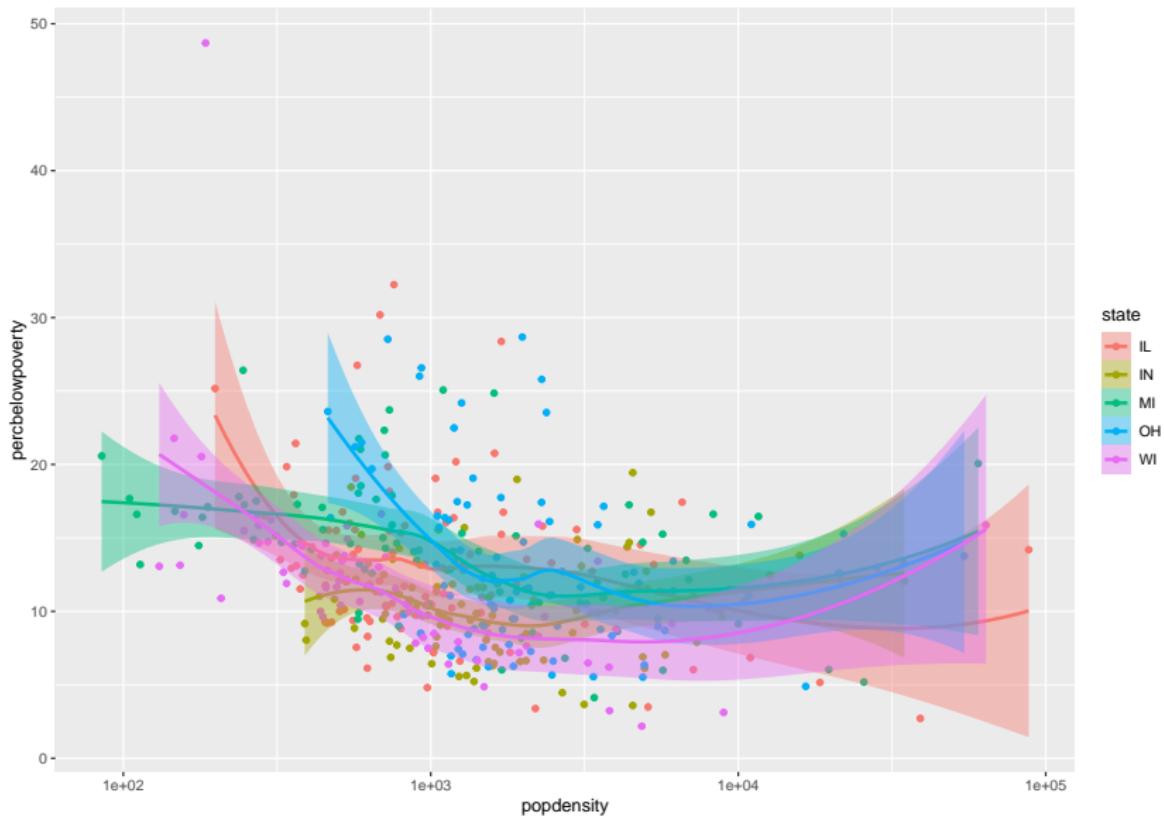
Set the color outside the mapping = aes() format

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



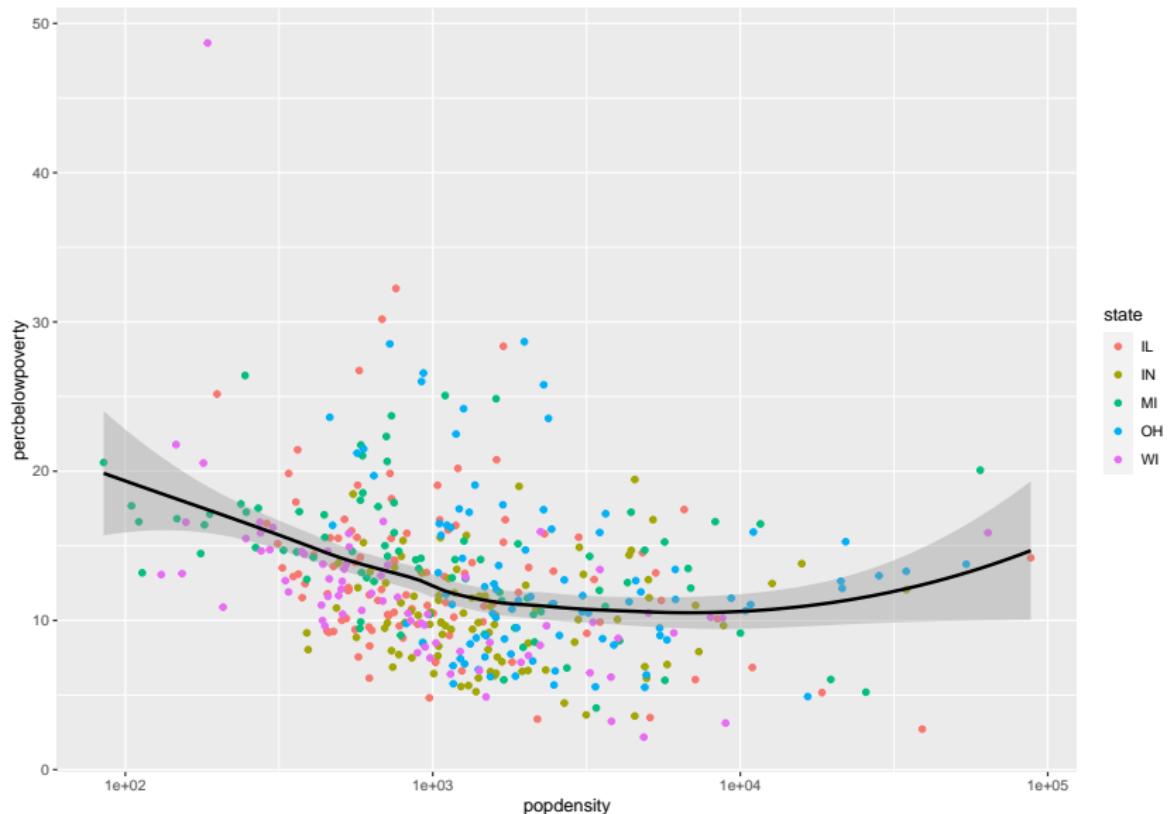
## Mapping more aesthetics

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



## 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()
```



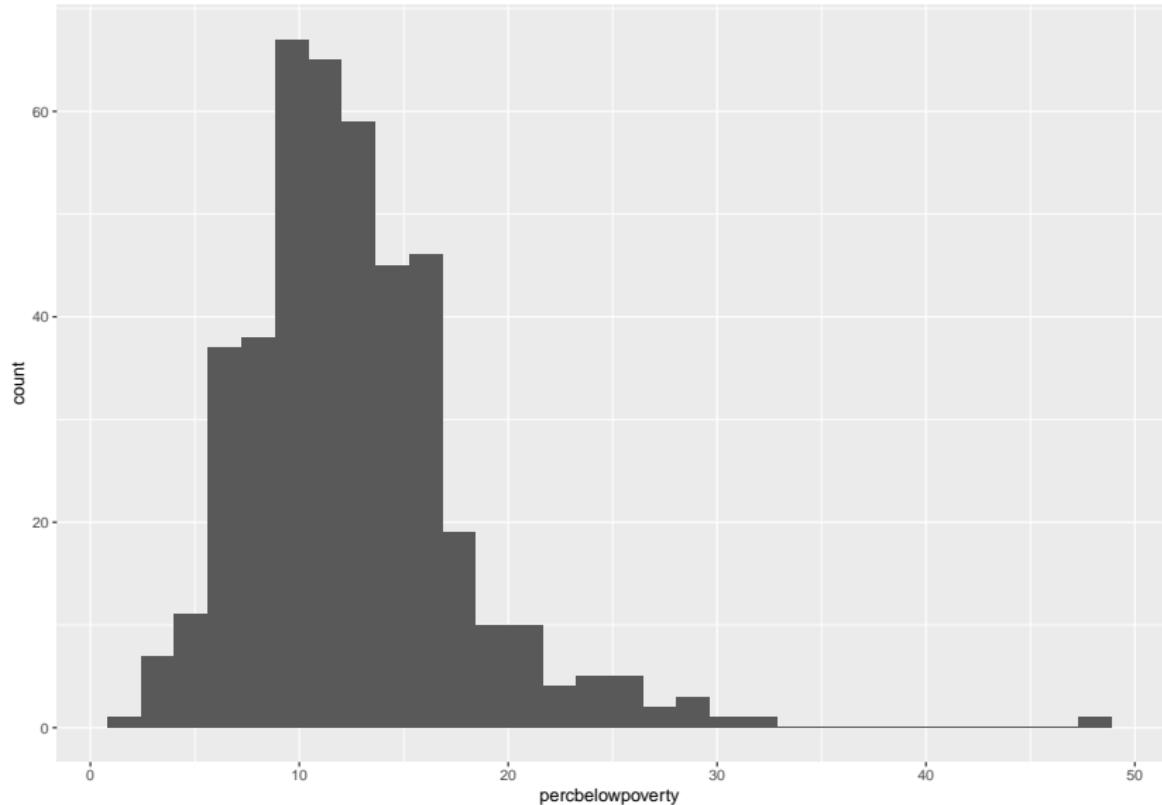
## 4.2

# 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

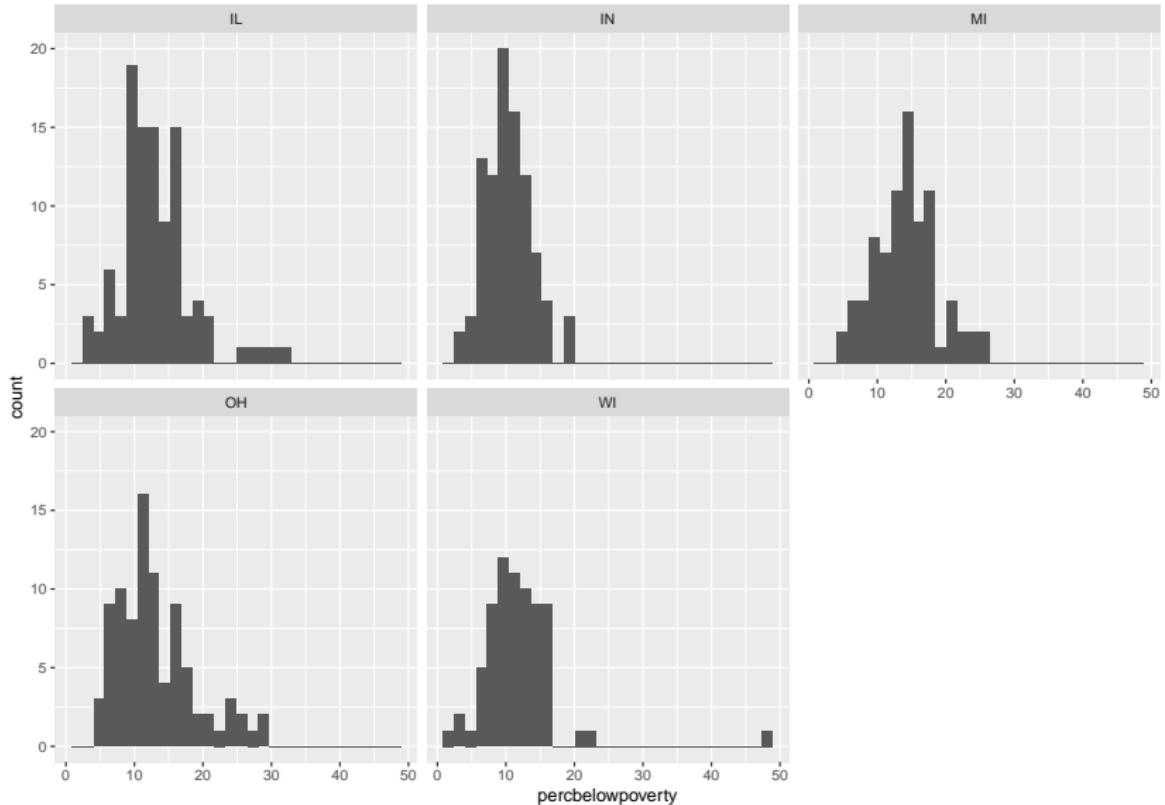


## Creating small multiples with facets

**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)
```

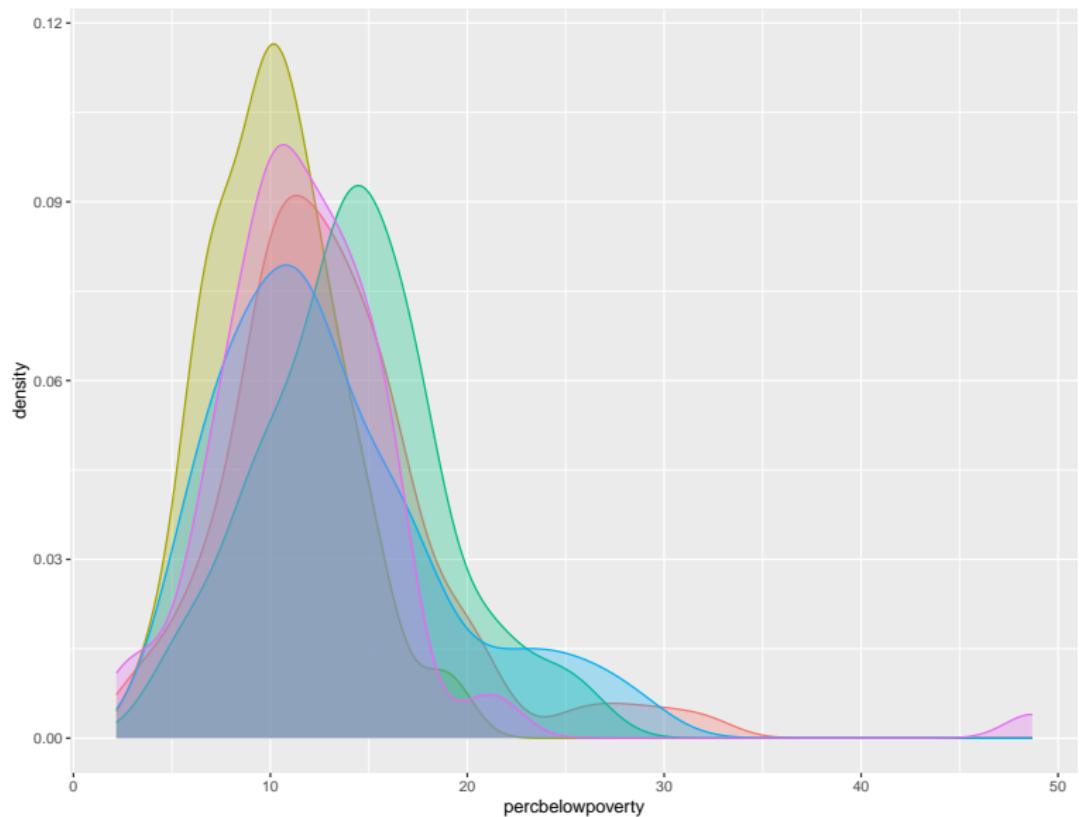
We'll see more of the `~` variable syntax (called a formula)



## Density as alternative to histograms

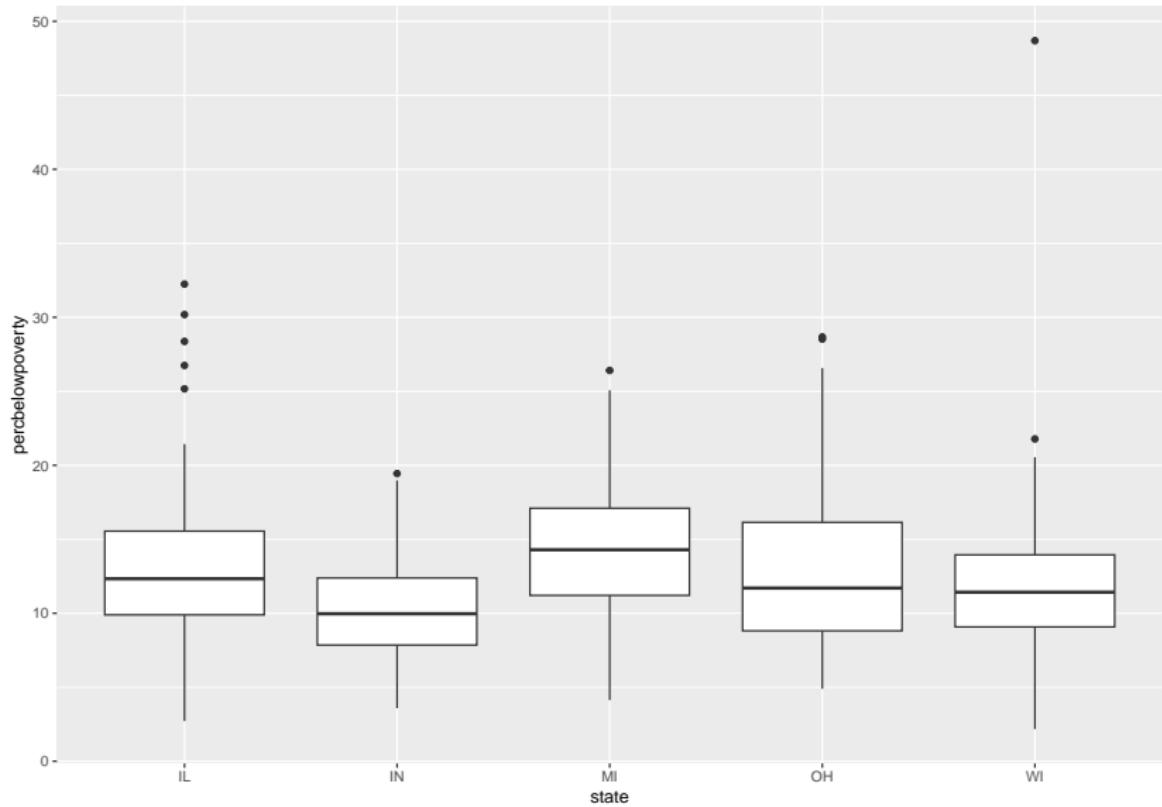
A **kernel density** plot is a smoothed version of a histogram and slightly easier to overlay

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



Boxplots are another way to compare distributions across discrete groups.

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



- “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

- “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:

- “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:
  - $1.5 \times \text{IQR}$  or max/min of the data, whichever is smaller

- “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:
  - $1.5 \times \text{IQR}$  or max/min of the data, whichever is smaller
  - Points beyond whiskers are outliers

## 4.3

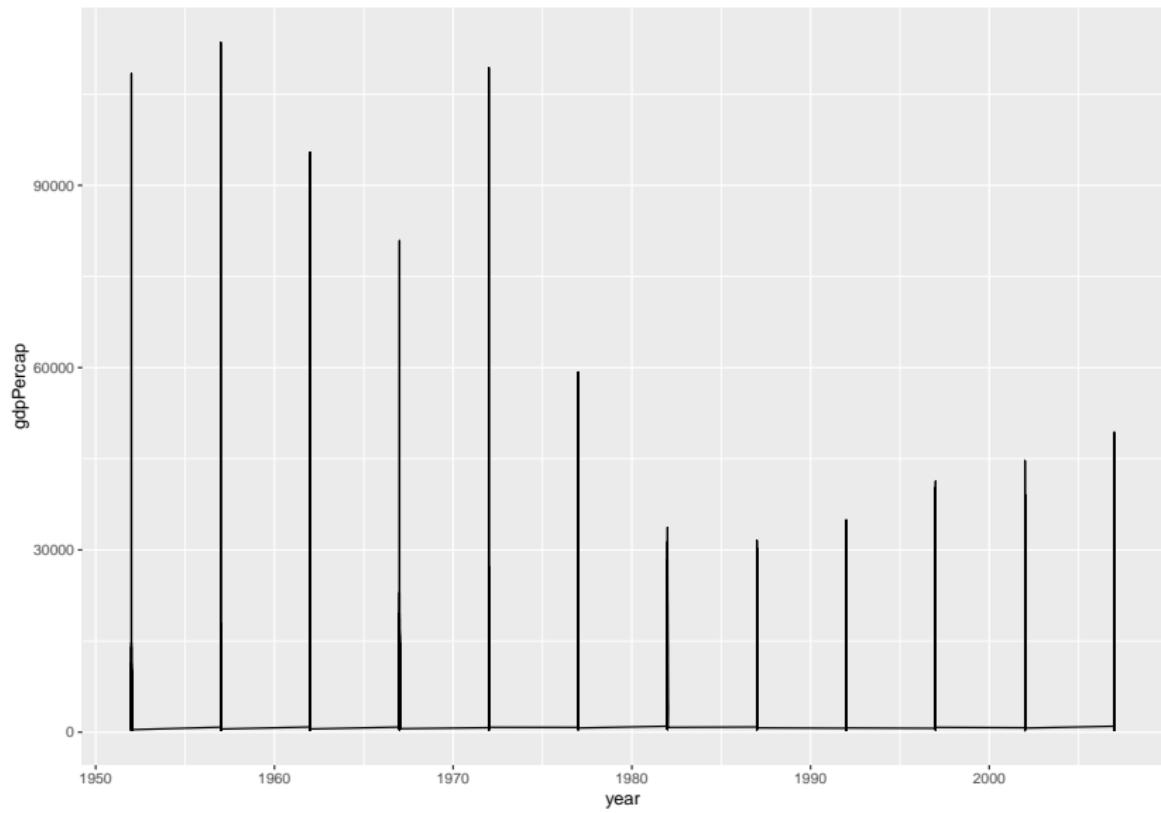
### Grouped data

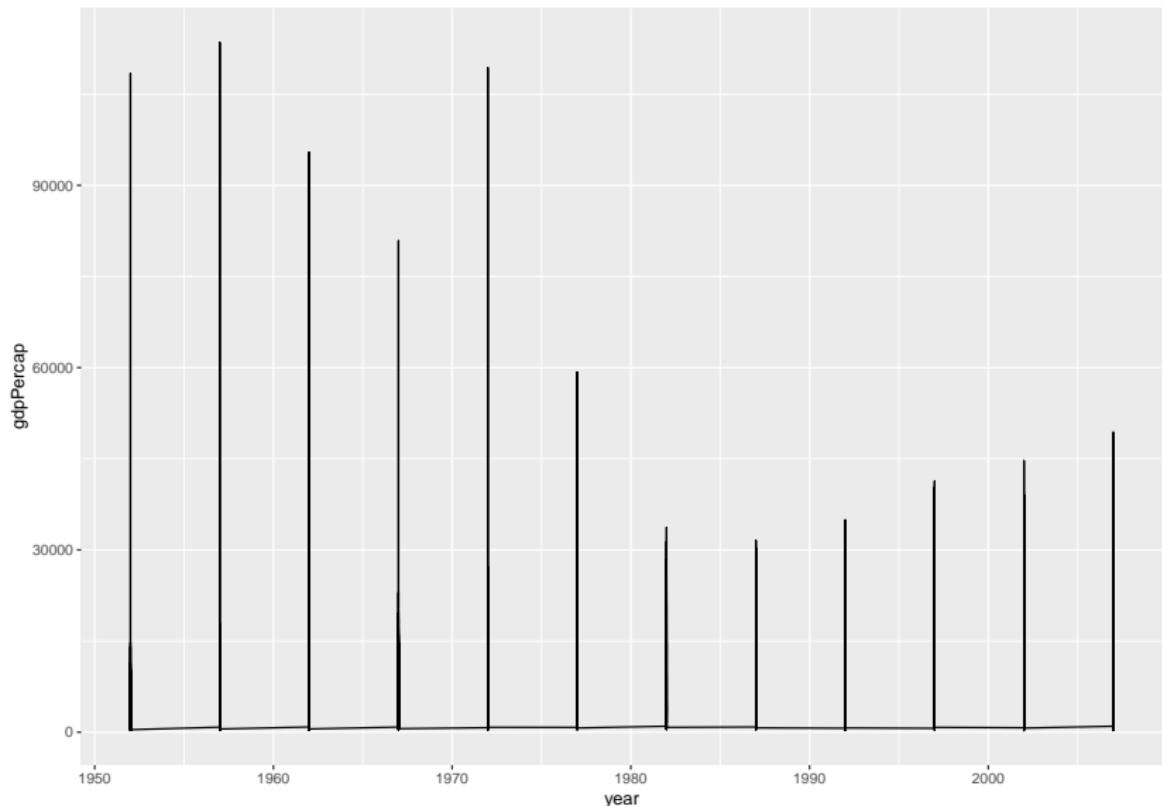
```
library(dplyr)
library(gapminder)
glimpse(gapminder)

## 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()
```

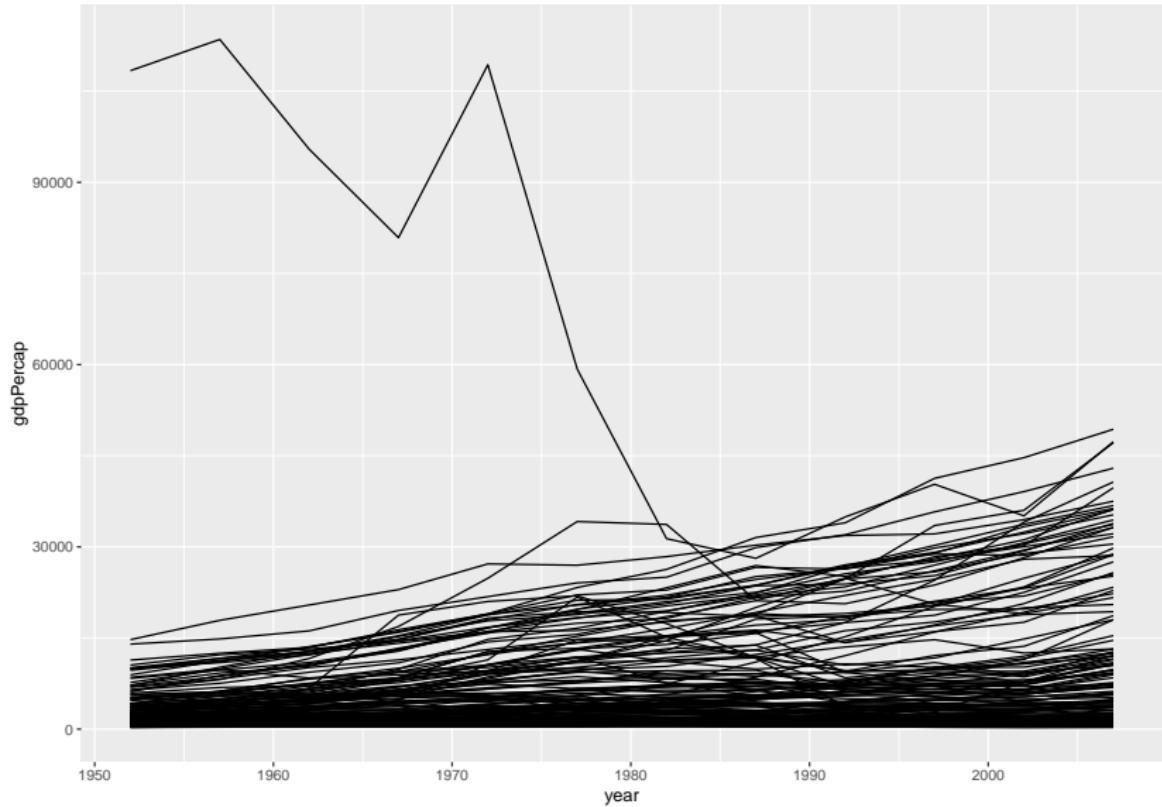




`geom_line` connects points from different countries in the same year.

## Tell geom\_line how to group the lines

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



```
ggplot(data = gapminder,
       mapping = aes(x = year,
                     y = gdpPercap)) +
  geom_line(mapping = aes(group = country), color = "grey70") +
  geom_smooth(method = "loess") +
  scale_y_log10(labels = scales::dollar)
```

