Big Data and Economics

Lecture 1: Introduction

Kyle Coombs (he/him/his)
Bates College | EC/DCS 368

Table of contents

1. Prologue
2. Dessert: What makes data big?
3. Veggies: Syllabus highlights
4. Appetizer: R for data science
5. Entree: Getting started
6. Second Dessert: Data visualization with ggplot2

Prologue

Introductions

Course

🌐 https://github.com/ECON368-fall2023-big-data-and-economics

You'll soon receive access to this repo, where we submit assignments, upload presentations, etc.

Me

Kyle Coombs

✉ kcoombs@bates.edu

🎓 Assistant Professor (economics)

📍 From Scotia, New York

🏠 Live in Maine and Massachusetts

You

A quick roundtable of names, fields/interests, and coding background.

Syllabus highlights

(Read the full document here.)

Why this course?

Fill in the gaps left by traditional econometrics and methods classes.

- Practical skills that tools that will benefit your thesis and future career.
- Neglected skills like how to actually find datasets in the wild and clean them.
- Apply skills to answer and address empirical questions on economic and *social* problems.

Data science skills are largely distinct from (and complementary to) the core 'metrics familiar to economists.

- Acquiring data; scraping; maintaining databases; etc.
- Data viz, cleaning and wrangling; programming; cloud computation; relational databases; machine learning; etc.

  > *"In short, we will cover things that I wish someone had taught me when I was starting out in college."*

You, at the end of this course

Syllabus readthrough

- Read over the syllabus for 3 minutes
- Identify at least one word you're unfamiliar with
- Anyone have guesses for how to define them?
- Mentimeter

Grading

Component	Weight	Graded
7 × homework assignments (10% each)	50%	Top 5
2 × short presentations (10% each)	10%	Top 1
1 × final project	40%	In parts

- Short presentations summarize either a key lecture reading, or an (approved) software package/platform.
- Extensions: Each of you gets three "grace period" days to extend deadlines.
    - You can use these days in any way you like, but once they're gone, they're gone.

PS — I'll also award a class participation bonus (2.5%) at my discretion.

Homework assignments

- 7 homework assignments will be assigned throughout the semester.
- Each assignment will be assigned and submitted via GitHub classroom
    - You will be submitting relevant data files, code, and a write-up
- 2 will be group assignments, the rest will be individual.
- I want you to practice pair coding and reconciling GitHub bugs

Presentations

- Every class will have a five-minute presentation on a key reading, software, data science technique, or programming skill
- You will sign up for a presentation slot on the course website
- Details on a strong presentation available in the syllabus
- You will submit a PDF (and any .tex or .Rmd files) of your slides to GitHub classroom

Final project

- Each of you will write an original 5-10 page research paper
- Throughout the semester, there will be three guided assignments that will help you develop your fine
- The final paper will be due on 12/11
- I expect your code to work and the results to be reproducible (i.e. I should be able to run your code and get the same results)

Component	Weight of Final	Date
Project Proposal	5%	9/22
Literature Review	5%	10/17
Data Description	5%	11/17
Peer Review	5%	12/1
Code	10%	12/11
Write up	10%	12/11
Total	**40%**	N/A

- Like problem sets, you will submit a PDF with all code and data files to reproduce your results
- Christine Murray has created a guide of potential data sources
- You can also check kaggle

Lecture outline

- Each lecture will cover either a "skill" (on Tuesdays) or an "application" (on Thursdays)

Data science basics

- Version control with Git and GitHub
- R language basics
- Data cleaning and wrangling
- Webscraping
- Data visualization

Analysis and Programming

- Regression analysis in R
- Spatial analysis in R
- Functions in R
- Parallel programming

Causal inference

- Regression discontinuity design
- Panel data and fixed effects
- Field experiments

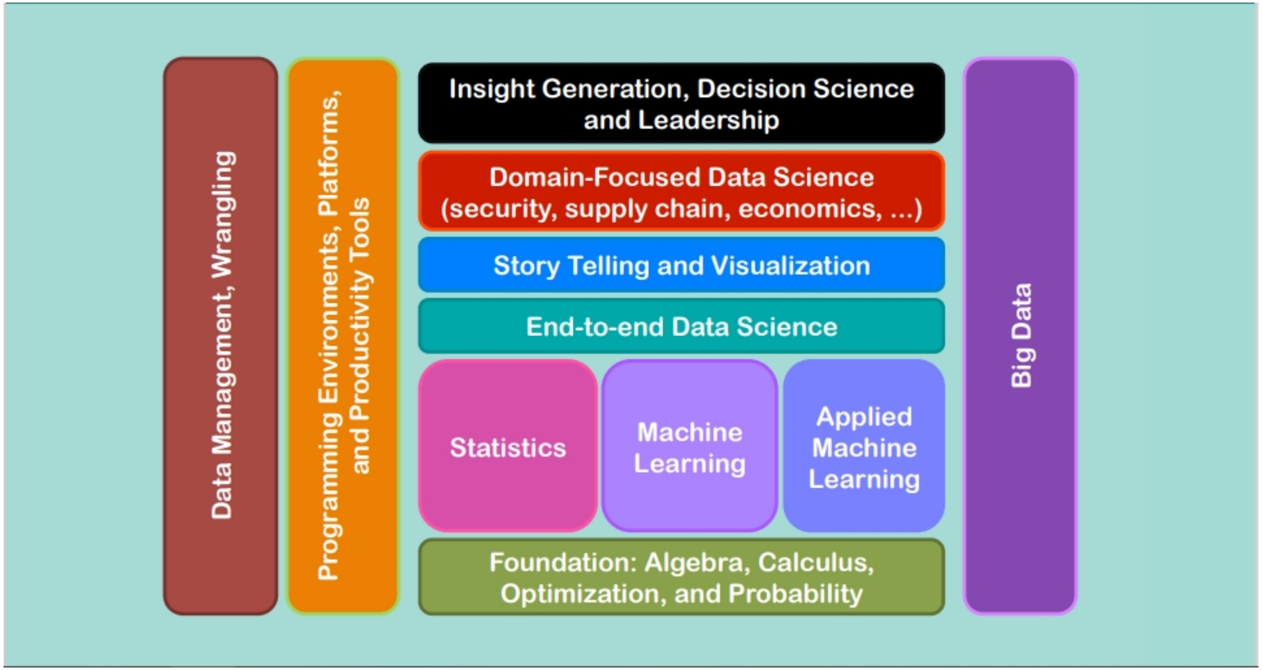
Scaling up: Big data, ML, and cloud computation

- High performance computing (Leavitt cluster)
- Databases: SQL(ite) and BigQuery
- Machine Learning techniques
- Text analysis

What is Data Science?

- **Data science (DS):** The scientific discipline that deals with transforming data into useful information ("insights") using a variety of stats/ML techniques
  - Amazon: Collects data on search history, cart history, purchases
  - Analyzes the data to estimate users' willingness to pay for various products (including Prime); recommend new products
- The rise of data science has come because of the so-called Big Data revolution
  - The rise of the internet in the late-1990s and 2000s $\Rightarrow \uparrow$ opportunities for companies and governments to collect data on consumers & citizens
  - Proliferation of mobile devices & social media from late 2000s until now has generated even more data

Skills required for data science


Data Management, Wrangling
Programming Environments, Platforms, and Productivity Tools
Insight Generation, Decision Science and Leadership
Domain-Focused Data Science (security, supply chain, economics, ...)
Story Telling and Visualization
End-to-end Data Science
Statistics
Machine Learning
Applied Machine Learning
Foundation: Algebra, Calculus, Optimization, and Probability
Big Data

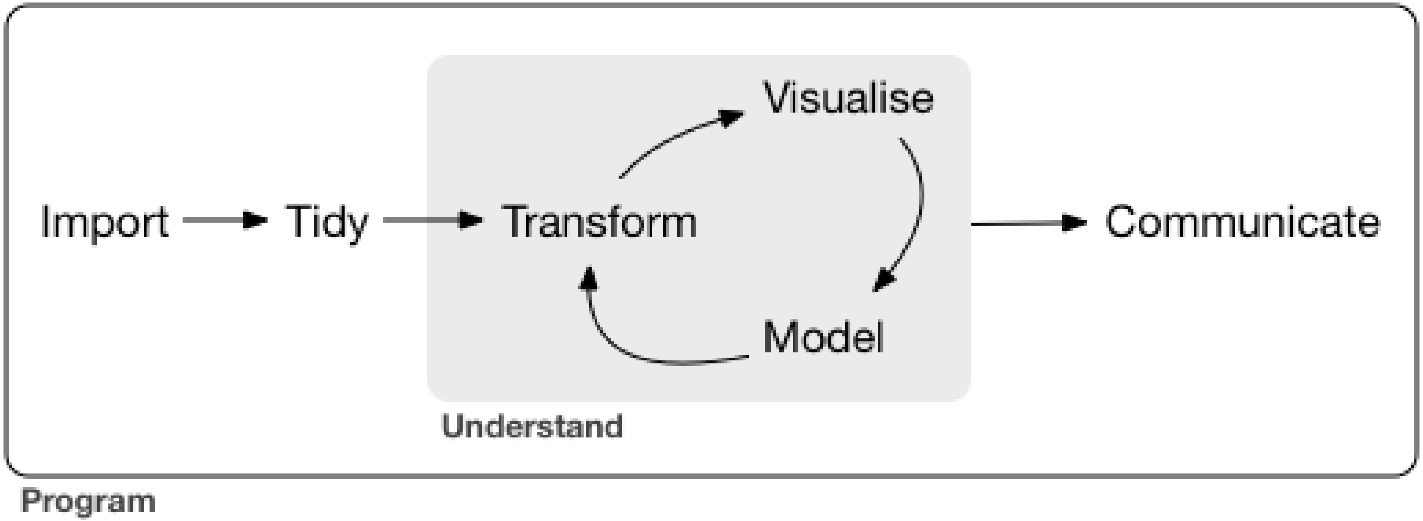
Source: NC State Univ. (p. 26)

Pillars of data science

- Programming (for automation of data collection, manipulation, cleaning, visualization, and modeling)
- Visualization & exploration
- Causal inference (to be able to make a policy prescription)
- Machine learning (to select models, compress data, predict outcomes)

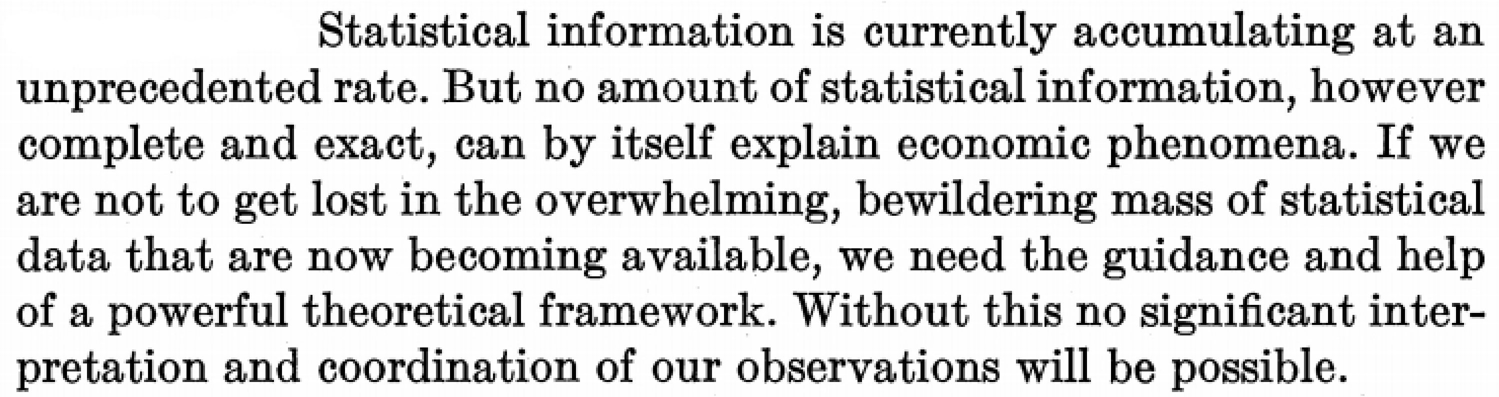
...Assuming one has the appropriate foundation of basic calculus and statistics

The data science workflow



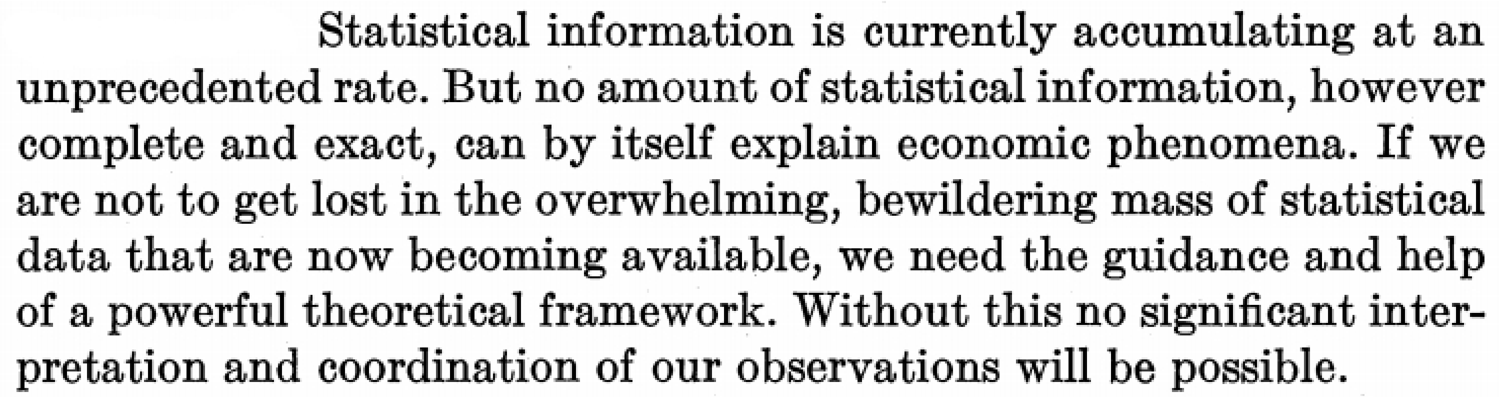
Source: R for Data Science

Big Data



Statistical information is currently accumulating at an unprecedented rate. But no amount of statistical information, however complete and exact, can by itself explain economic phenomena. If we are not to get lost in the overwhelming, bewildering mass of statistical data that are now becoming available, we need the guidance and help of a powerful theoretical framework. Without this no significant interpretation and coordination of our observations will be possible.

Big Data



> Statistical information is currently accumulating at an unprecedented rate. But no amount of statistical information, however complete and exact, can by itself explain economic phenomena. If we are not to get lost in the overwhelming, bewildering mass of statistical data that are now becoming available, we need the guidance and help of a powerful theoretical framework. Without this no significant interpretation and coordination of our observations will be possible.

Source: Frisch, Ragnar. 1933. "Editor's Note" *Econometrica* 1(1): 1--4

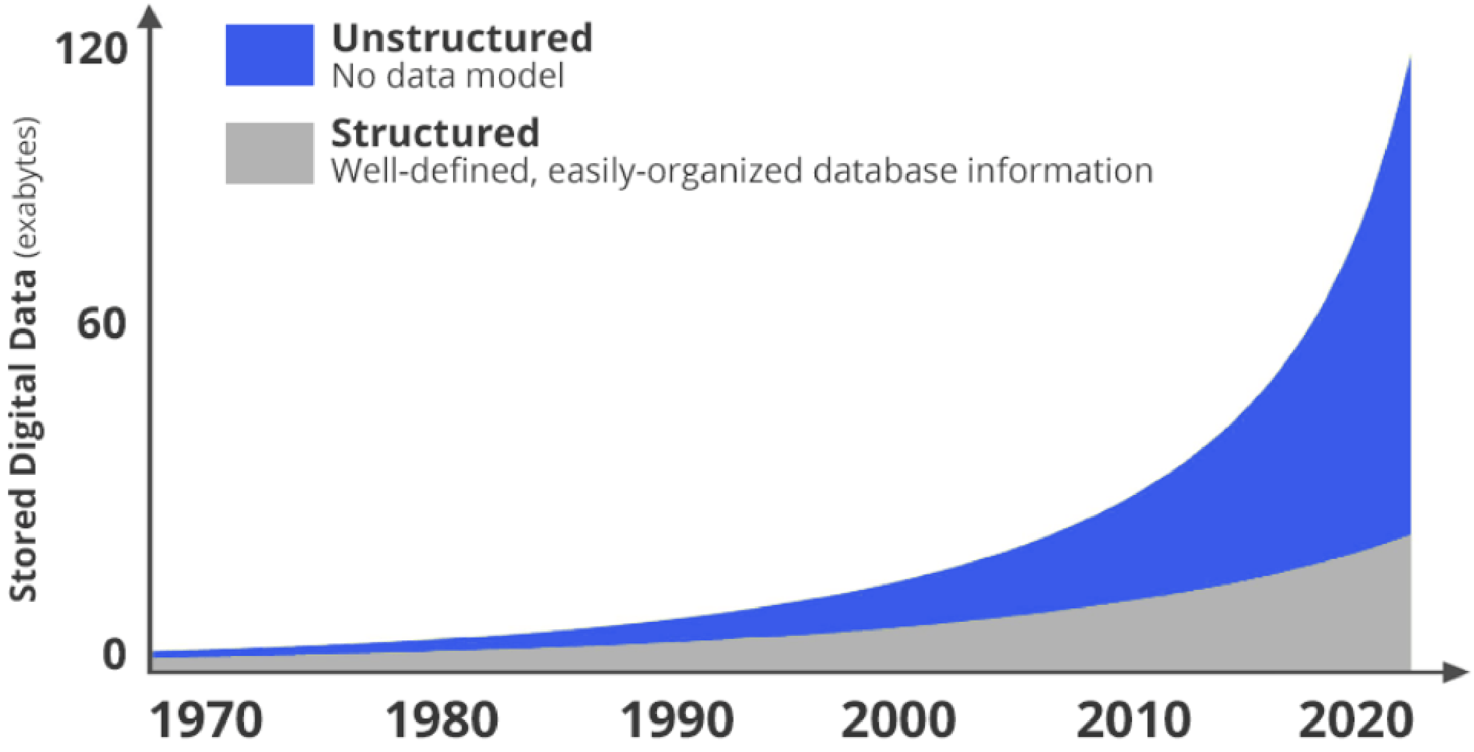
What is Big Data?

It depends on who you ask. It could mean:

1. "Wild" data (unstructured; happenstance; collected without a particular intention; e.g. twitter, contrast with Census surveys)
2. "Wide" data (a.k.a. "Big-K" data because $K > N$, customer data sets where each click is a variable)
3. "Long" data (a.k.a. "Big-N" data because N very, very large [and may not all fit onto a single hard drive!], government tax records, Medicare claims data, etc.)
4. Any data set that cannot be analyzed with classical methods like OLS (e.g. all combinations of the above three types)

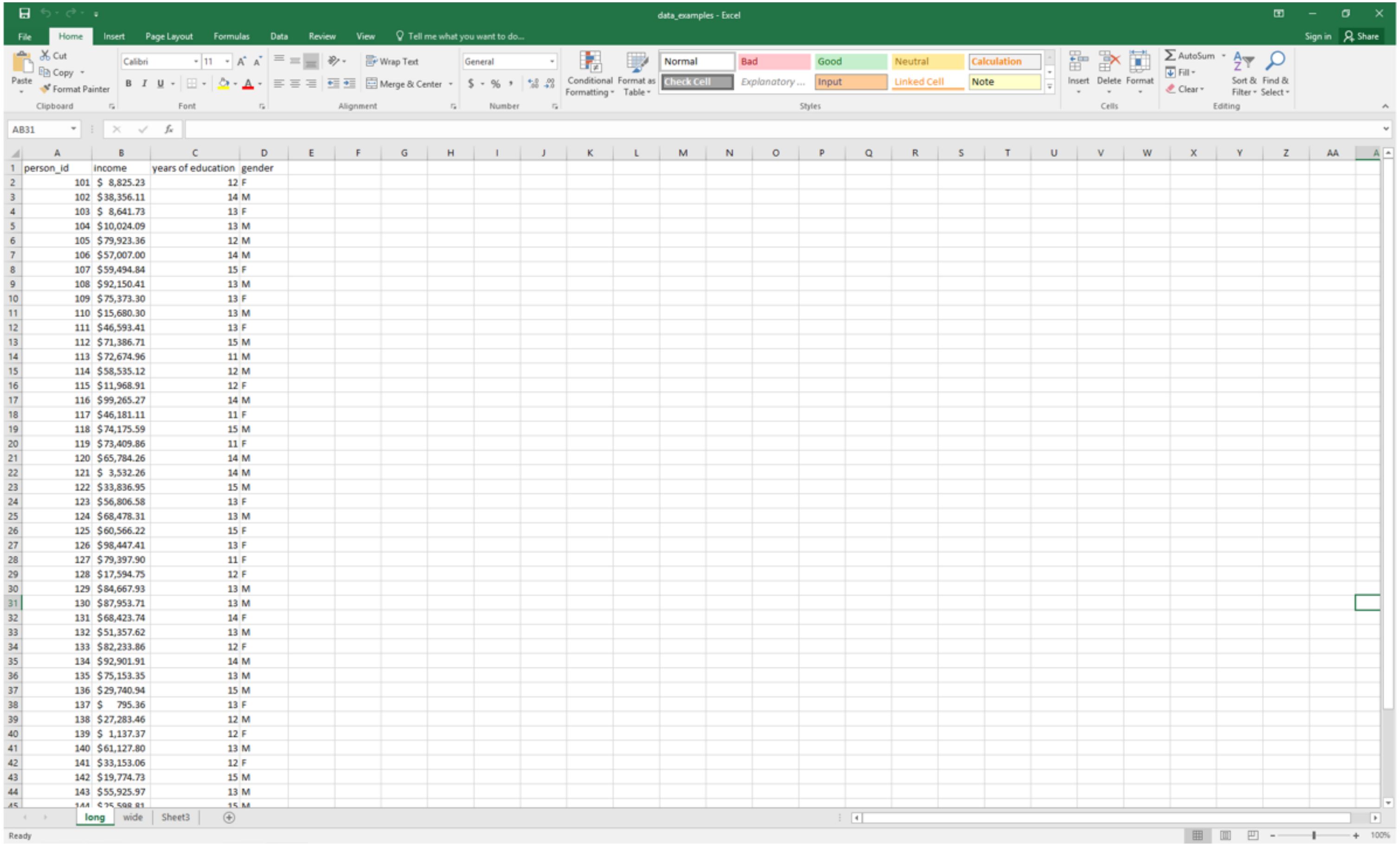
"Big Data" not so much about size of data, but about whether or not "small data" (read: classical) methods can be used

Unstructured data boom



Source: IDC

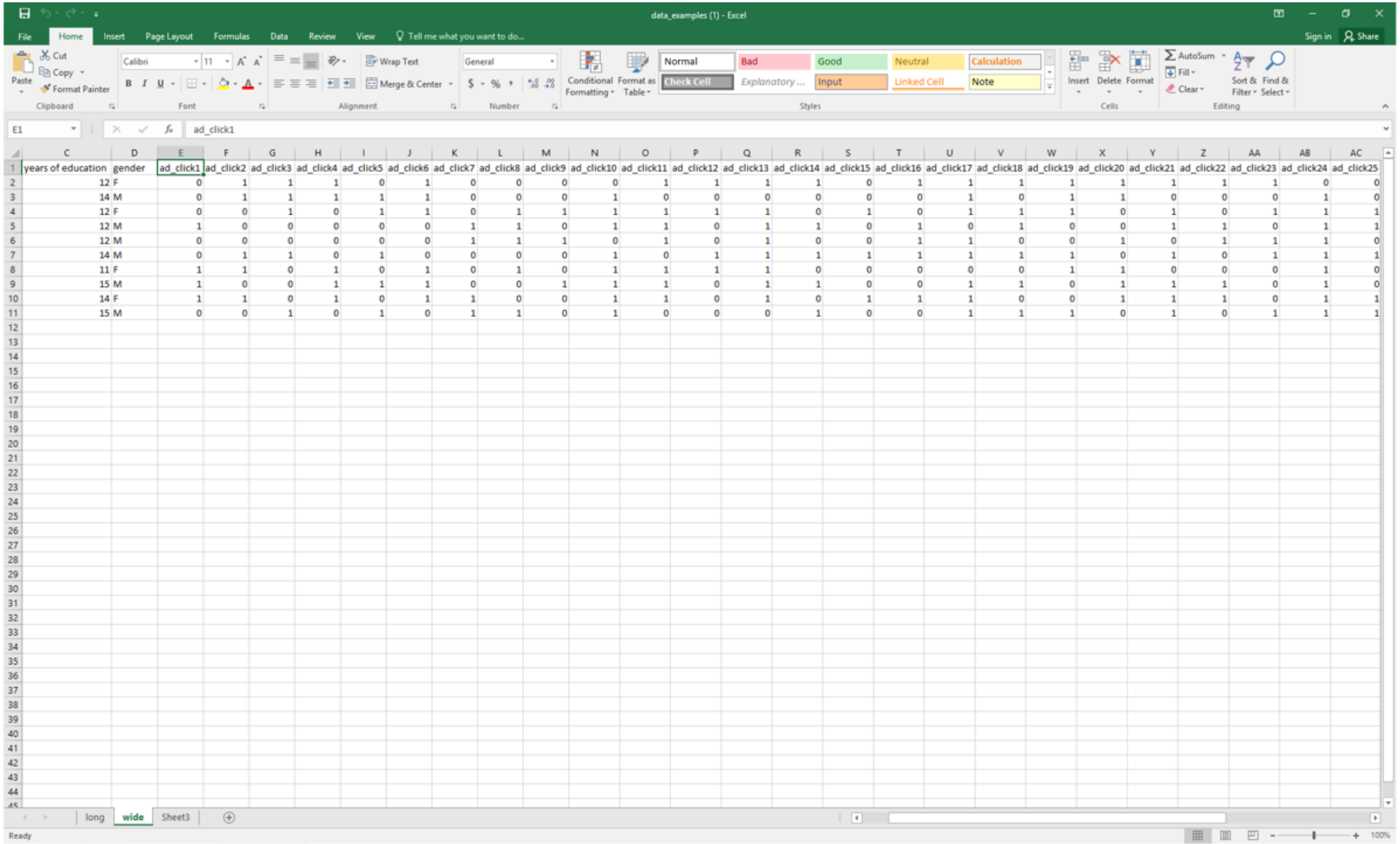
Long data



person_id	income	years of education	gender
101	$ 8,825.23	12	F
102	$38,356.11	14	M
103	$ 8,641.73	13	F
104	$10,024.09	13	M
105	$79,923.36	12	M
106	$57,007.00	14	M
107	$59,494.84	15	F
108	$92,150.41	13	M
109	$75,373.30	13	F
110	$15,680.30	13	M
111	$46,593.41	13	F
112	$71,386.71	15	M
113	$72,674.96	11	M
114	$58,535.12	12	M
115	$11,968.91	12	F
116	$99,265.27	14	M
117	$46,181.11	11	F
118	$74,175.59	15	M
119	$73,409.86	11	F
120	$65,784.26	14	M
121	$ 3,532.26	14	M
122	$33,836.95	15	M
123	$56,806.58	13	F
124	$68,478.31	13	M
125	$60,566.22	15	F
126	$98,447.41	13	F
127	$79,397.90	11	F
128	$17,594.75	12	F
129	$84,667.93	13	M
130	$87,953.71	13	M
131	$68,423.74	14	F
132	$51,357.62	13	M
133	$82,233.86	12	F
134	$92,901.91	14	M
135	$75,153.35	13	M
136	$29,740.94	15	M
137	$ 795.36	13	F
138	$27,283.46	12	M
139	$ 1,137.37	12	F
140	$61,127.80	13	M
141	$33,153.06	12	F
142	$19,774.73	15	M
143	$55,925.97	13	M
144	$25,598.81	15	M

- Main application: *identifying causal effects*
- Example: effects of improving schools on income

Wide data



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	years of education	gender	ad_click1	ad_click2	ad_click3	ad_click4	ad_click5	ad_click6	ad_click7	ad_click8	ad_click9	ad_click10	ad_click11	ad_click12	ad_click13	ad_click14	ad_click15	ad_click16	ad_click17	ad_click18	ad_click19	ad_click20	ad_click21	ad_click22	ad_click23	ad_click24	ad_click25
2	12	F	0	1	1	1	0	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0
3	14	M	0	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0
4	12	F	0	0	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1
5	12	M	1	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	0	1	1
6	12	M	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	1	1	1	0
7	14	M	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1
8	11	F	1	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0
9	15	M	1	0	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	0
10	14	F	1	1	0	1	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	1
11	15	M	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	1	0	1	0	1	1	1

- Main application: *prediction*
- Example: predicting income to target ads

What is machine learning? What is AI?

- **Machine learning (ML):** Allowing computers to learn for themselves without explicitly being programmed
  - USPS: Computer to read handwriting on envelopes
  - Google: AlphaGo, computer that defeated world champion Go player
  - Apple/Amazon/Microsoft: Siri, Alexa, Cortana, Talon voice assistants
- **Artificial intelligence (AI):** Constructing machines (robots, computers) to think and act like human beings
- ML is a subset of AI

Big data & machine learning

- You'll often hear the phrase "big data and machine learning"
- This is because many machine learning algorithms are helpful for big data problems:
    - Selecting which $k < K$ covariates should enter your model
    - Streamlined techniques for processing "wild" data
    - New modeling approaches that can leverage the greater amount of information that Big Data has

ChatGPT and GitHub CoPilot

- I strongly encourage you to use ChatGPT and GitHub CoPilot in all aspects of this class
- ChatGPT is a large language model (LLM) developed by OpenAI
- Create an account at http://chat.openai.com/
- ChatGPT is excellent at producing basic code and helping resolve bugs
- It takes most of the headache away from coding, leaving only the "fun" parts

ChatGPT example

Human:

Write a SQL script that does the following:

(a) Read in the Florida insurance data CSV file

(b) Print out the first 10 rows of the data set

(c) List which counties are in the sample (i.e. list unique values of the county variable)

(d) Compute the average property appreciation from 2011 to 2012 (i.e. compute the mean of tiv_2012 - tiv_2011)

(e) Create a frequency table of the construction variable to see what fraction of buildings are made out of wood or some other material

ChatGPT example (continued)

Assistant:

Assuming the Florida insurance data is stored in a table called "insurance" and the columns are named "county", "tiv_2011", "tiv_2012", and "construction", the SQL script would be as follows:

```sql
-- (a) Read in the Florida insurance data CSV file
CREATE TABLE insurance (county VARCHAR(255), tiv_2011 INT, tiv_2012 INT, construction VARCHAR(255));
LOAD DATA INFILE 'florida_insurance.csv' INTO TABLE insurance
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Note: This is basic SQL script and it may vary based on the database you are using. You may need to make some modification accordingly.

Exported on January 17, 2023 10:11 AM.

AI assistance and accessibility

- I actively used GitHub CoPilot to write these Rmarkdown slides
- I used a voice-to-text tool (Talon) to dictate text and comments while in a sling
- Asked CoPilot to write code for me based on the comments
- Quick demo if time allows

Correlation vs. causation

- Machine learning is not the end-all, be-all of data science
- A good data scientist knows that correlation is not causation!
- ML is good at prediction, but not necessarily at causal inference
- More on this later! First, let's get into software

Getting started

Software installation and registration

1. Download R.
2. Download RStudio.
3. Download Visual Studio Code.
4. Download Git.
5. Create an account on GitHub and register for a student/educator discount.
    - You will soon receive an invitation to the semester-specific course org. on GitHub, as well as GitHub classroom, which is how we'll disseminate and submit assignments, receive feedback and grading, etc.

If you had trouble completing any of these steps, please raise your hand.

- My go-to place for installation guidance and troubleshooting is Jenny Bryan's http://happygitwithr.com.

Some OS-specific extras

I'll detail further software requirements as and when the need arises. However, to help smooth some software installation issues further down the road, please also do the following (depending on your OS):

- **Windows:** Install Rtools. I also recommend that you install Chocolately. Windows Subsystem for Linx.
- **Mac:** Install Homebrew. I also recommend that you configure/open your C++ toolchain (see here.)
- **Linux:** None (you should be good to go).

Checklist

☑ Do you have the most recent version of R?

```
version$version.string
```

```
## [1] "R version 4.3.1 (2023-06-16 ucrt)"
```

☑ Do you have the most recent version of RStudio? (The preview version is fine.)

```
RStudio.Version()$version
## Requires an interactive session but should return something like "[1] ‘1.4.1100’"
```

☑ Have you updated all of your R packages?

```
update.packages(ask = FALSE, checkBuilt = TRUE)
```

Checklist (cont.)

Open up the shell.

- Windows users, make sure that you installed a Bash-compatible version of the shell. If you installed Git for Windows, then you should be good to go.

☑ Which version of Git have you installed?

```
git --version
```

```
## git version 2.34.1
```

☑ Did you introduce yourself to Git? (Substitute in your details.)

```
git config --global user.name 'Kyle Coombs'
git config --global user.email 'kcoombs@bates.edu'
git config --global --list
```

☑ Did you register an account in GitHub?

Checklist (cont.)

We will make sure that everything is working properly with your R, VS Code, and GitHub setup next lecture.

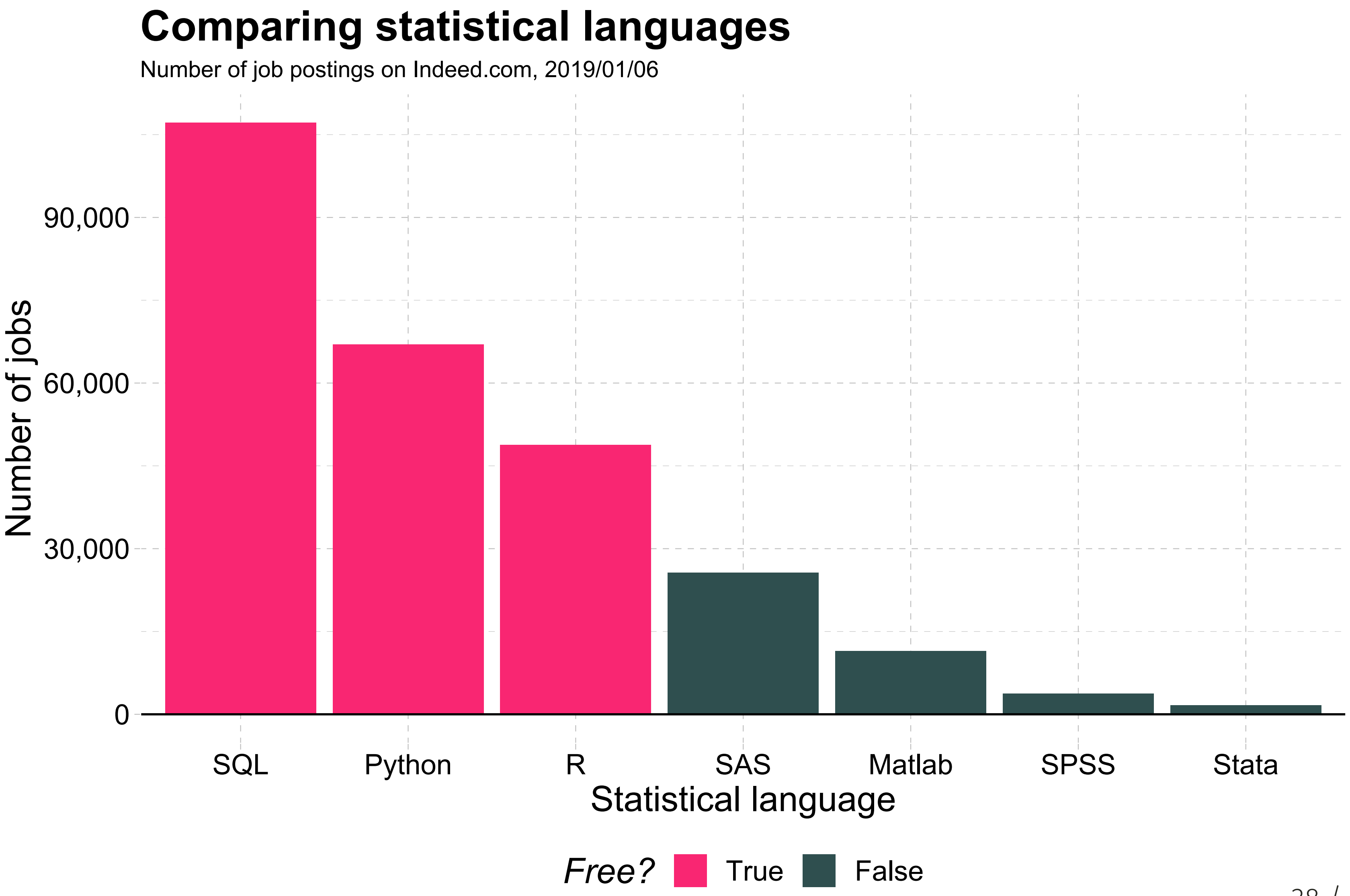
For the rest of today's lecture, I want to go over some very basic R concepts.

PS — Just so you know where we're headed: We'll return to these R concepts (and delve much deeper) next week after a brief, but important detour to the lands of Git(Hub) and coding best practices.

- Don't worry, it will all make sense. You'll see.

R for data science

Why R and RStudio? (cont.)



Why R and RStudio? (cont.)

Data science positivism

- Alongside Python, R has become the *de facto* language for data science.
    - See: *The Impressive Growth of R*, *The Popularity of Data Science Software*
- Open-source (free!) with a global user-base spanning academia and industry.
    - "Do you want to be a profit source or a cost center?"

Bridge to applied economics and other tools

- Already has all of the statistics and econometrics support, and is amazingly adaptable as a “glue” language to other programming languages and APIs.
- The RStudio IDE and ecosystem allow for further, seemless integration.

Path dependency

- It's also the language that I know (second) best.
- (Learning multiple languages is a good idea, though.)

Some R basics

1. Everything is an object.

2. Everything has a name.

3. You do things using functions.

4. Functions come pre-written in packages (i.e. "libraries"), although you can — and should — write your own functions too.

Points 1. and 2. can be summarised as an object-orientated programming (OOP) approach.

- This may sound super abstract now, but we'll see *lots* of examples over the coming weeks that will make things clear.

R vs Stata

If you're coming from Stata, some additional things worth emphasizing:

- Multiple objects (e.g. data frames) can exist happily in the same workspace.
  - No more `keep`, `preserve`, `restore` hackery. (Though, props to Stata 16.)
  - This is a direct consequence of the OOP approach.
- You will load packages at the start of every new R session. Make peace with this.
  - "Base" R comes with tons of useful in-built functions. It also provides all the tools necessary for you to write your own functions.
  - However, many of R's best data science functions and tools come from external packages written by other users.
- R easily and infinitely parallelizes. For free.
  - Compare the cost of a Stata/MP license, nevermind the fact that you effectively pay per core...
- You don't need to `tsset or xtset` your data. (Although you can too.)

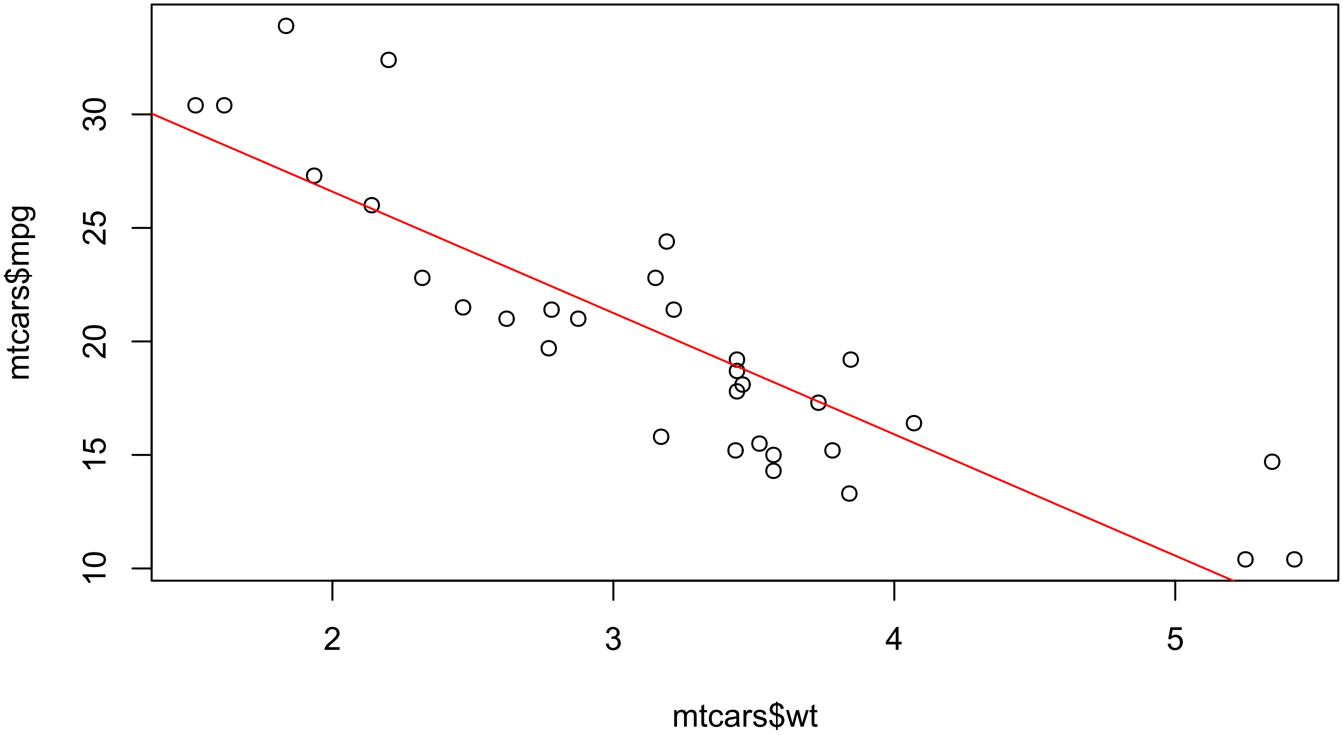
R code example (linear regression)

```r
fit = lm(mpg ~ wt, data = mtcars)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

Base R plot

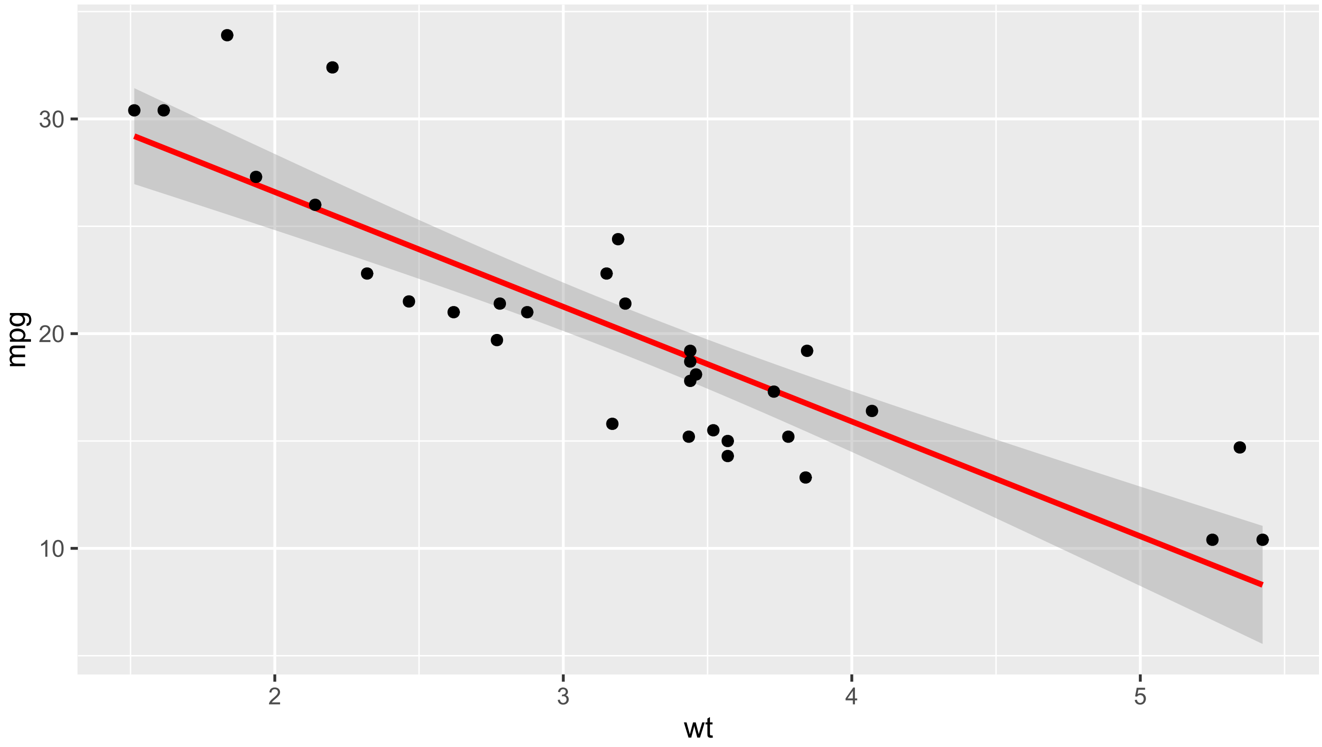
```
par(mar = c(4, 4, 1, .1)) ## Just for nice plot margins on this slide deck
plot(mtcars$wt, mtcars$mpg)
abline(fit, col = "red")
```



ggplot2

```
library(ggplot2)
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_smooth(method = "lm", col = "red") +
  geom_point()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Do increases in GDP cause life

- Let's use our new-found R knowledge to try to separate correlation from causation for a critical question in economics:
    - Does increasing the economic pie (GDP) lead to longer lives (life expectancy)?
- We can use the gapminder dataset to explore this question
- The gapminder dataset contains panel data on life expectancy, population size, and GDP per capita for 142 countries since the 1950s
- Any predictions about what we'll learn?

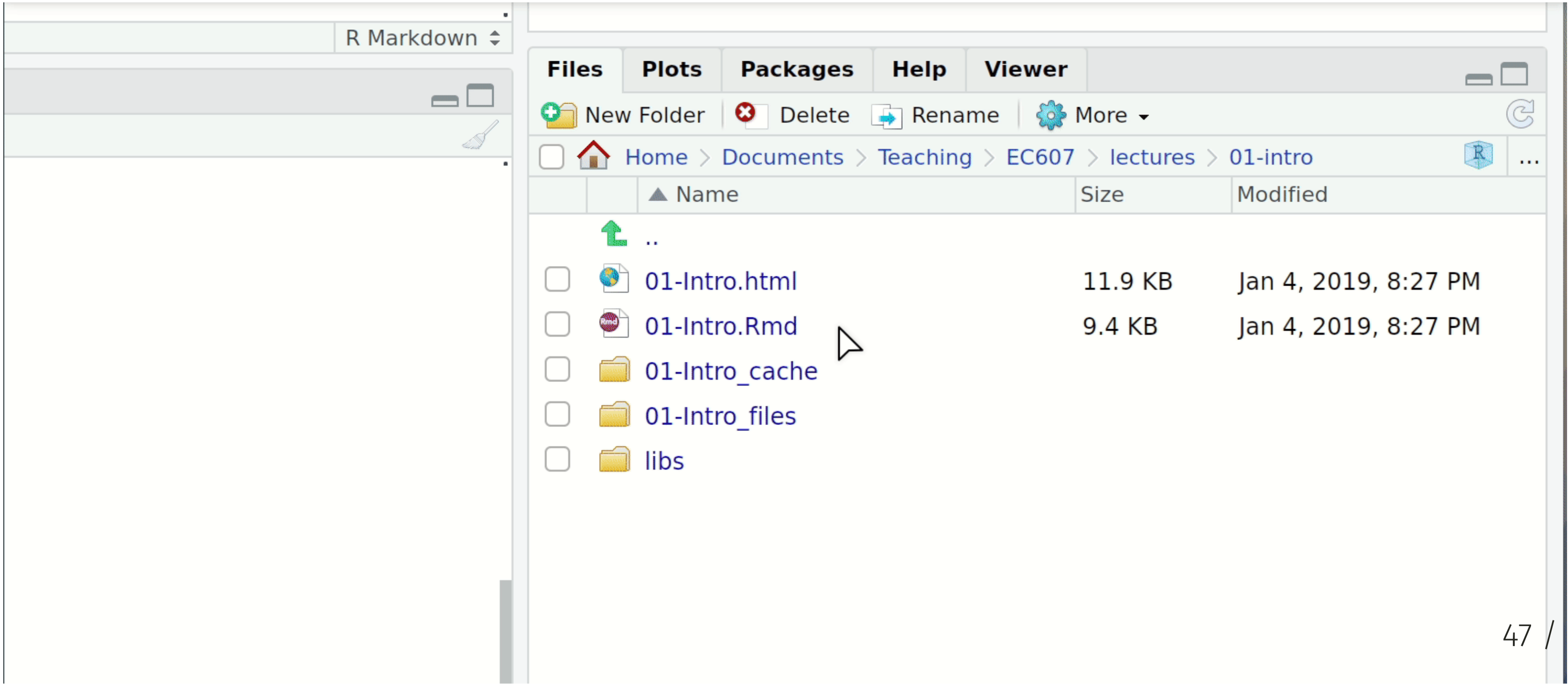
More ggplot2

Install and load

Open up your laptops. For the remainder of this first lecture, we're going to pla around with ggplot2 (i.e. livecoding) to answer our question about GDP and life expectancy.

If you don't have them already, install the `ggplot2` and `causaldata` packages via either:

- **Console:** Enter `install.packages(c("ggplot2", "causaldata"), dependencies=T)`.
- **RStudio:** Click the "Packages" tab in the bottom-right window pane. Then click "Install" and search for these two packages.



Name	Size	Modified
..		
01-Intro.html	11.9 KB	Jan 4, 2019, 8:27 PM
01-Intro.Rmd	9.4 KB	Jan 4, 2019, 8:27 PM
01-Intro_cache		
01-Intro_files		
libs		

Install and load (cont.)

Once the packages are installed, load them into your R session with the `library()` function.

```
library(ggplot2)
library(causaldata) ## We're just using this package for the gapminder data, it has many useful datasets!
```

```
##
## Attaching package: 'causaldata'

## The following object is masked from 'package:gapminder':
##
##     gapminder
```

Notice too that you don't need quotes around the package names any more. Reason: R now recognises these packages as defined objects with given names. ("Everything in R is an object and everything has a name.")

PS — A convenient way to combine the package installation and loading steps is with the pacman package's `p_load()` function. If you run `pacman::p_load(ggplot, causaldata)` it will first look to see whether it needs to install either package before loading them. Clever.

- We'll get to this next week, but if you want to run a function from an (installed) package without loading it, you can use the `PACKAGE::package_function()` syntax.

Brief aside: The gapminder dataset

Because we're going to be plotting the gapminder dataset, it is helpful to know that it contains panel data on life expectancy, population size, and GDP per capita for 142 countries since the 1950s.

```
gapminder
```

```
## # A tibble: 1,704 × 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ℹ 1,694 more rows
```

When opening a new dataset, it is important to know why. What question are we asking?

Today we want to know:

- Does increased GDP per capita lead to increased life expectancy?
- If not, what can we say about the two?

Elements of ggplot2

Hadley Wickham's ggplot2 is one of the most popular packages in the entire R canon.

- It also happens to be built upon some deep visualization theory: i.e. Leland Wilkinson's *The Grammar of Graphics*.

There's a lot to say about ggplot2's implementation of this "grammar of graphics" approach, but the three key elements are:

1. Your plot ("the visualization") is linked to your variables ("the data") through various **aesthetic mappings**.

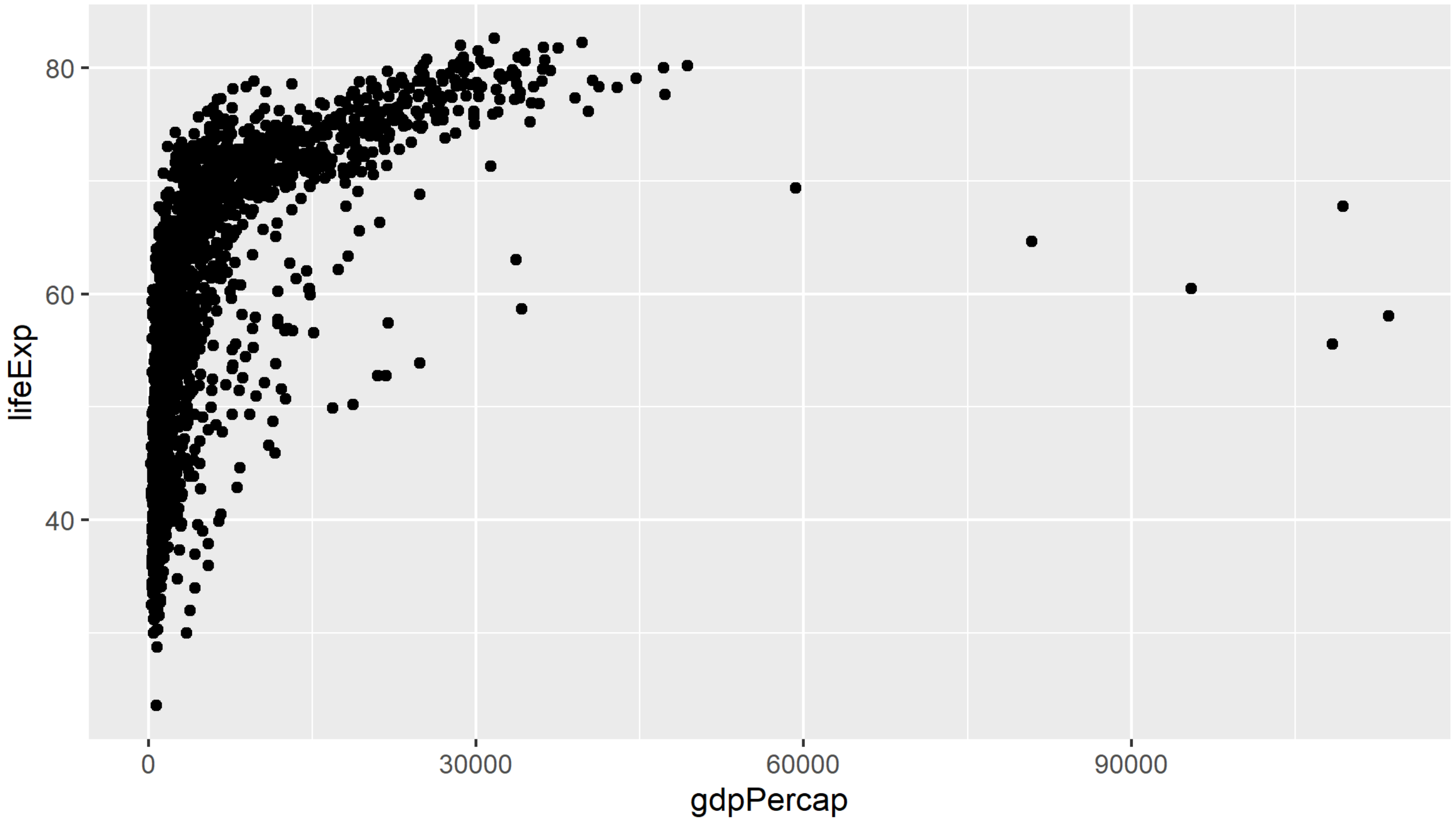
2. Once the aesthetic mappings are defined, you can represent your data in different ways by choosing different **geoms** (i.e. "geometric objects" like points, lines or bars).

3. You build your plot in **layers**.

That's kind of abstract. Let's review each element in turn with some actual plots.

1. Aesthetic mappings

```
ggplot(data = gapminder, mapping = aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



1. Aesthetic mappings (cont.)

```r
ggplot(data = gapminder, mapping = aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```

Focus on the top line, which contains the initialising `ggplot()` function call. This function accepts various arguments, including:

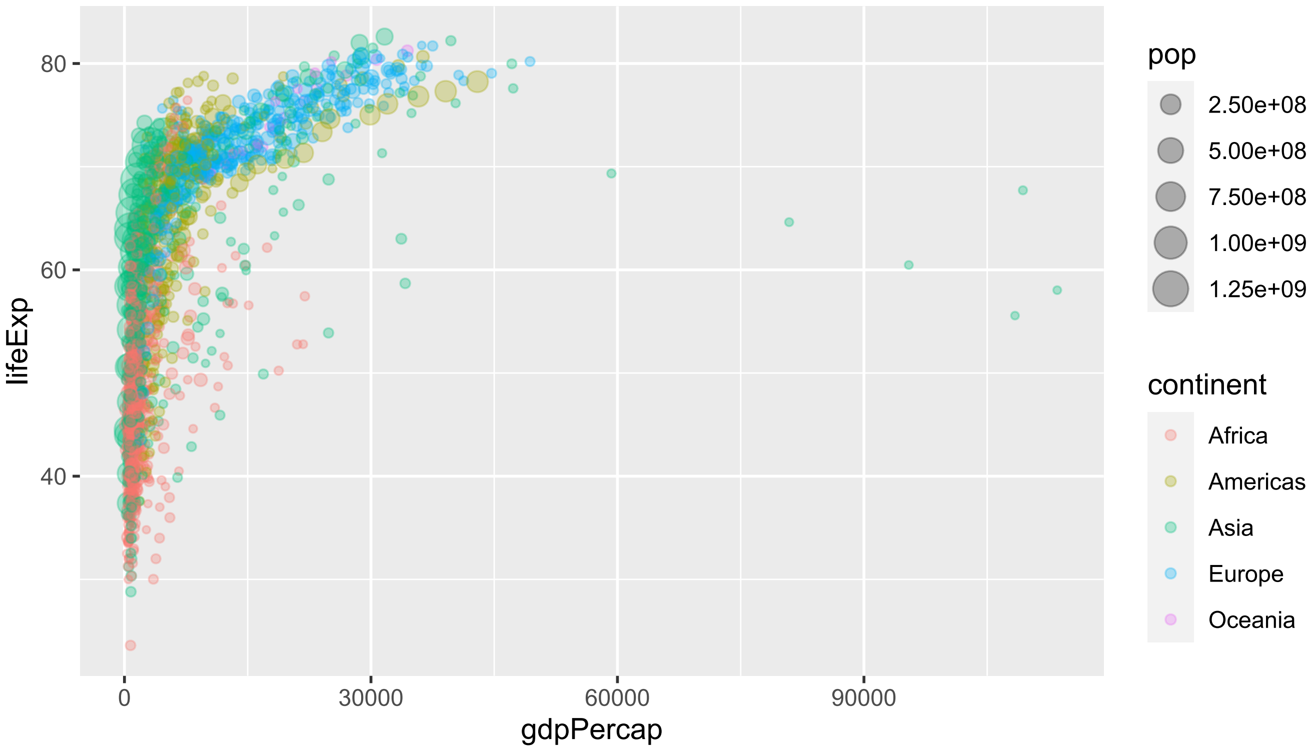
- Where the data come from (i.e. `data = gapminder`).
- What the aesthetic mappings are (i.e. `mapping = aes(x = gdpPercap, y = lifeExp)`).

The aesthetic mappings here are pretty simple: They just define an x-axis (GDP per capita) and a y-axis (life expecancy).

- To get a sense of the power and flexibility that comes with this approach, however, consider what happens if we add more aesthetics to the plot call...

1. Aesthetic mappings (cont.)

```r
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp, size = pop, col = continent)) +
  geom_point(alpha = 0.3) ## "alpha" controls transparency. Takes a value between 0 and 1.
```

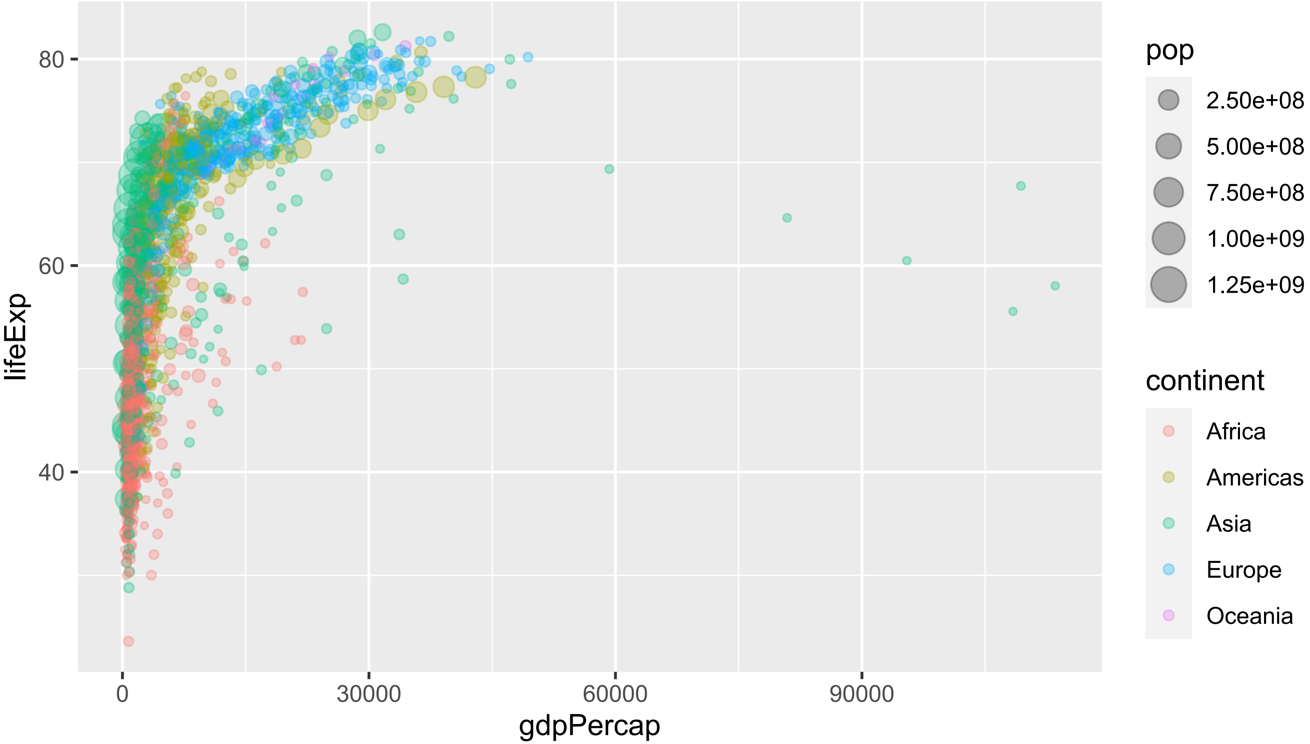


Note that I've dropped the "mapping =" part of the ggplot call. Most people just start with "aes(...)", since `ggplot2` knows the order of the arguments.

1. Aesthetic mappings (cont.)

We can specify aesthetic mappings in the geom layer too.

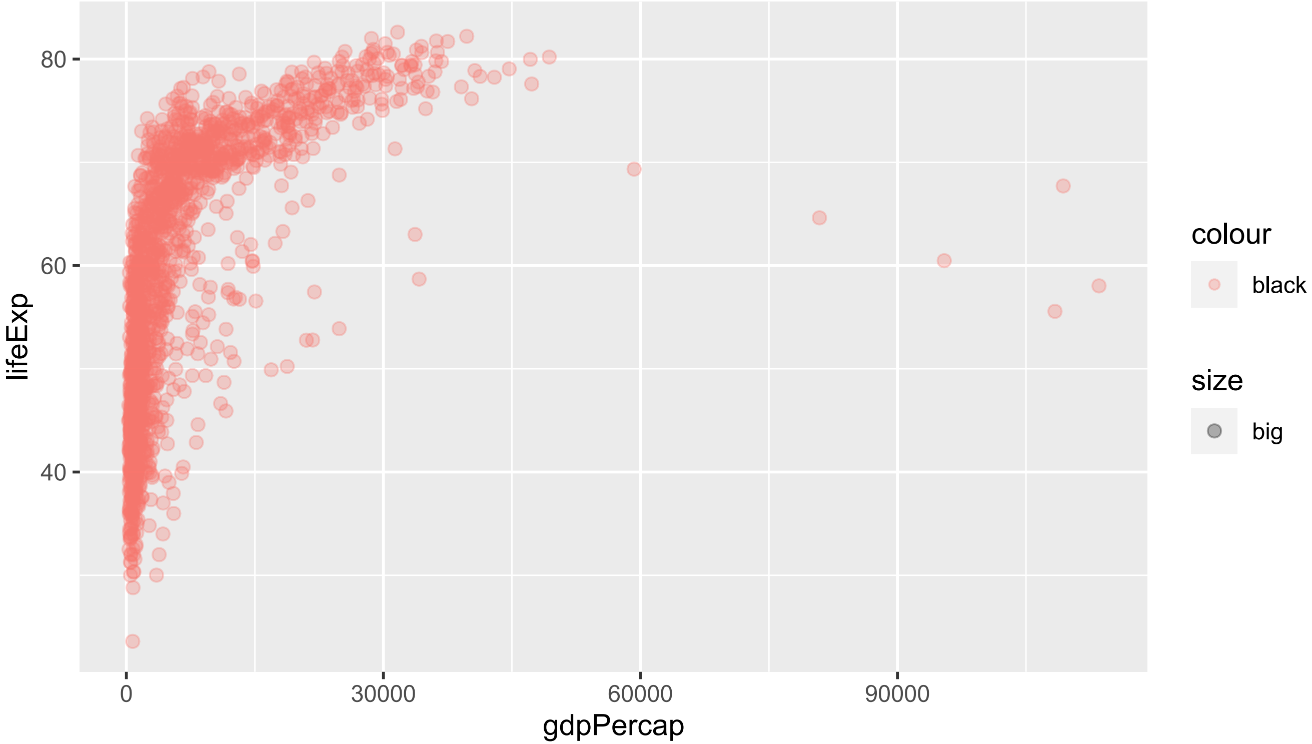
```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) + ## Applicable to all geoms
  geom_point(aes(size = pop, col = continent), alpha = 0.3) ## Applicable to this geom only
```



1. Aesthetic mappings (cont.)

Oops. What went wrong here?

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(size = "big", col="black"), alpha = 0.3)
```

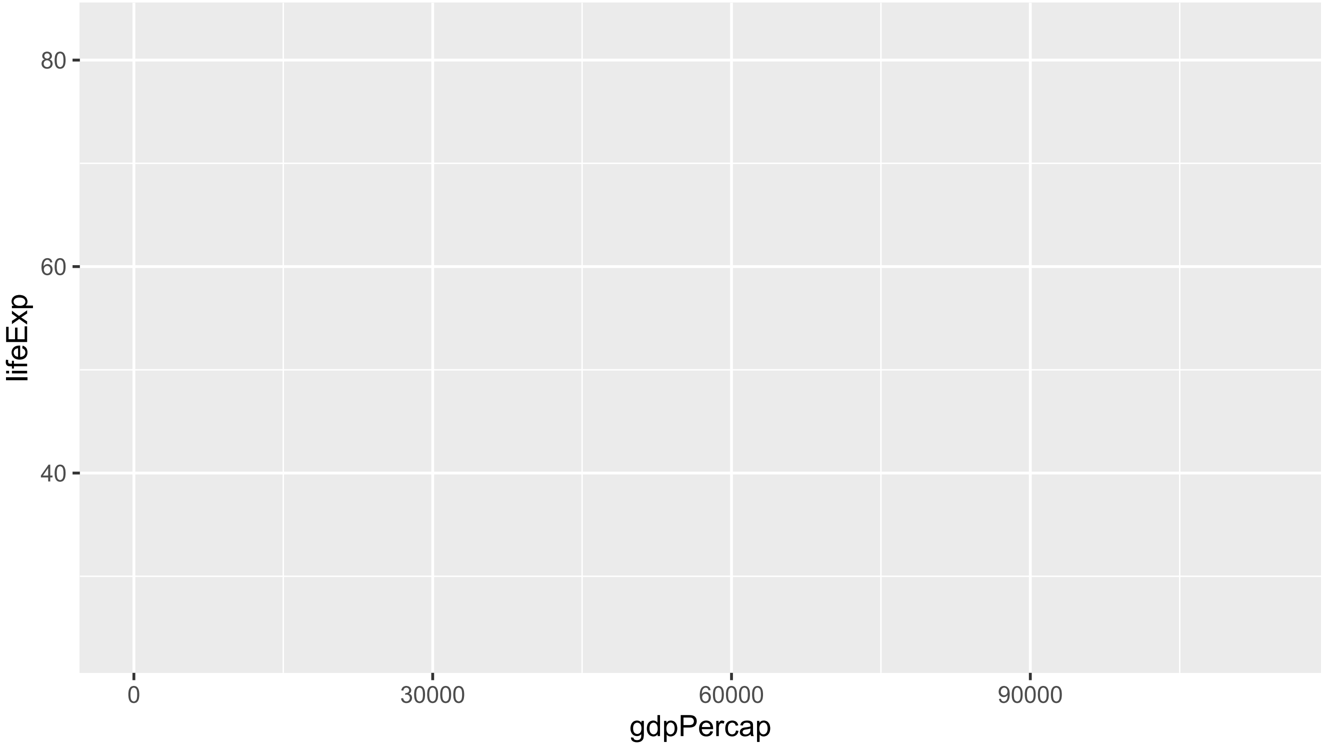


Answer: Aesthetics must be mapped to variables, not descriptions!

1. Aesthetic mappings (cont.)

At this point, instead of repeating the same ggplot2 call every time, it will prove convenient to define an intermediate plot object that we can re-use.

```
p = ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp))
p
```

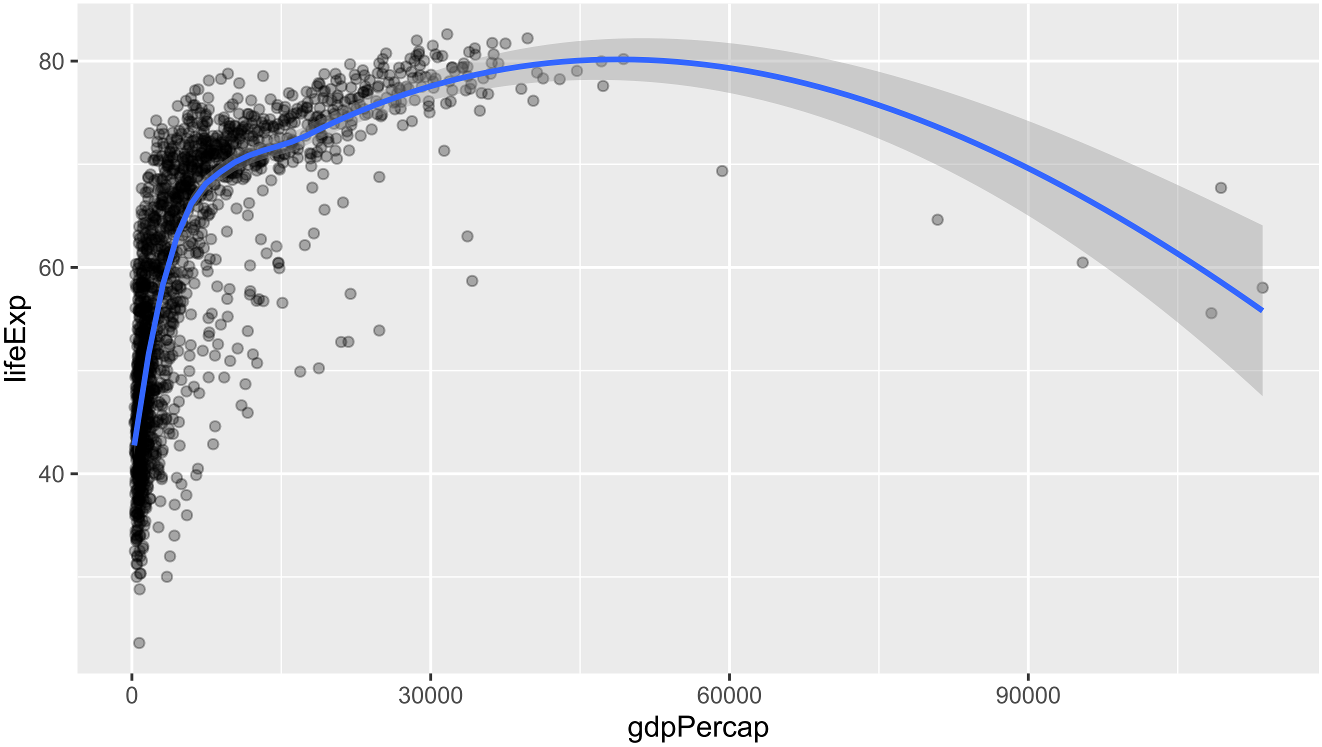


2. Geoms

Once your variable relationships have been defined by the aesthetic mappings, you can invoke and combine different geoms to generate different visulaizations.

```
p +
  geom_point(alpha = 0.3)  +
  geom_smooth(method = "loess")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Check back on question

Does increased GDP per capita lead to increased life expectancy?

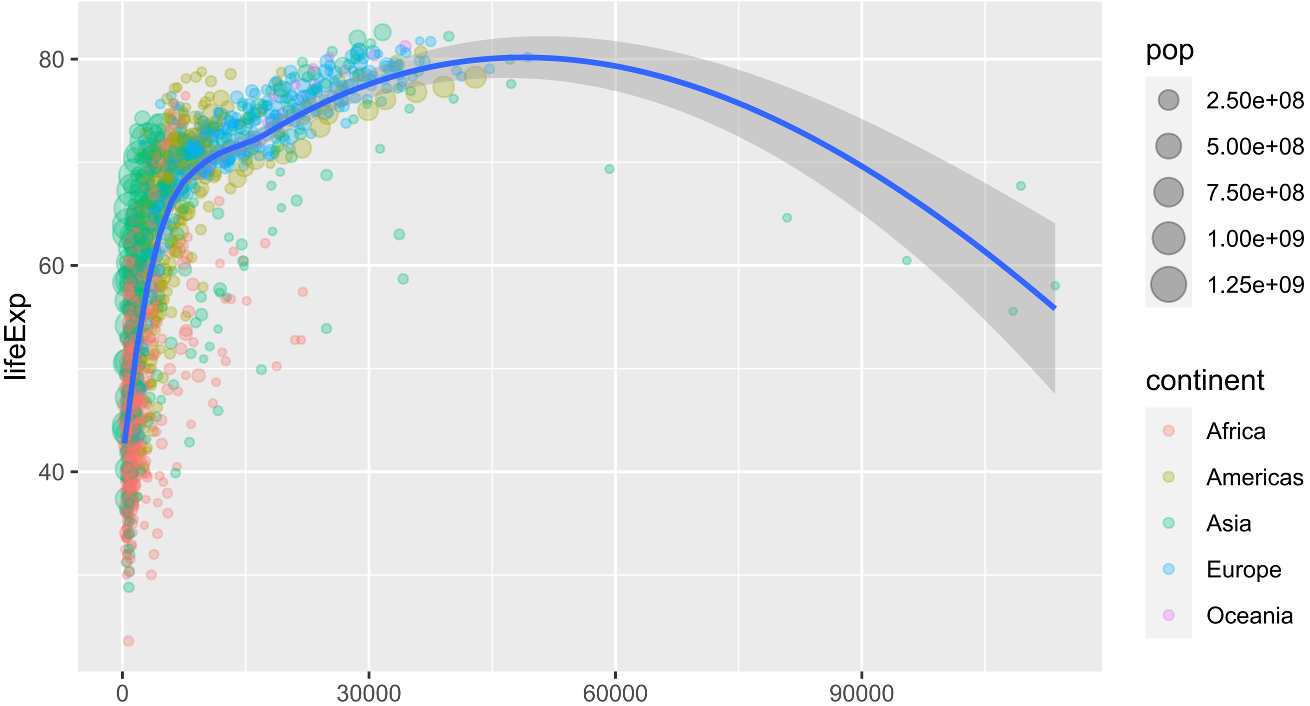
- There is a positive relationship between GDP per capita and life expectancy
- Is it causal? Unclear.
- What other factors might be at play?
- Can we rule those out?
- What about time?

2. Geoms (cont.): account for population

Aesthetics can be applied differentially across geoms.

```
p +
  geom_point(aes(size = pop, col = continent), alpha = 0.3)  +
  geom_smooth(method = "loess")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



2. Geoms (cont.)

The previous plot provides a good illustration of the power (or effect) that comes from assigning aesthetic mappings "globally" vs in the individual geom layers.

- Compare: What happens if you run the below code chunk?

```r
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp, size = pop, col = continent)) +
  geom_point(alpha = 0.3)  +
  geom_smooth(method = "loess")
```

2. Geoms (cont.)

Similarly, note that some geoms only accept a subset of mappings. E.g. `geom_density()` doesn't know what to do with the "y" aesthetic mapping.

```r
p + geom_density()
```

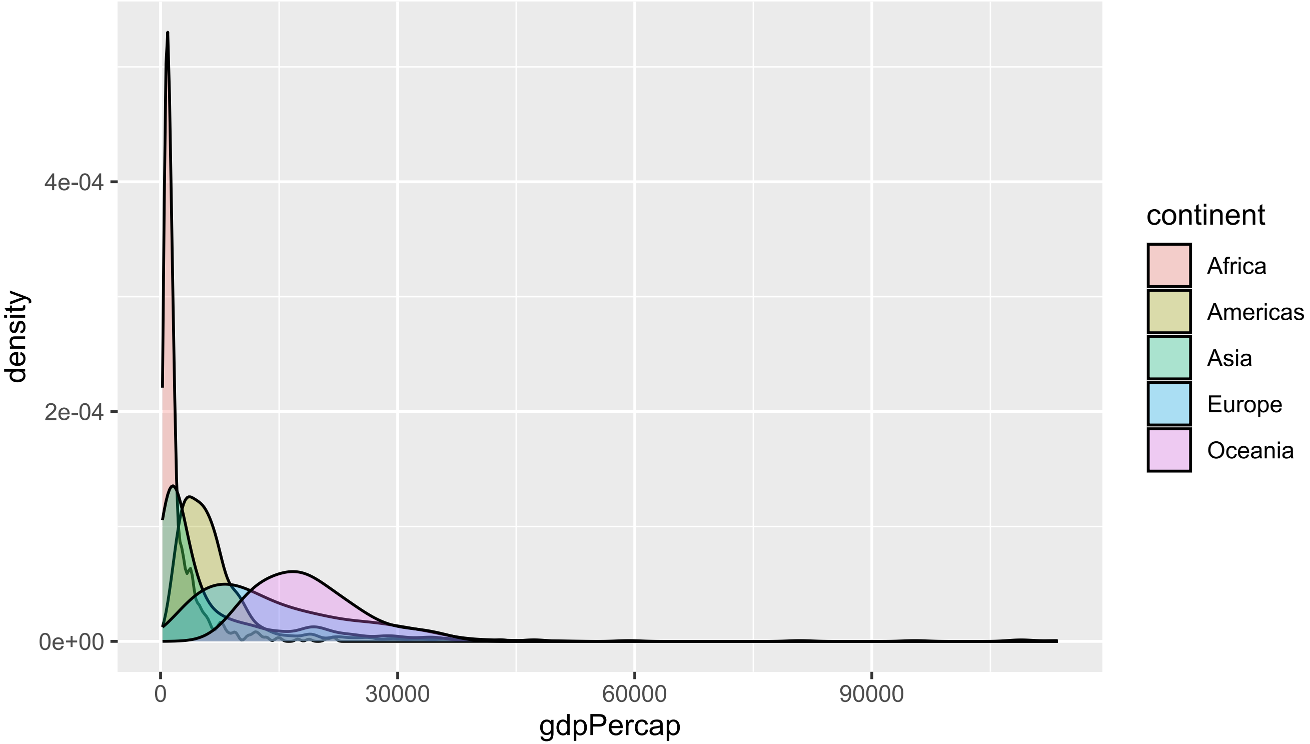
```
## Warning: The following aesthetics were dropped during statistical transformation: y
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## Error in `geom_density()`:
## ! Problem while setting up geom.
## ℹ Error occurred in the 1st layer.
## Caused by error in `compute_geom_1()`:
## ! `geom_density()` requires the following missing aesthetics: y
```

2. Geoms (cont.)

We can fix that by being more careful about how we build the plot.

```
ggplot(data = gapminder) + ## i.e. No "global" aesthetic mappings"
  geom_density(aes(x = gdpPercap, fill = continent), alpha=0.3)
```



3. Build your plot in layers

We've already seen how we can chain (or "layer") consecutive plot elements using the `+` connector.

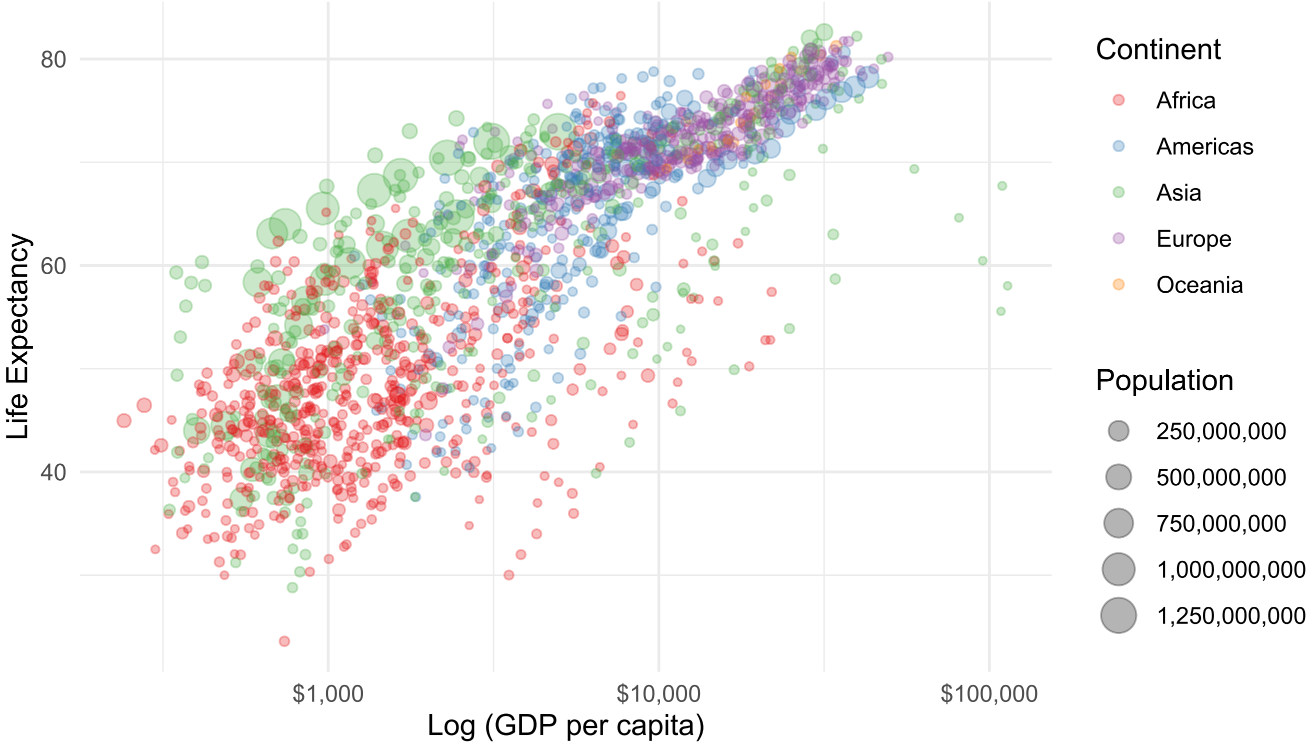
- The fact that we can create and then re-use an intermediate plot object (e.g. "p") is testament to this.

But it bears repeating: You can build out some truly impressive complexity and transformation of your visualization through this simple layering process.

- You don't have to transform your original data; ggplot2 takes care of all of that.
- For example (see next slide for figure).
- **Bonus:** Maybe this will help make sense of the non-linear relationship between GDP per capita and life expectancy?

```r
p2 =
  p +
  geom_point(aes(size = pop, col = continent), alpha = 0.3) +
  scale_color_brewer(name = "Continent", palette = "Set1") + ## Different colour scale
  scale_size(name = "Population", labels = scales::comma) + ## Different point (i.e. legend) scale
  scale_x_log10(labels = scales::dollar) + ## Switch to logarithmic scale on x-axis. Use dollar units.
  labs(x = "Log (GDP per capita)", y = "Life Expectancy") + ## Better axis titles
  theme_minimal() ## Try a minimal (b&w) plot theme
```

3. Build your plot in layers (cont.)



What else?

We have barely scratched the surface of ggplot2's functionality... let alone talked about the entire ecosystem of packages that has been built around it.

- Here's are two quick additional examples to whet your appetite

Note that you will need to install and load some additional packages if you want to recreate the next two figures on your own machine. A quick way to do this:

```r
if (!require("pacman")) install.packages("pacman")
```

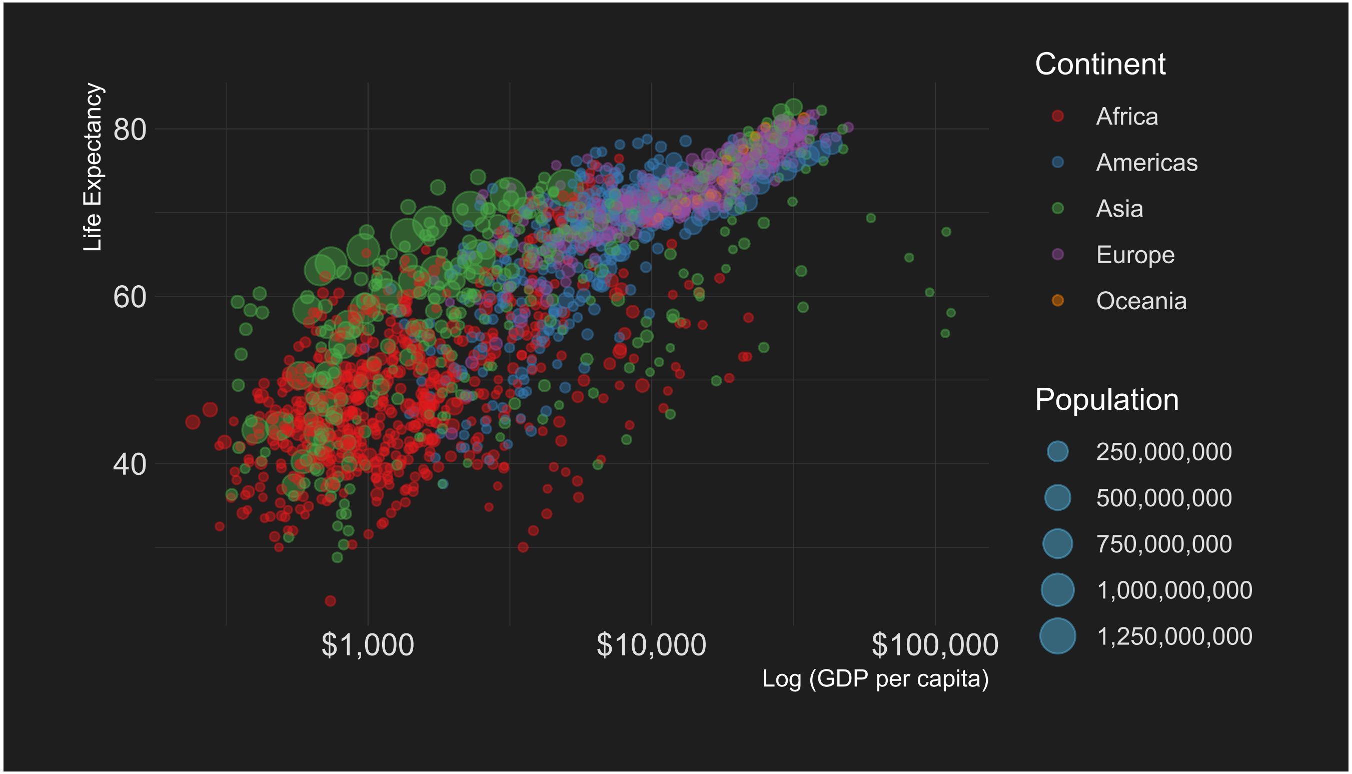
```
## Loading required package: pacman
```

```r
pacman::p_load(hrbrthemes, gganimate)
```

What else? (cont.)

Simple extension: Use an external package theme.

```r
# library(hrbrthemes)
p2 + theme_modern_rc() + geom_point(aes(size = pop, col = continent), alpha = 0.2)
```



Check back on question

Does increased GDP per capita lead to increased life expectancy?

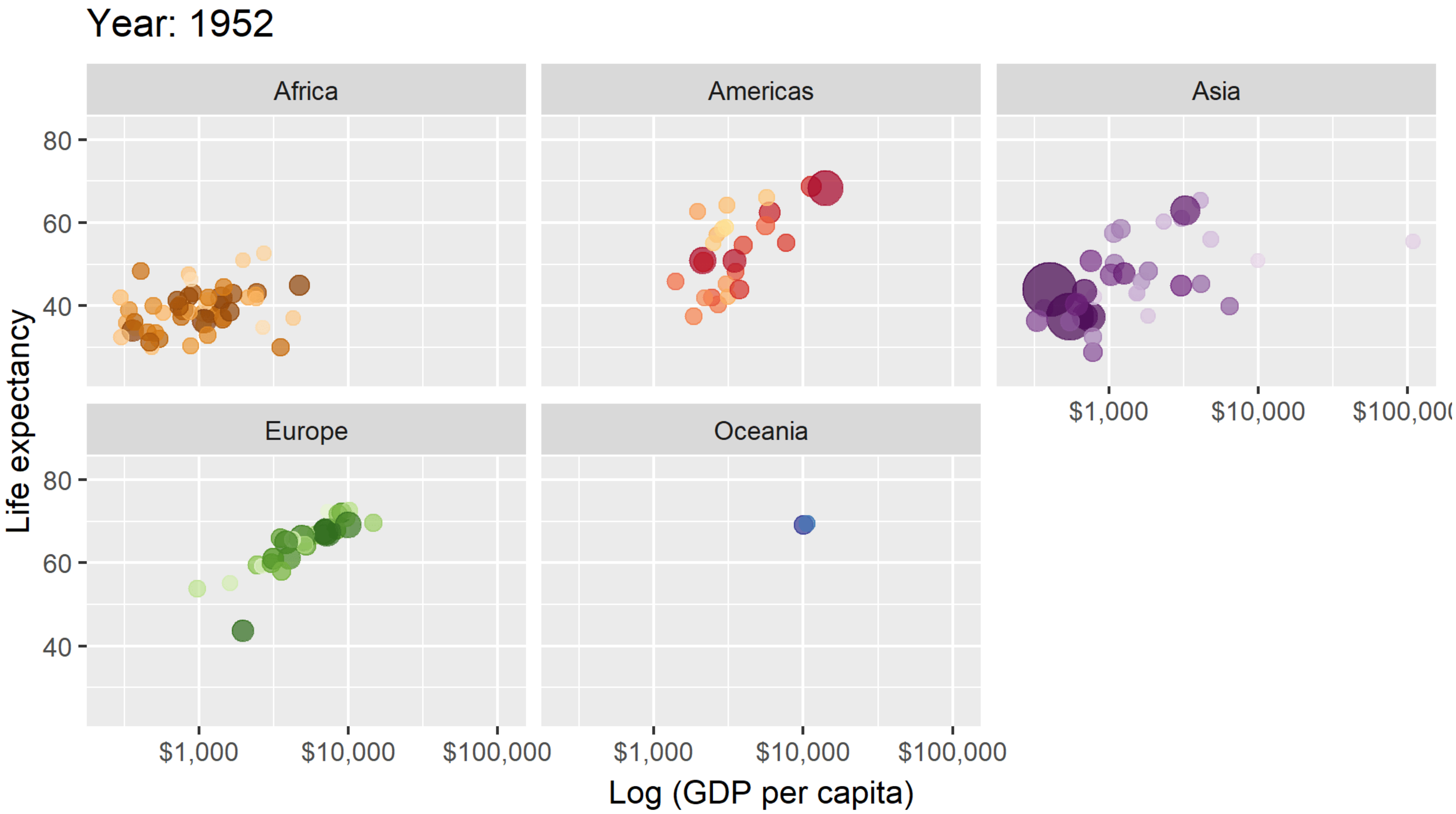
- There is a positive relationship between GDP per capita and life expectancy
- Is it causal? Unclear.
- What other factors might be at play?
- Can we rule those out?
- What about time?

What else? (cont.)

Elaborate extension: Animation! (See the next slide for the resulting GIF.)

```r
# library(gganimate)
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10(labels = scales::dollar) +
  facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  labs(title = 'Year: {frame_time}', x = 'Log (GDP per capita)', y = 'Life expectancy') +
  transition_time(year) +
  ease_aes('linear')
```

What else? (cont.)



Note that this animated plot provides a much more intuitive understanding of the underlying data. Just as Hans Rosling intended.

Correlation vs. causation

- A good ML model will account for continent, time, GDP per capita, and many other factors to predict life expectancy

- How do you answer each of these to a policymaker?

  - What is the *predicted effect* of increasing GDP by \$1000 on life expectancy?
  - What is the *counterfacual effect* of increasing GDP by \$1000 on life expectancy?

- Economists' comparative advantage is in combining machine learning with economic theory to answer these questions

But do increases in GDP per capita cause

- We can't answer this question with a simple plot.
- We also can't answer this question with a very fancy plot.
- What do we need?

> - A model
> - A causal identification strategy
> - More granular (bigger) data

What else? (cont.)

There's a lot more to say, but I think we'll stop now for today's lecture.

We also haven't touched on ggplot2's relationship to "tidy" data.

- It actually forms part of a suite of packages collectively known as the tidyverse.
- We will get back to this in Lecture 5.

Rest assured, you will be using ggplot2 throughout the rest of this course and developing your skills along the way.

- Your very first assignment (coming up) is a chance specifically to hone some of those skills.

In the meantime, I want you to do some reading and practice on your own. Pick either of the following (or choose among the litany of online resources) and work through their examples:

- Chapter 3 of *R for Data Science* by Hadley Wickham and Garett Grolemund.
- *Data Visualization: A Practical Guide* by Kieran Healy.
- Designing ggplots by Malcom Barrett.

Next lecture: Deep dive into Git(Hub).