

DATA MATTERS.TM

DATA SCIENCE SHORT COURSE SERIES

AUGUST 7 - 11, 2023

Welcome!
We'll be
starting soon!

VIRTUAL VIA ZOOM



ODUM INSTITUTE FOR
RESEARCH IN SOCIAL SCIENCE

ncds

THE NATIONAL CONSORTIUM
for DATA SCIENCE

renci

Visualization for Data Science in R

Angela Zoss

Data Matters Fall 2023

<https://www.angelazoss.com/RVis-2Day/>

Try right now:
Open RStudio
Try running “library(tidyverse)”
Tell me about any errors

Slides and files

<https://github.com/amzoss/RVis-2Day>

Schedule, Day 1

Session	Topics	Duration
Session 1	Visualization and data science Intro, setup, basic ggplot2 syntax	9:30 a.m. – 10:35 a.m.
Morning break		10:35 a.m. – 10:50 a.m.
Session 2	Trying more charts	10:50 a.m. – 11:55 a.m.
Lunch		11:55 a.m. – 1:10 p.m.
Session 3	Customizing plots, saving charts out	1:10 p.m. – 2:15 p.m.
Afternoon break		2:15 p.m. – 2:30 p.m.
Session 4	Plot inheritance, advanced examples	2:30 p.m. – 3:35 p.m.
Q&A		3:35 p.m. – 3:40 p.m.

Schedule, Day 2

Session	Topics	Duration
Session 1	ggplot2 review, advanced techniques	9:30 a.m. – 10:35 a.m.
Morning break		10:35 a.m. – 10:50 a.m.
Session 2	Working with text variables	10:50 a.m. – 11:55 a.m.
Lunch		11:55 a.m. – 1:10 p.m.
Session 3	Simple interactive plots	1:10 p.m. – 2:15 p.m.
Afternoon break		2:15 p.m. – 2:30 p.m.
Session 4	Building visualizations into layouts	2:30 p.m. – 3:35 p.m.
Q&A		3:35 p.m. – 3:40 p.m.

Other course logistics

- We all have different skill levels here. That's great!
- Questions and interruptions are welcome, especially if you are lost. I want everyone to be able to follow along.
- You may know the answer to someone else's question. If it's quick, feel free to make suggestions in the Zoom chat. Otherwise, I'll be happy to address it myself.
- You may have more advanced suggestions on top of what I'm teaching. Please try not to share these in chat. Too much chat can be a distraction, and I have a specific sequence I follow to keep the content approachable. You can share advanced things in Slack instead.

Set up environment

- R
- RStudio
- packages

Packages:

- tidyverse
- readxl
- markdown
- knitr
- plotly
- colorspace
- dt
- crosstalk
- flexdashboard
- here

Visualization for Data Science

Why visualize in R?

- Quickly explore data
- Save time switching to another tool
- Use charts to inspire new analyses and vice versa
- Reproducibility

Why care about reproducibility?

- Open science makes review easier
- Increasingly a requirement
- Saves you a lot of time trying to figure out what you did last time!

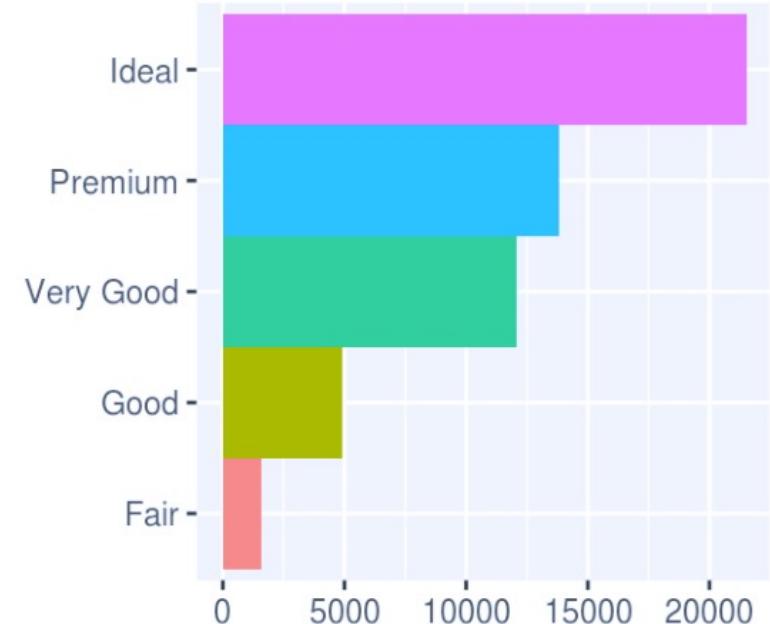
*“Your closest collaborator is **you** six months ago,
but you don’t reply to emails.”*

- *Mark Holder*

ggplot2

What is ggplot2?

an R package designed to create plots based on a theory of the grammar of graphics.



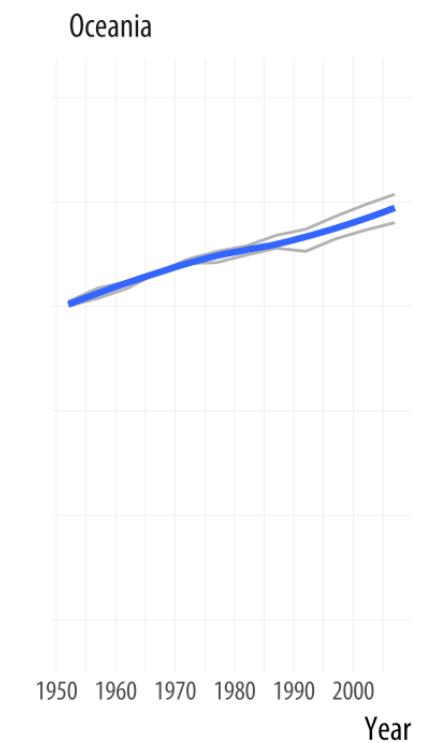
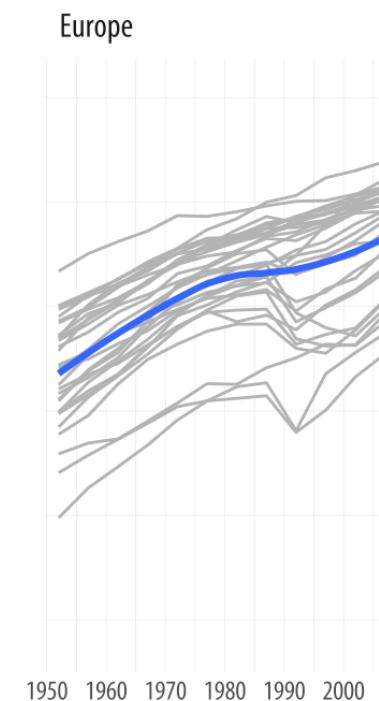
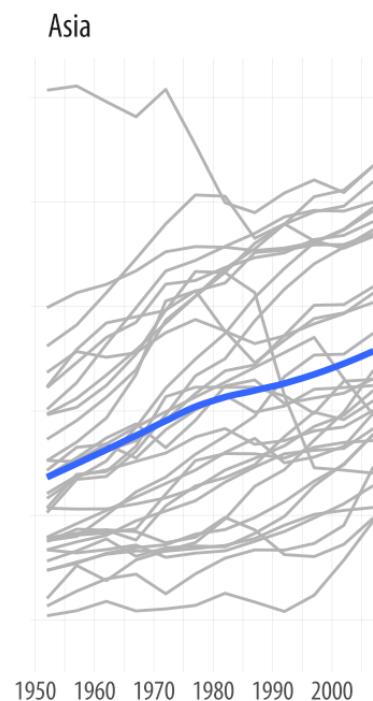
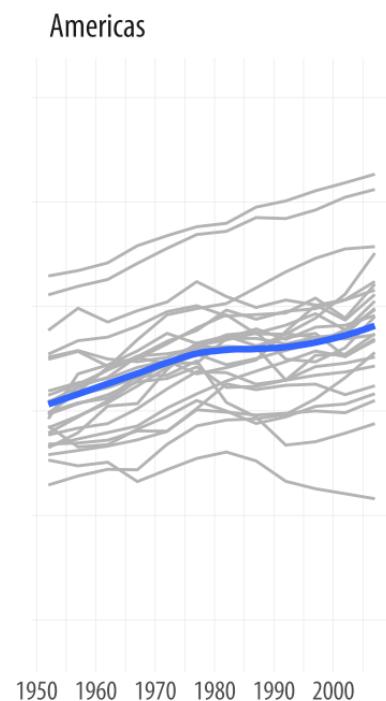
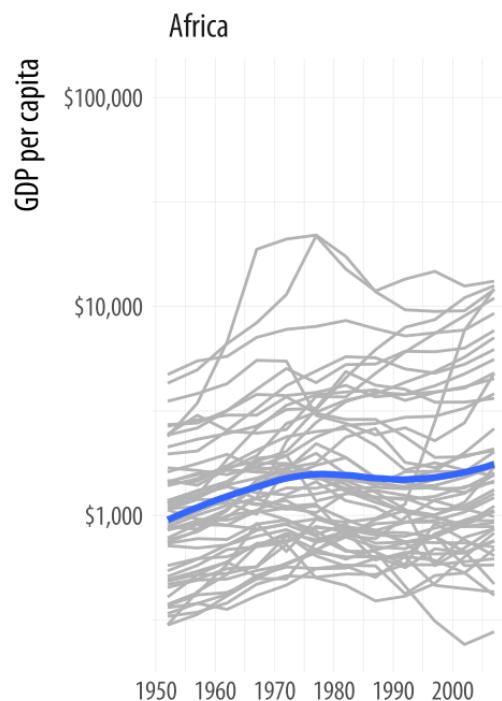
Grammar of graphics

1. DATA: a set of data operations that create variables from datasets
2. TRANS: variable transformations (e.g., rank)
3. SCALE: scale transformations (e.g., log)
4. COORD: a coordinate system (e.g., polar)
5. ELEMENT: graphs (e.g., points) and their aesthetic attributes (e.g., color)
6. GUIDE: one or more guides (axes, legends, etc.).

Wilkinson, Leland. (2005). *The grammar of graphics (2nd ed)*. New York: Springer.

ggplot2 examples

GDP per capita on Five Continents

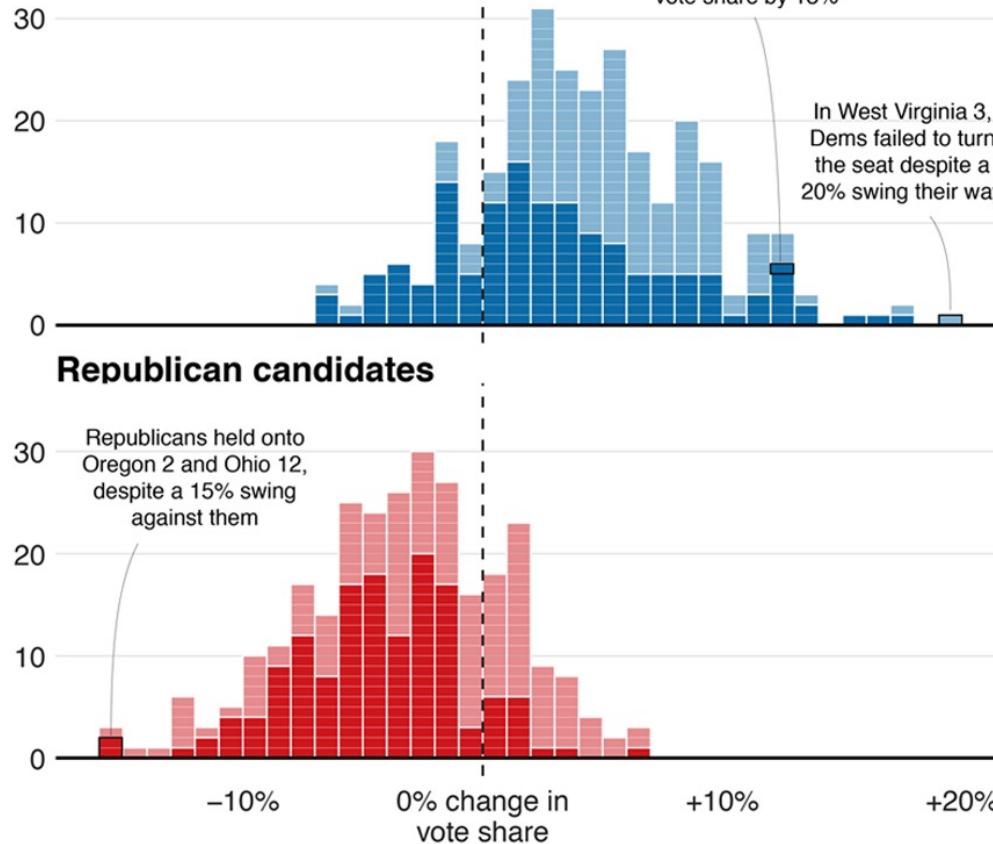


<http://socviz.co/groupfacettx.html>

Blue wave

■ Won seat ■ Didn't win

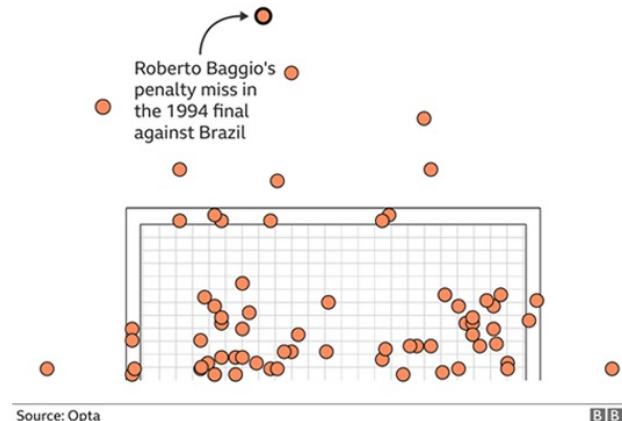
Democrat candidates



Source: AP, 19:01 ET

Where penalties are saved

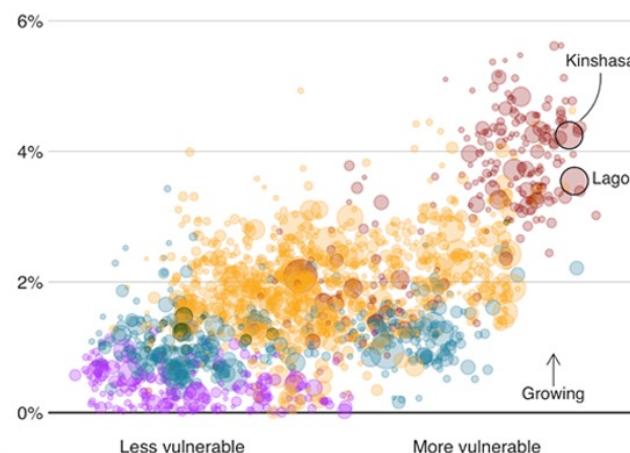
World Cup shootout misses and saves, 1982-2014



Fast-growing cities face worse climate risks

Population growth 2018-2035 over climate change vulnerability

■ Africa ■ Asia ■ Americas ■ Europe ■ Oceania



BBC

Source: Verisk Maplecroft. Circle size represents current population.

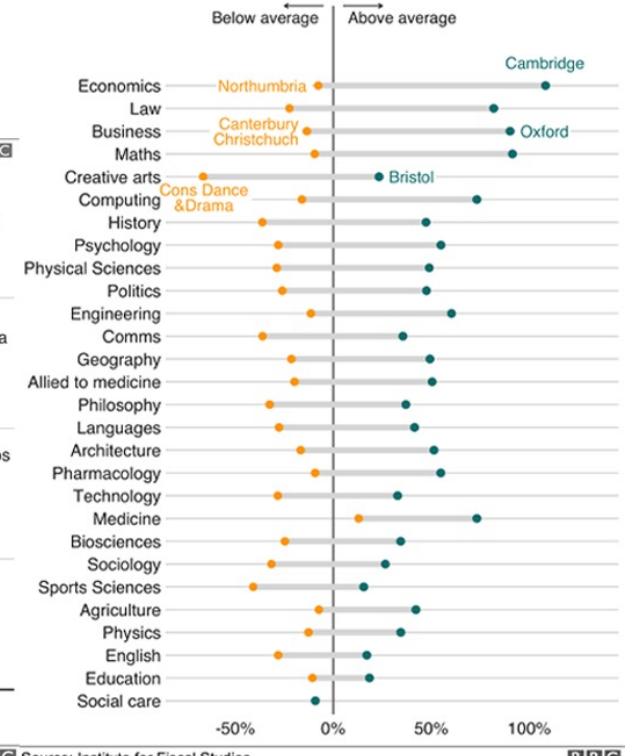
MPs rejected Theresa May's deal by 230 votes



Source: Commons Votes Services. Excludes 'tellers', the Speaker and deputies BBC

Earnings vary across unis even within subjects

Impact on men's earnings relative to the average degree

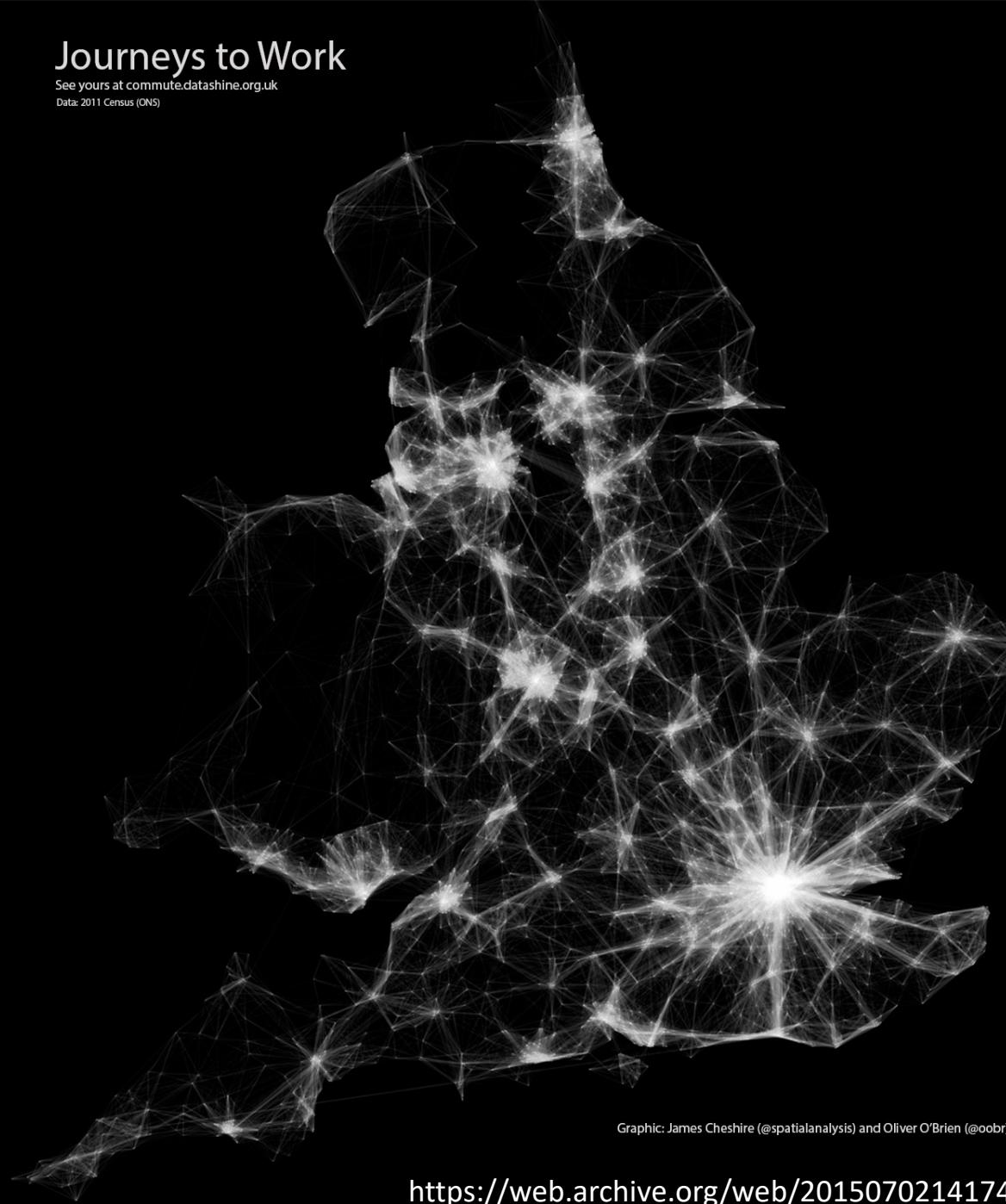


BBC Source: Institute for Fiscal Studies

Journeys to Work

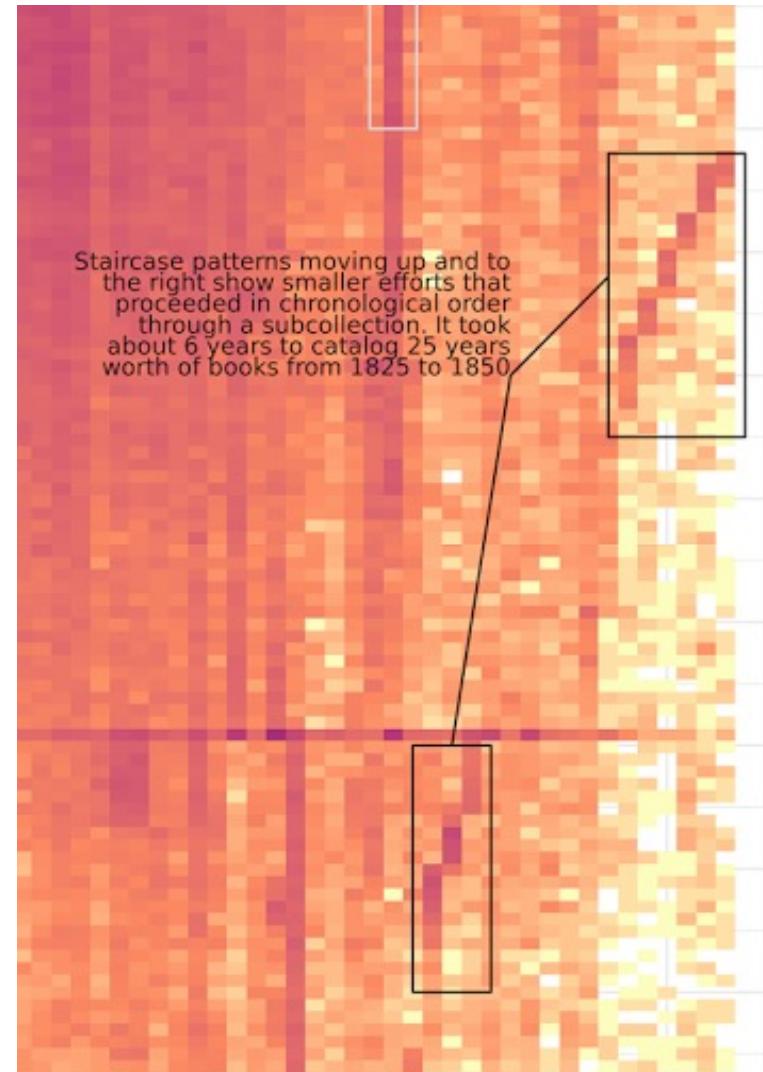
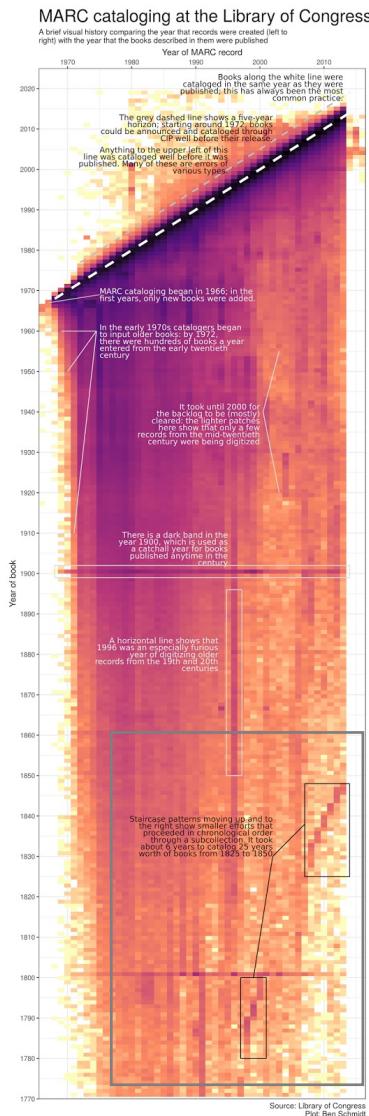
See yours at commute.datashine.org.uk

Data: 2011 Census (ONS)



Graphic: James Cheshire (@spatialanalysis) and Oliver O'Brien (@ooibr)

<https://web.archive.org/web/20150702141747/http://spatial.ly/2015/03/mapping-flows/>



Why ggplot2 instead of base R?

- nice defaults
- easy faceting
- (arguably) more natural syntax
- can switch chart types more easily

“Why I use ggplot2”, David Robinson

<http://varianceexplained.org/r/why-i-use-ggplot2/>

R vs. Excel, Tableau, etc.

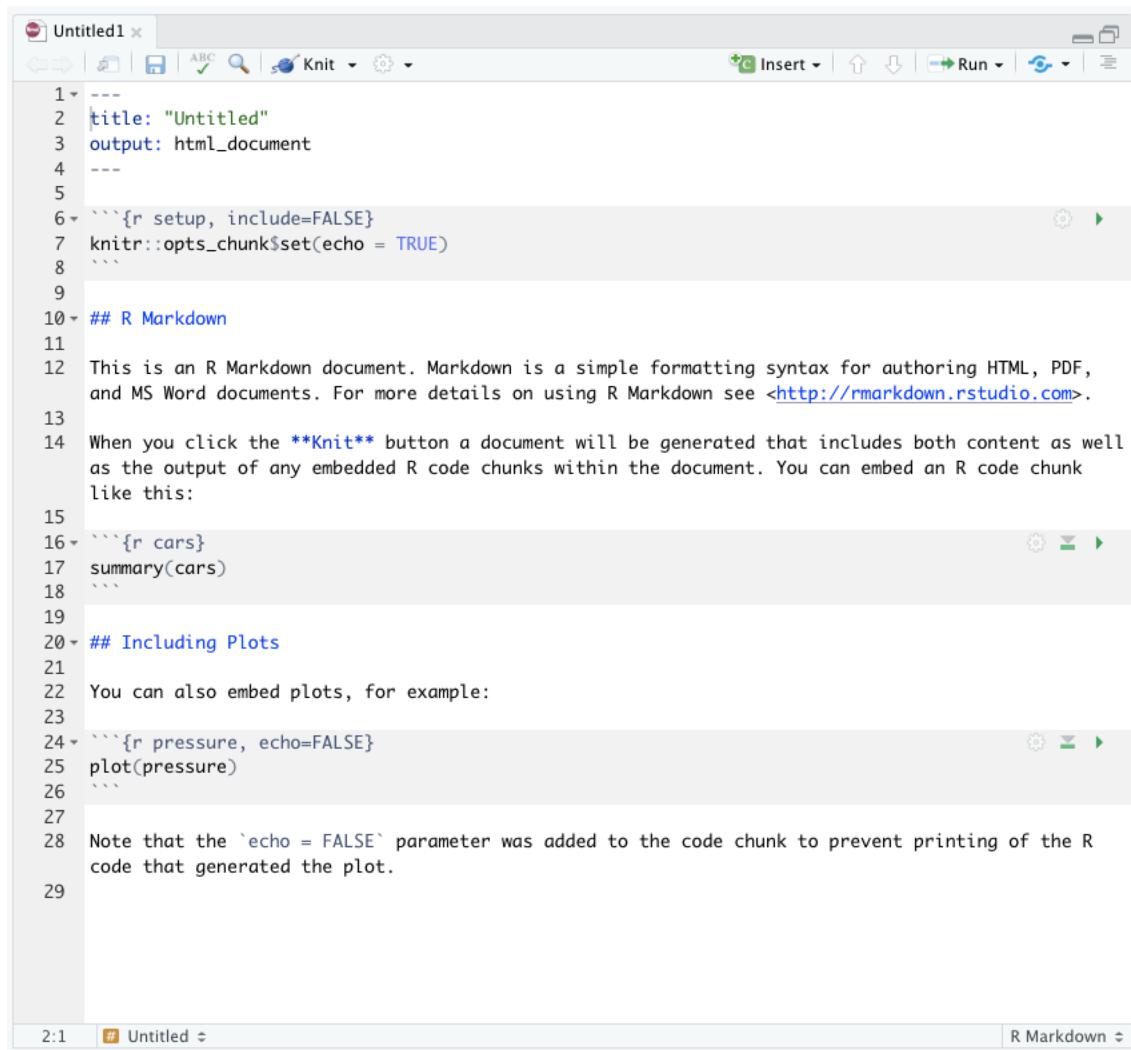
Questions to ask:

- Are you already using R? Why switch?
- Are you going to have to share this process or reproduce it? Try R!
- Is it a quick project, or will others work on it? Maybe Excel is fine.
- Do you need to try a bunch of charts quickly, build interactive components, etc.? Tableau might be more powerful and faster.

Working in RStudio

R Markdown files

- Blend “normal” text (using Markdown syntax for formatting) with code chunks and their output
- Can be compiled (“knit”) into other formats (HTML, Word, PDF)
- Similar to Jupyter Notebooks for Python
- NB: The next generation of R Markdown is [Quarto](#)



The screenshot shows the RStudio interface with an R Markdown file open. The title bar says "Untitled1". The toolbar includes icons for file operations, ABC, Knit, and settings. The main pane displays the following R Markdown code:

```
1 ---  
2 title: "Untitled"  
3 output: html_document  
4 ---  
5  
6 ```{r setup, include=FALSE}  
7 knitr::opts_chunk$set(echo = TRUE)  
8 ```  
9  
10 ## R Markdown  
11  
12 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>.  
13  
14 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:  
15  
16 ```{r cars}  
17 summary(cars)  
18 ```  
19  
20 ## Including Plots  
21  
22 You can also embed plots, for example:  
23  
24 ```{r pressure, echo=FALSE}  
25 plot(pressure)  
26 ```  
27  
28 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.  
29
```

The status bar at the bottom shows "2:1" and "Untitled".

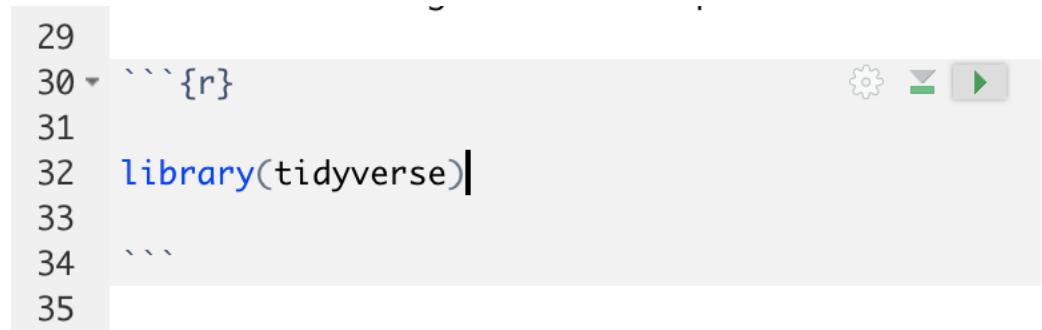
Why R Markdown?

- Plots show up inline
- Easier to incorporate explanatory text and materials
- Like to be able to easily run one chunk at a time

Caution: Running things out of order can mean your code won't work again later. Clear your environment often and run code chunks in order to be safe.

R Markdown test

- File → New File → R Markdown
- Click OK to accept defaults
- Type inside the first few lines to edit the YAML header (edit title, add author, etc.)
- Add a new R code chunk at the end of the file using Insert → R
- Type some R code inside the code chunk:
library(tidyverse)
- Run the new code chunk



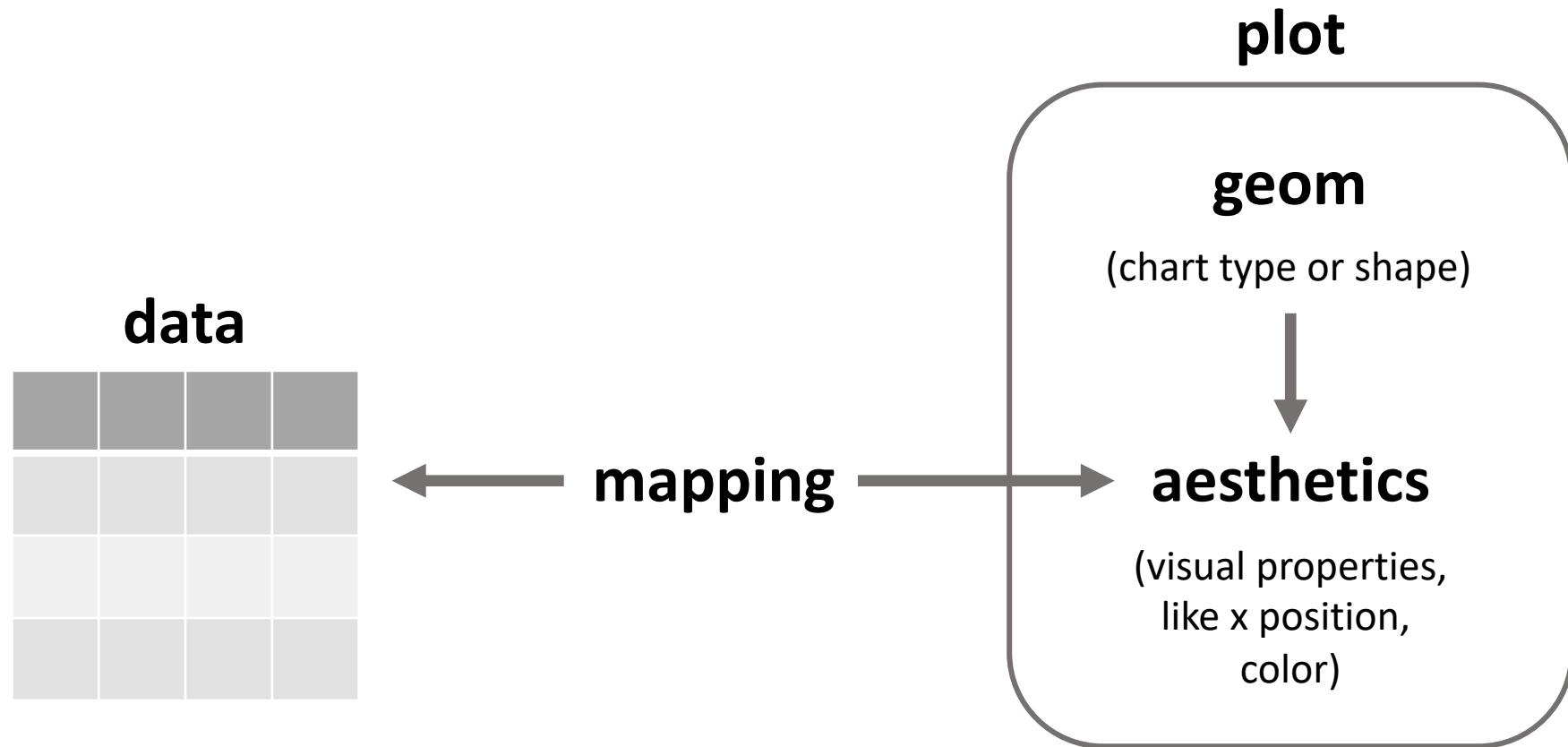
A screenshot of the RStudio interface showing an R code editor. The code in the editor is:

```
29
30  ````{r}
31
32  library(tidyverse)|
33
34  ...
35
```

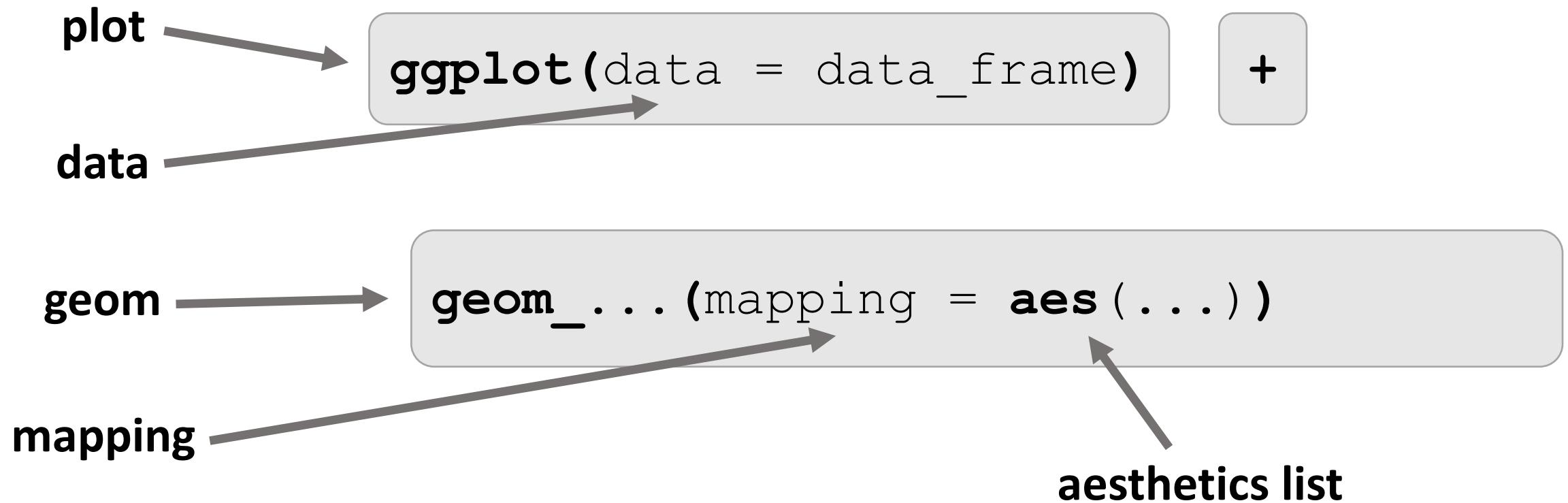
The code editor has a light gray background. Lines 29 through 35 are visible. Line 32 contains the text "library(tidyverse)" with the word "library" highlighted in blue. The RStudio toolbar is visible at the top right of the editor window.

ggplot2: making a basic plot

Basic elements in any ggplot2 visualization



Template for a simple plot

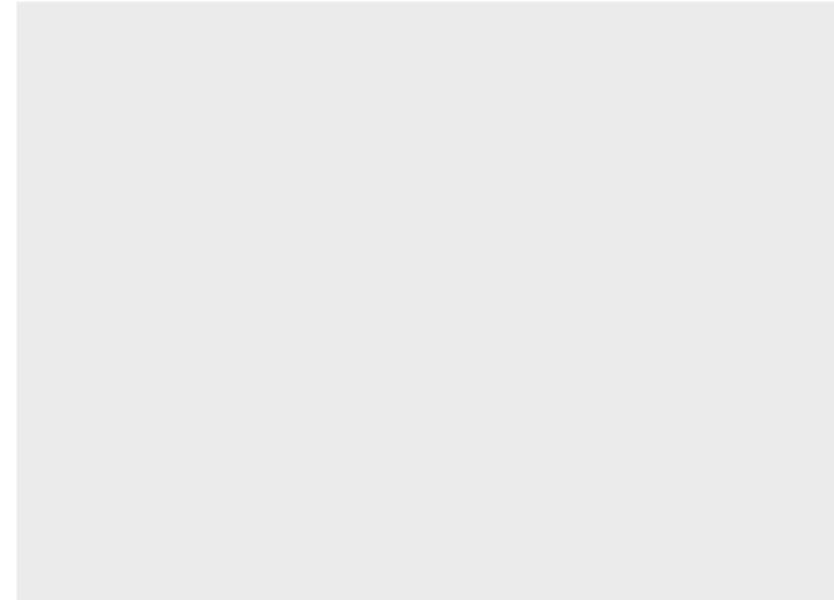


1. Set the data

"iris"

	Petal.Width	Petal.Length	Species
	0.3	1.4	setosa
	1.3	4.0	versicolor
	2.1	5.7	virginica

```
ggplot(data=iris)
```



2. Choose a shape layer

"iris"

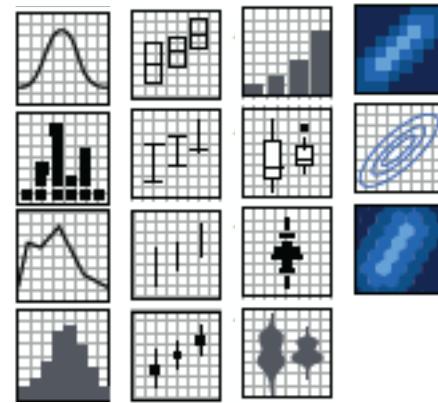
	Petal.Width	Petal.Length	Species
	0.3	1.4	setosa
	1.3	4.0	versicolor
	2.1	5.7	virginica

```
ggplot(data=iris) +  
  geom_point()
```

Error: geom_point requires
the following missing
aesthetics: x and y

Types of geoms

- geom_bar()
- geom_point()
- geom_histogram()
- geom_map()
- etc.



[ggplot2 cheatsheet](#)

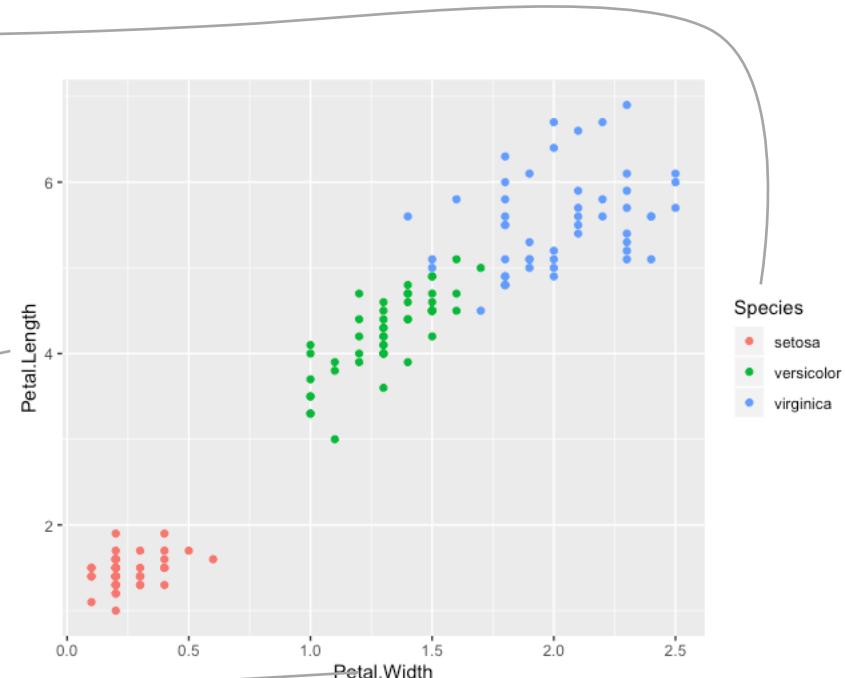
3. Map variables to aesthetics

“iris”

Petal.Width	Petal.Length	Species
0.3	1.4	setosa
1.3	4.0	versicolor
2.1	5.7	virginica

x position y position color

```
ggplot(data=iris) +  
  geom_point(  
    mapping=aes(x=Petal.Width,  
                y=Petal.Length,  
                color=Species))
```

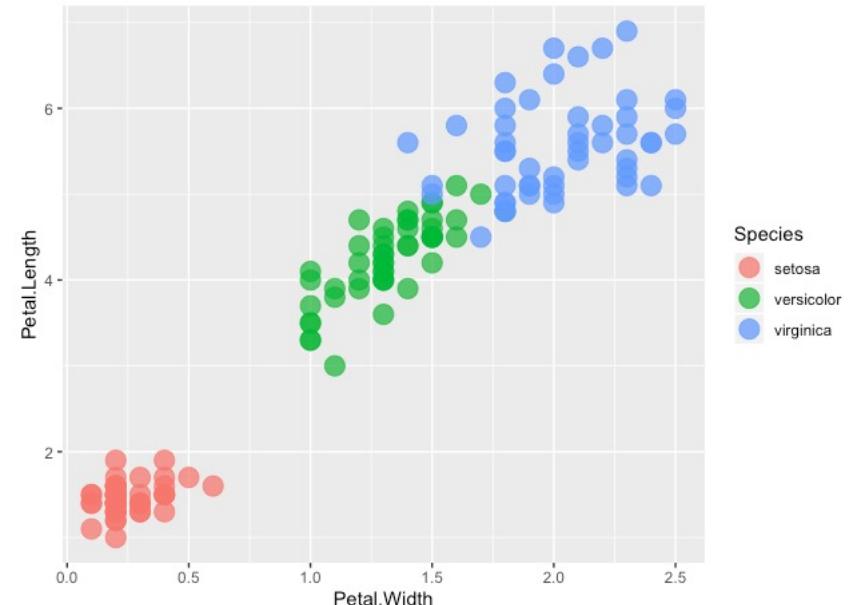


4. Add non-variable adjustments

"iris"

	Petal.Width	Petal.Length	Species
	0.3	1.4	setosa
	1.3	4.0	versicolor
	2.1	5.7	virginica

```
ggplot(data=iris) +  
  geom_point(  
    mapping=aes(x=Petal.Width,  
                y=Petal.Length,  
                color=Species),  
    size=5, alpha=.75)
```



Fixing Errors

Debugging code

- Start simple
 - If you see an error:
 - read error message for hints
 - check for problems with spelling/punctuation marks
 - Get code to run without errors
 - Check result to see if it makes sense
- 
- Add a small change
 - Get code to run without errors
 - Check result to see if it makes sense
 - etc.

RStudio built-in help documentation

- In console, type
?<function or package name>
- In help tab (not help menu), type into main search box
- In package tab, click on package name
- In help menu, use the Cheat Sheets submenu to download cheat sheet PDFs

ggplot2 Cheat Sheet

Help →

Cheatsheets →

Data Visualization with ggplot2

Data Visualization with ggplot2 :: CHEAT SHEET



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(<mapping> = aes(<MAPPINGS>),  
  stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTIONS> +  
  <SCALE_FUNCTIONS> +  
  <THEME_FUNCTIONS>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

```
aesthetic mappings    data    geom  
qplot(x = cyl, y = hwy, data = mpg, geom = "point")  
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.  
last_plot() Returns the last plot  
ggsave("plot.png", width = 5, height = 5) Saves last plot as 5'x5' file named "plot.png" in working directory. Matches file type to file extension.

### Geoms



Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.



#### GRAPHICAL PRIMITIVES



a <- ggplot(economics, aes(date, unemployed))  
b <- ggplot(seals, aes(x = long, y = lat))



a + geom_blank()  
(Useful for expanding limits)



b + geom_curve(aes(end = lat + 1, nudge_y = -1, check_overlap = TRUE), x, y, yend, alpha, angle, color, curvature, linetype, size, linemetre=1)



a + geom_path(linend = "butt", linejoin = "round", x, y, alpha, color, group, linetype, size)



a + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), x, y, alpha, color, fill, group, linetype, size)



b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), x, y, alpha, color, fill, group, linetype, size)



a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900), x, y, alpha, color, fill, group, linetype, size)



a + geom_smooth(method = lm), x, y, alpha, color, fill, group, linetype, size, weight



a + geom_text(aes(label = cyl), nudge_x = 1, nudge_y = -1, check_overlap = TRUE), x, y, label, alpha, angle, color, fontfamily, fontface, hjust, linheight, size, vjust



#### TWO VARIABLES



continuous x, continuous y



e + geom_label(aes(label = cyl), nudge_x = 1, nudge_y = -1, check_overlap = TRUE), x, y, label, alpha, angle, color, fontfamily, fontface, hjust, linheight, size, vjust



e + geom_liner(hight = 2, width = 2), x, y, alpha, color, fill, shape, size, stroke



e + geom_point(), x, y, alpha, color, fill, shape, size



e + geom_quantile(), x, y, alpha, color, group, linetype, size, weight



e + geom_rug(sides = "bl"), x, y, alpha, color, fill, group, linetype, size, weight



e + geom_smooth(method = lm), x, y, alpha, color, fill, group, linetype, size, weight



e + geom_text(aes(label = cyl), nudge_x = 1, nudge_y = -1, check_overlap = TRUE), x, y, label, alpha, angle, color, fontfamily, fontface, hjust, linheight, size, vjust



#### continuous bivariate distribution



h <- ggplot(diamonds, aes(carat, price))



h + geom_bin2d(binwidth = c(0.25, 500)), x, y, alpha, color, fill, linetype, size, weight



h + geom_density2d(), x, y, alpha, colour, group, linetype, size



h + geom_hex(), x, y, alpha, colour, fill, size



#### continuous function



i <- ggplot(economics, aes(date, unemployed))



i + geom_line(), x, y, alpha, color, fill, linetype, size



i + geom_step(direction = "hv"), x, y, alpha, color, group, linetype, size



#### visualizing error



j <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)



j + geom_crossbar(fatten = 2), x, y, ymin, ymax, alpha, color, fill, group, linetype, size



j + geom_errorbar(), x, y, ymin, ymax, alpha, color, fill, group, linetype, size, width (also geom_errorbari())



j + geom_linerange(), x, ymin, ymax, alpha, color, group, linetype, size



j + geom_pointrange(), x, ymin, ymax, alpha, color, fill, group, linetype, size



#### maps



data <- data.frame(murder = USArrests$Murder, state = tolowerrownames(USArrests))  
map <- map_data("us-states")  
k <- ggplot(data, aes(fill = murder))



k + geom_map(aes(map_id = state), map = map) + expand_limits(x = map$long, y = map$lat), map_id, alpha, color, fill, linetype, size



#### THREE VARIABLES



seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))



l <- ggplot(mpg, aes(flf))



d + geom_bar()



x, y, alpha, color, fill, linetype, size, weight



l + geom_contour(aes(z = z)), x, y, alpha, color, group, linetype, size, weight



l + geom_interpolate(aes(fill = z), hjust = -0.5, vjust = 0.5, interpolate = FALSE), x, y, alpha, fill



l + geom_tile(aes(fill = z)), x, y, alpha, color, fill, linetype, size, width



RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at http://ggplot2.tidyverse.org • ggplot2 2.1.0 • Updated: 2016-11


```



ggplot2 cheatsheet

Get workshop files

URL: <https://github.com/amzoss/RVis-2Day>

On GitHub:

- Click green “Code” button and select “Download ZIP”
- Unzip files on your computer
 - Windows: Double-click, then look for “Extract Files” at the top
 - Mac: Double-click
- Note: have noticed some issues when using OneDrive to store files

In RStudio:

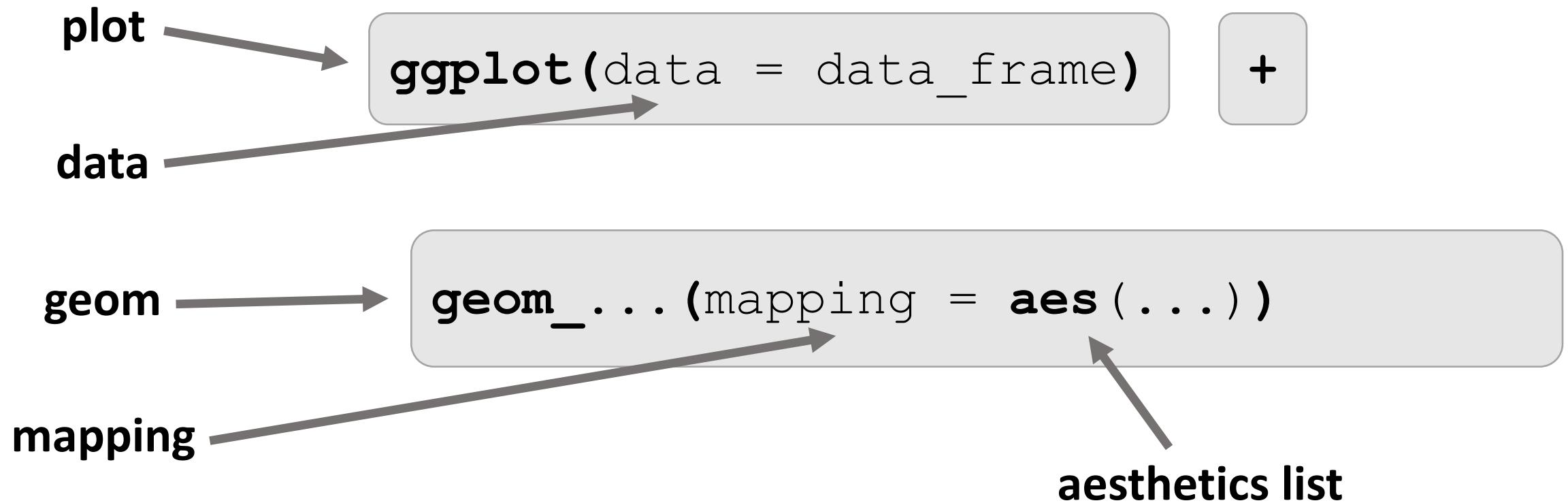
- Project → New project...
- Existing directory
- Select unzipped folder
- Create Project

Morning Break

Exercise 1: Inclusiveness Index

<https://belonging.berkeley.edu/inclusiveness-index>

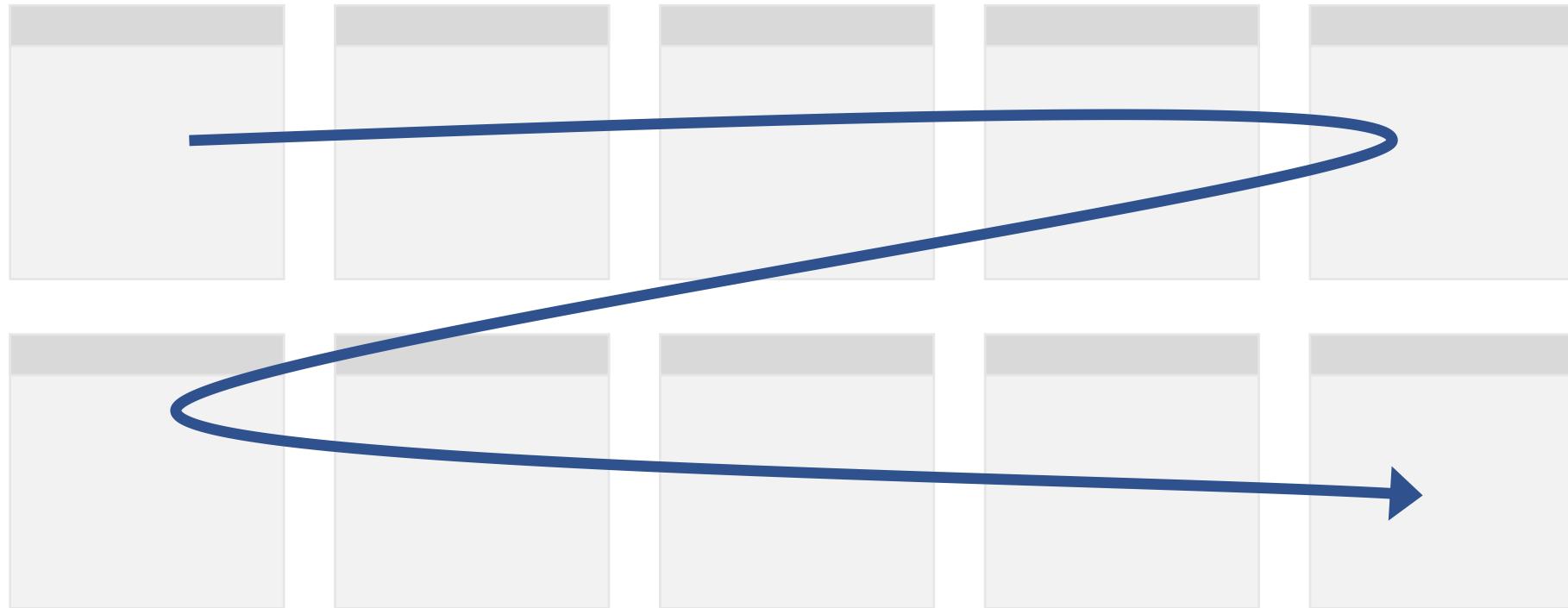
Template for a simple plot



Creating repeated charts

facet_wrap()

```
+ facet_wrap(vars(variable))
```

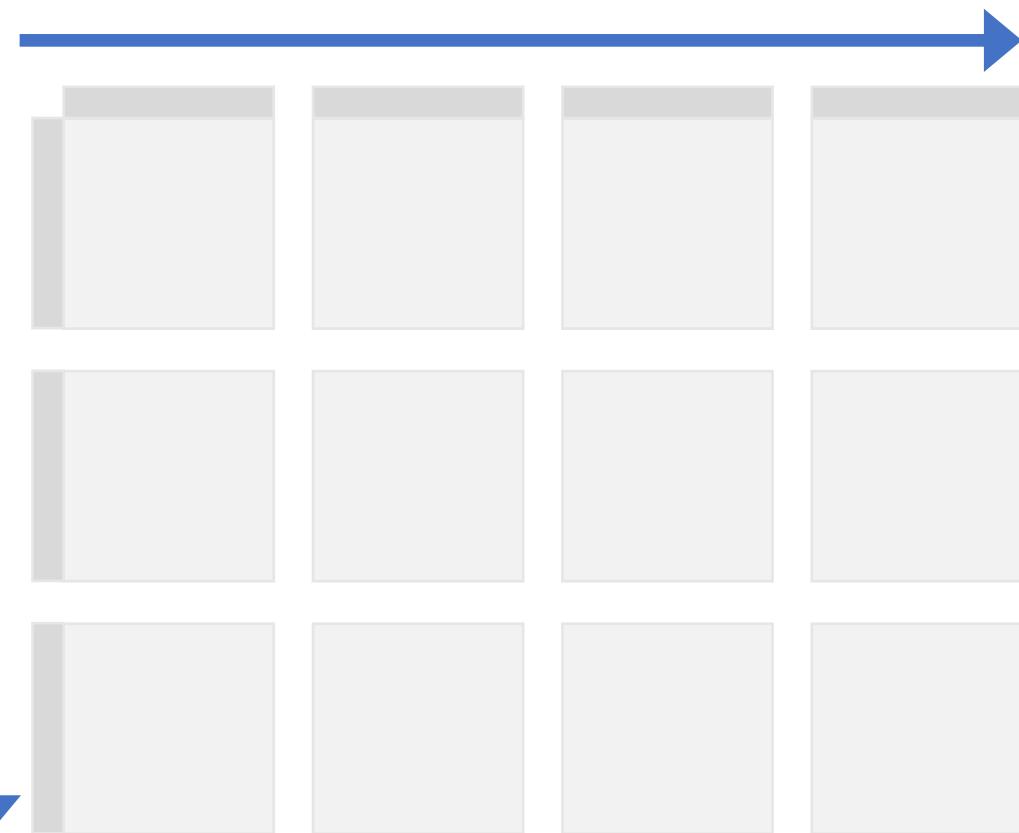


facet_grid()

```
+ facet_grid(rows=vars(yvar),  
             cols=vars(xvar))
```

Another categorical variable

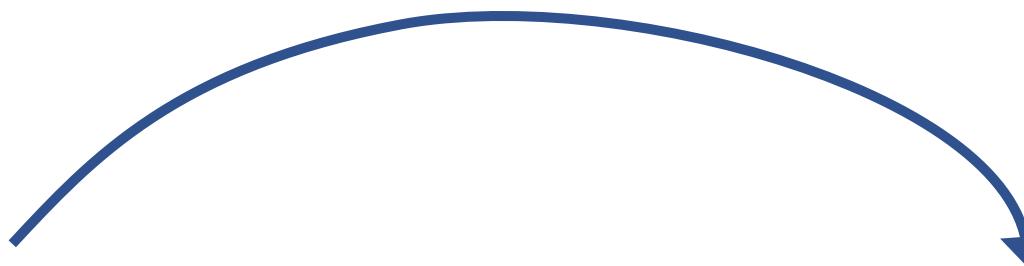
One categorical variable



Helpful data manipulation

Note: about %>%

- Loads automatically with tidyverse
- Used throughout tidyverse (except for ggplot2)
- Pushes data from the left into the function on the right



data_frame %>% function(args)

A blue curved arrow originates from the left side of the word "data_frame" and points to the right side of the word "function".

filter

Select a subset of rows

```
data %>% dplyr::filter(name == "John")
```

same as

```
dplyr::filter(data, name == "John")
```

[dplyr cheatsheet](#)

select

Select a subset of columns (many options!)

```
data %>% dplyr::select(id, name, age)
```

```
data %>% dplyr::select(-count)
```

[dplyr cheatsheet](#)

drop_na

Remove rows with NA values, either in any column or in specified columns

```
data %>% drop_na()
```

```
data %>% drop_na(age)
```

[tidyR cheatsheet](#)

count

Take a dataset, group it by one or more variables, and count the number of rows grouped. Count will be stored in a variable called “n”.

```
data %>% count(fruit)
```

fruit	n
apple	4
kiwi	10
orange	2

[dplyr cheatsheet](#)

```
data %>% count(fruit, quality)
```

fruit	quality	n
apple	low	1
apple	high	3
kiwi	high	6
kiwi	medium	4
orange	Low	2

count is same as group_by -> summarise

count() is shorthand for grouping by the categorical variable and then summarizing by the number of rows in each group.

```
data %>% count(fruit)
```

fruit	n
apple	4
kiwi	10
orange	2

```
data %>% group_by(fruit) %>%  
  summarise(n = n())
```

fruit	n
apple	4
kiwi	10
orange	2

Pipe data into ggplot

When doing data manipulation, can be easier to pipe results to ggplot

```
data_frame %>% ggplot()
```

same as

```
ggplot(data = data_frame)
```

Lunch

Exercise 2: Customizing charts

Accessibility

All graphics need alternative text for screen reader users.

alt= “**Chart type** of **type of data**
where **reason for including chart**”

Include a **link to data source**
somewhere in the text

Note: Alt text should be relatively short. For longer descriptions, use add_description() from the [savonliquide package](#)

[Writing alt text for data visualization/](#)

Alternative Text in R and R Markdown

- ggplot2 now has [alt option in labs\(\)](#); gets read by shiny but not knitr
- in the meantime, use [fig.alt](#) in code chunk (just for HTML output)
 - can pull ggplot2 alt text into fig.alt with:

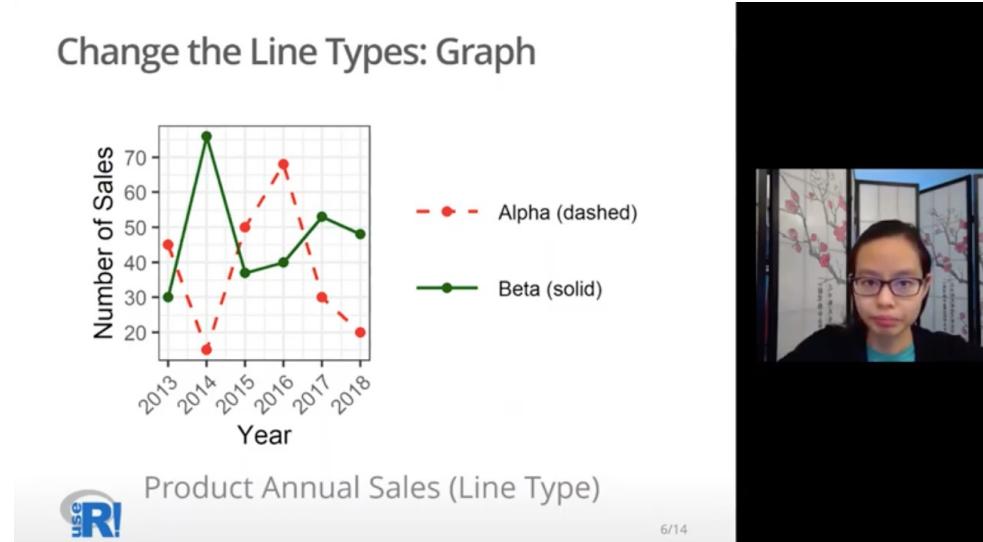
```
```{r, fig.alt=ggplot2::get_alt_text(g)} g```
```
  - [fig.cap](#) will be used instead, if there is no fig.alt
- embedded images in the Markdown:  
`![text used for both alt text and figure caption](path/to/image)`

[Alt Text in R](#)

# Color Vision Deficiency

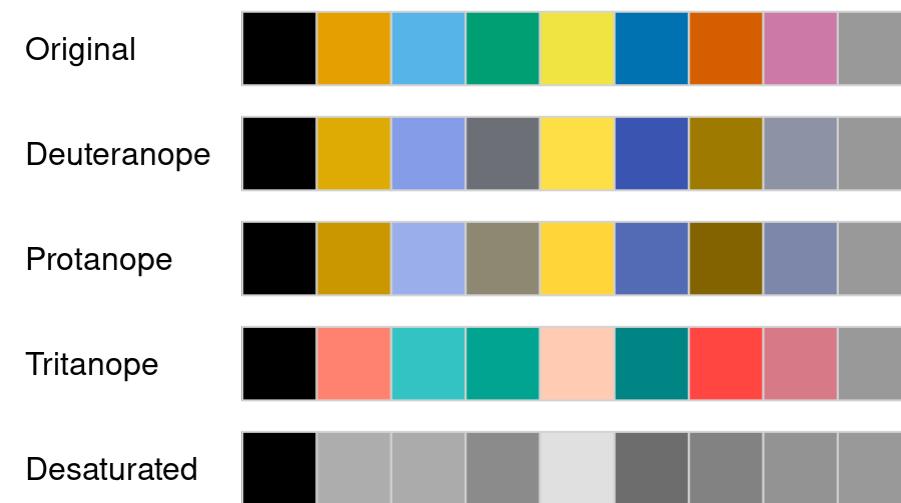
## Use dual encoding (not just color)

- Line color – also vary line type
- Point color – also vary point shape



## Use safe color palettes

- evaluate palettes to see how they look different for people with different types of color vision deficiency (CVD)



[Improving accessibility in data visualizations created by ggplot2](#)

[colorspace package: CVD emulation](#)

# Low Vision

- High color contrast
  - Both marks/text on background and labels on marks
  - Check contrast with [savonliquide package](#)
- Large text
  - See [“output-examples.md” file](#) for more sample code
  - Will cover in a later session

# Converting graphics to sound, touch, text

- [sonify package](#)
- [tactileR package](#)
- [BrailleR package](#)
  - Note: set plot title, subtitle, caption using labs()

# Accessibility Resources

- [savonliquide package](#)
- [Making betteR figures: Accessibility and Universal Design](#)
- [Highlights from the DVS accessibility fireside chat](#)

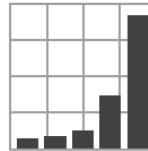
# Scales

- Scales control how an aesthetics mapping displays in the chart, e.g.:
  - the labels that show up on the axis
  - the number of example sizes in a size legend
  - the colors used for a “fill” or “color” mapping
- Modify these properties by adding a scale layer to the chart

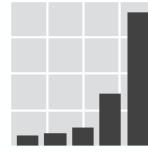
```
scale_x_continuous()
scale_y_log10()
scale_fill_discrete()
```

# Themes

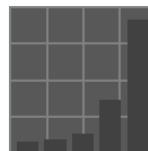
- Themes control properties of various visual elements, including:
  - Axis titles, text, ticks, lines
  - Plot colors, margins, text
  - Legend colors, margins, text
- Can add built-in themes as new layers, override specific theme elements, or build your own custom theme



**r + theme\_bw()**  
White background with grid lines.

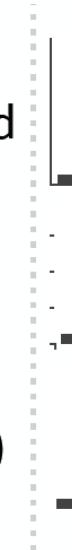


**r + theme\_gray()**  
Grey background (default theme).



**r + theme\_dark()**  
Dark for contrast.

[ggplot2 cheatsheet](#)



**r + theme\_classic()**  
**r + theme\_light()**  
**r + theme\_linedraw()**  
**r + theme\_minimal()**  
Minimal theme.  
**r + theme\_void()**  
Empty theme.

# geom vs. scale vs. theme

Adding something that will appear  
inside the **chart coordinate space**?

You will (almost always) be adding a **geom**!

Changing the way a **variable is displayed**?  
(e.g., different axis breaks, different color mapping)

You will be adding a **scale**!

Changing the **look and feel** of the chart?

You will be adding or making changes to a **theme**!

# More practice: Advanced ggplot2 workshop

[Workshop video](#)

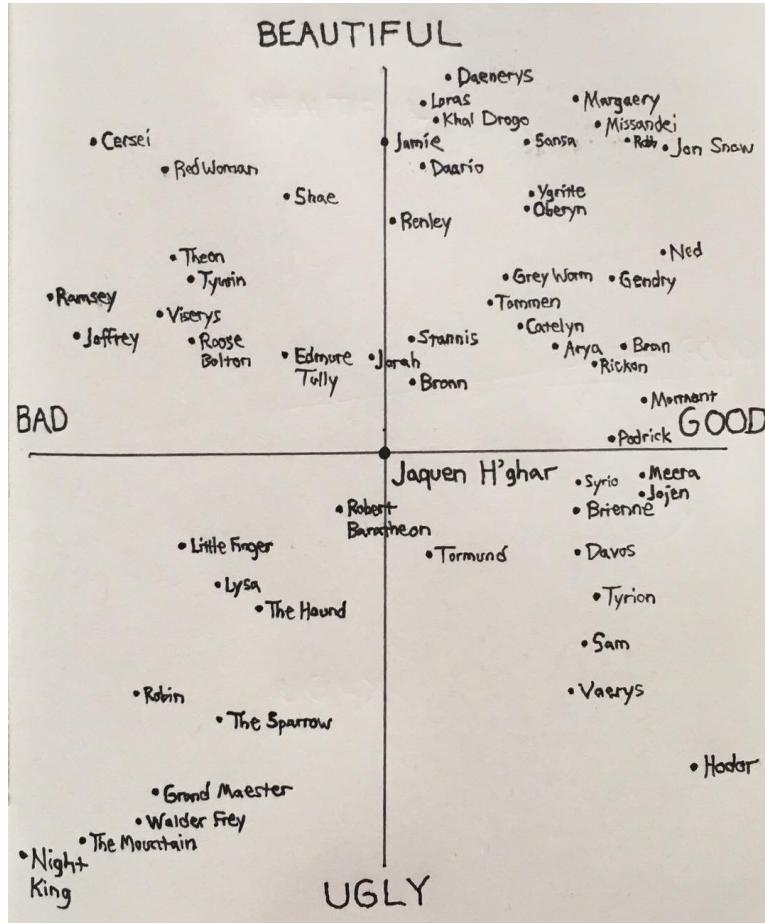
[Workshop materials](#)

Afternoon Break

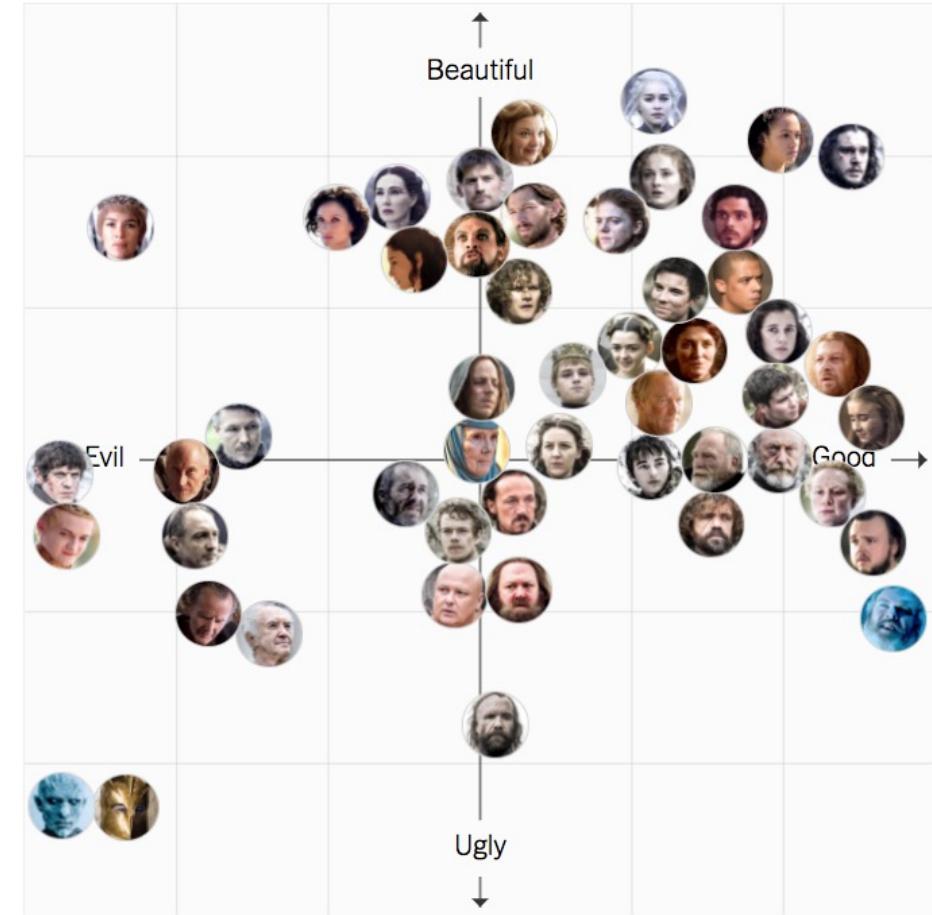
# Exercise 3: Game of Thrones character ratings

<https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-chart.html>

# Game of Thrones character ratings



<https://www.instagram.com/p/BWnn-YogX1n/>



<https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-chart.html>

# ggplot2: inheritance

# Template for a simple plot (review)

**main plot  
function**

```
ggplot(data = data_frame)
```

+

**shape  
layer**

```
geom_... (mapping = aes(...),
non-variable adjustments)
```

# Expanded template

**main plot  
function**

```
ggplot(data = data_frame,
 mapping = aes(...))
```

+

**shape  
layer**

```
geom_... (data = data_frame,
 mapping = aes(...),
 non-variable adjustments)
```

# Inheritance

data and aesthetics will carry through from main function to shape layers

**main plot  
function**

```
ggplot(data = data_frame,
 mapping = aes(...))
```

+

**shape  
layer**

```
geom_... (data = data_frame,
 mapping = aes(...),
 non-variable adjustments)
```

+

**shape  
layer**

```
geom_... (data = data_frame,
 mapping = aes(...),
 non-variable adjustments)
```

+

# Other helper packages

- [`gganonymize`](#) to randomize text in `ggplot2` figures
- [`visdat`](#) to visualize variable classes and missing data
- [`ggthemes`](#) for additional themes and scales, especially ones that match software defaults (e.g., Tableau)
- [`esquisse`](#) for building `ggplot2` charts interactively
- [`colorblindr`](#) for simulating color vision deficiency
- [`ggpubr`](#) for publication-ready plots

# ggplot2 Resources

- [General ggplot2 information](#)
- [R Graphics Cookbook](#) (recipes for plots)
- [R for Data Science](#) (online book that includes ggplot2)
- [ggplot2: Elegant Graphics for Data Analysis](#) (book by Hadley Wickham)
- [ggplot2 cheatsheet](#) (also in RStudio)
- [Data Carpentry lesson on ggplot2](#)
- [Data Visualization: A Practical Introduction](#), by Kieran Healy
- [RStudio “Visualize Data” Primer](#)

Thanks for your feedback!

[angela.zoss@duke.edu](mailto:angela.zoss@duke.edu)