Visualization for Data Science in R

Angela Zoss

Data Matters 2021

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

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

Objectives/Outline

Day 1: Basic visualizations

- Visualization and data science
- Basic ggplot2 syntax
- Basics of geoms and aes
- Categorical variables
- Manipulating data
- Customizing plots
- Advanced topics: mapping, saving charts out

Day 2: Websites and Shiny apps

- Simple interactive plots
- Arranging charts into dashboards
- Incorporating Shiny elements into documents, dashboards
- Advanced topics: full Shiny apps

Set up environment

- R
- RStudio
- packages

Packages:

- tidyverse
- knitr
- shiny
- plotly
- DT
- crosstalk
- flexdashboard
- maps
- mapproj
- sf

What does it mean to represent data visually? Why do it?

Math is hard

1		2		3		4	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Almost identical summary statistics:

x & y mean

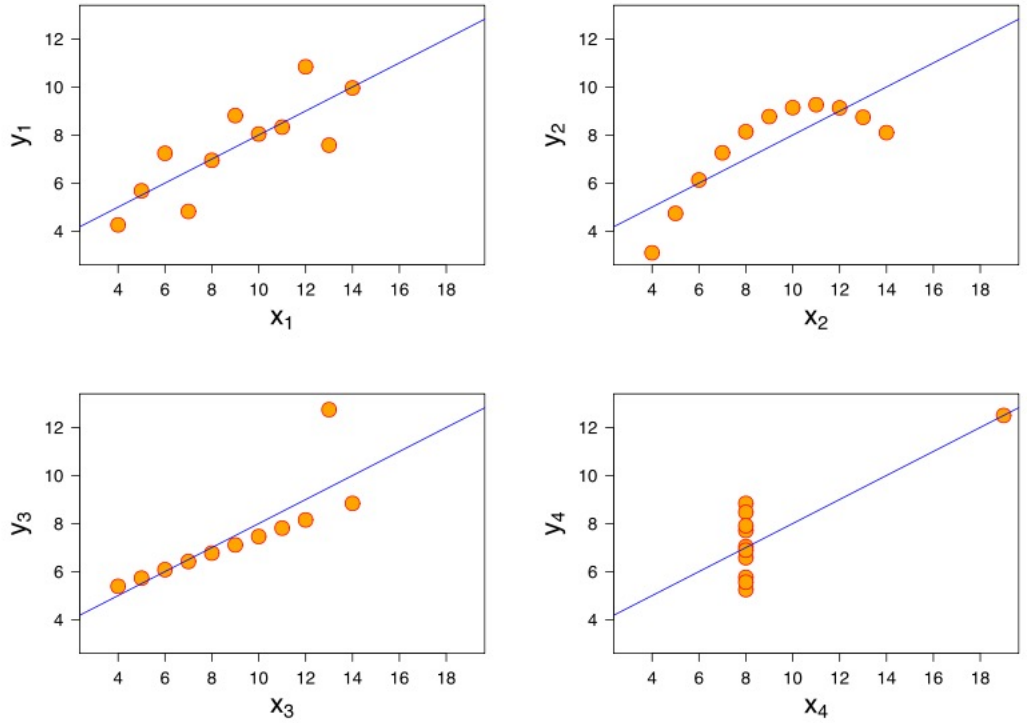
x & y variance

x-y correlation

x-y linear regression

https://en.wikipedia.org/wiki/Anscombe%27s_quartet

Shapes are much easier



Anscombe's Quartet

http://en.wikipedia.org/wiki/Anscombe%27s_quartet

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!

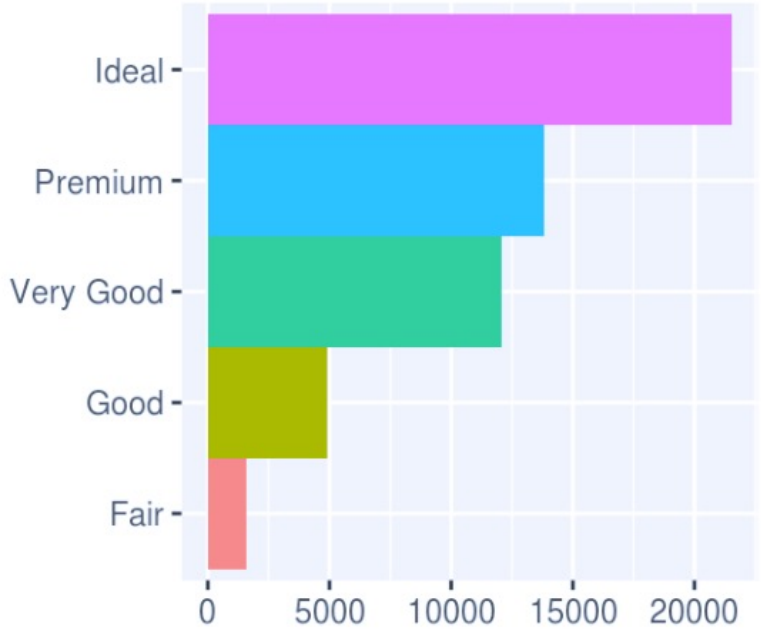
*“Your closest collaborator is **you** six months ago, but you don’t reply to emails.”*

- Mark Holder

ggplot2

What is ggplot2?

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



http://r4ds.had.co.nz/data-visualisation.html

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.

Why ggplot2 instead of base R?

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

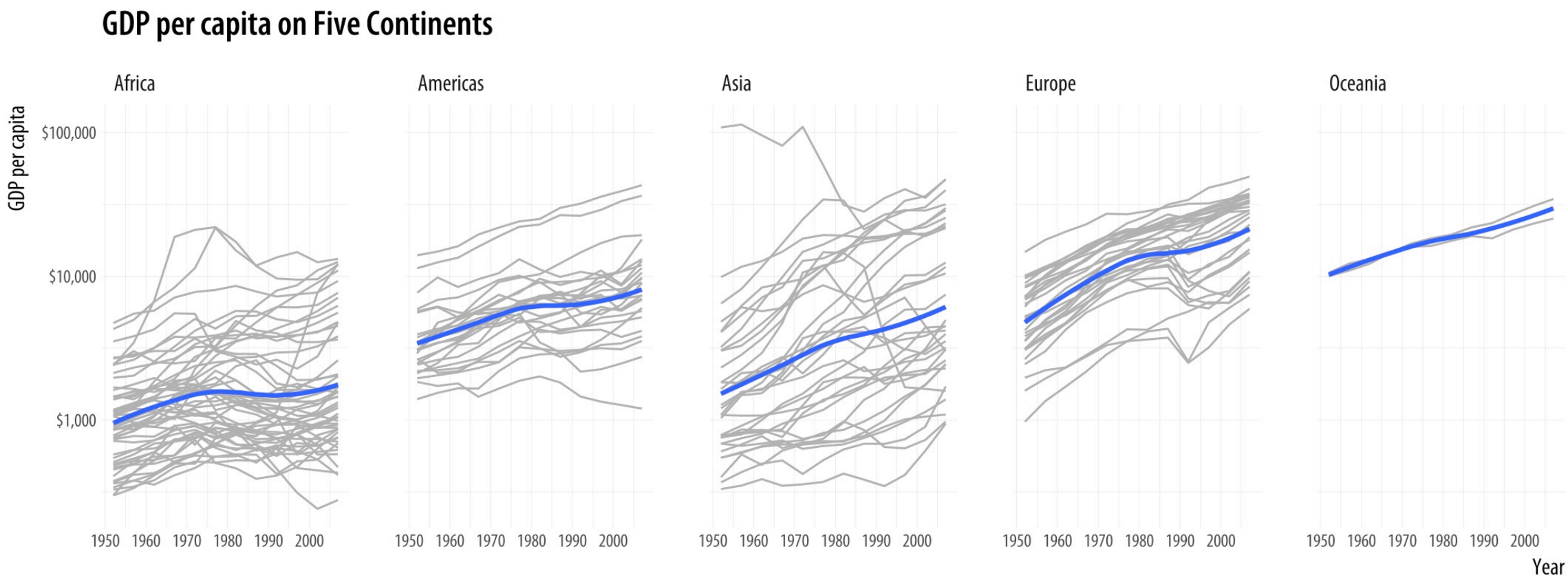
“Why I use ggplot2”, David Robinson
http://varianceexplained.org/r/why-I-use-ggplot2/

R vs. Excel, Tableau, etc.

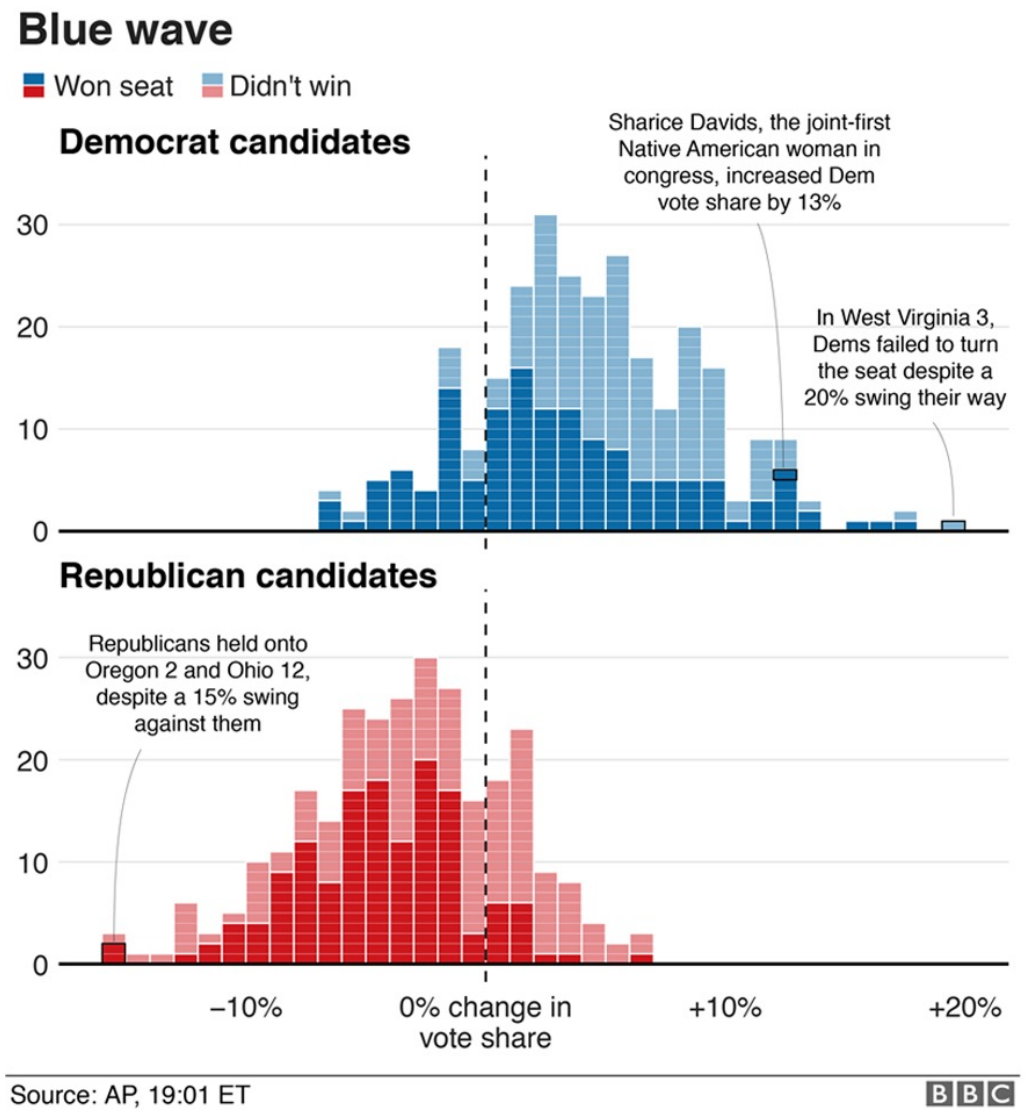
Questions to ask:

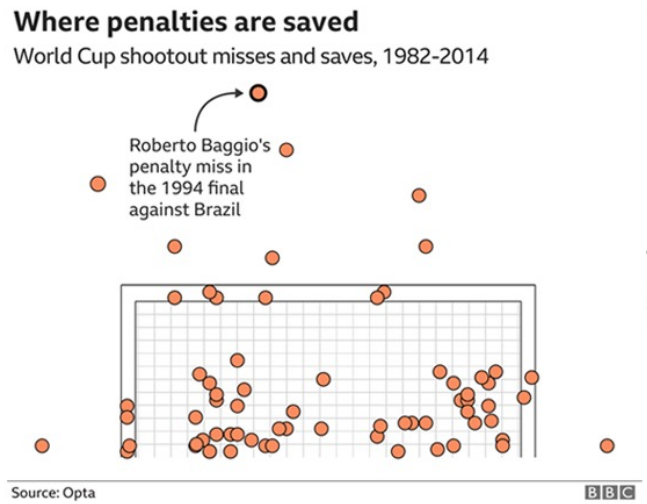
- Are you already using R? Why switch?
- Are you going to have to share this process or reproduce it? Try R!
- Is it a quick project, or will others work on it? Maybe Excel is fine.
- Do you need to try a bunch of charts quickly, build interactive components, etc.? Tableau might be more powerful and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

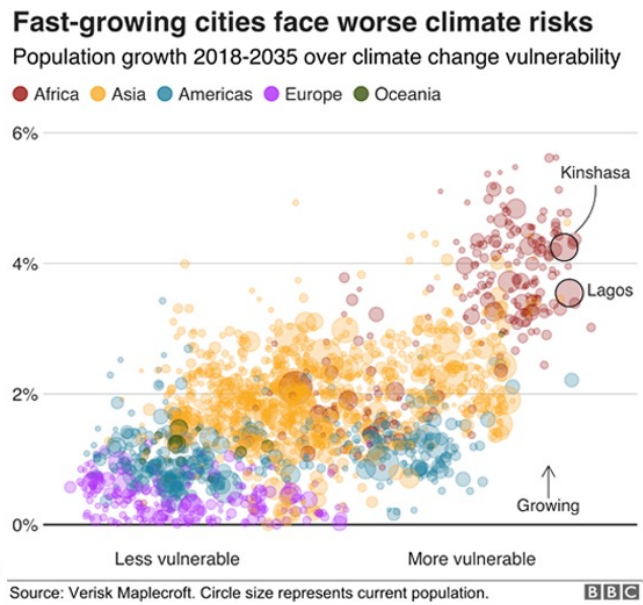
ggplot2 examples

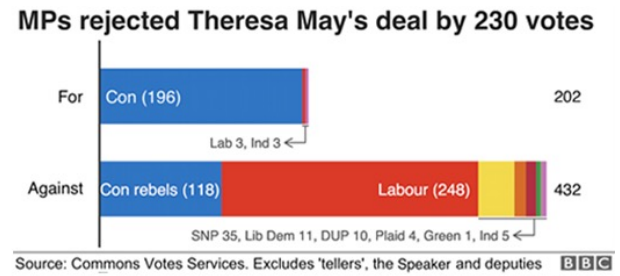


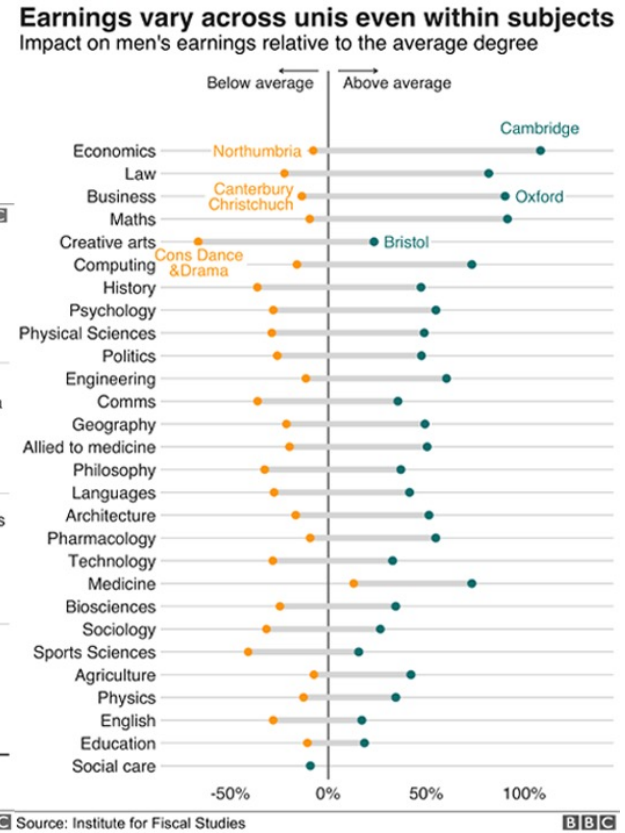
http://socviz.co/groupfacettx.html



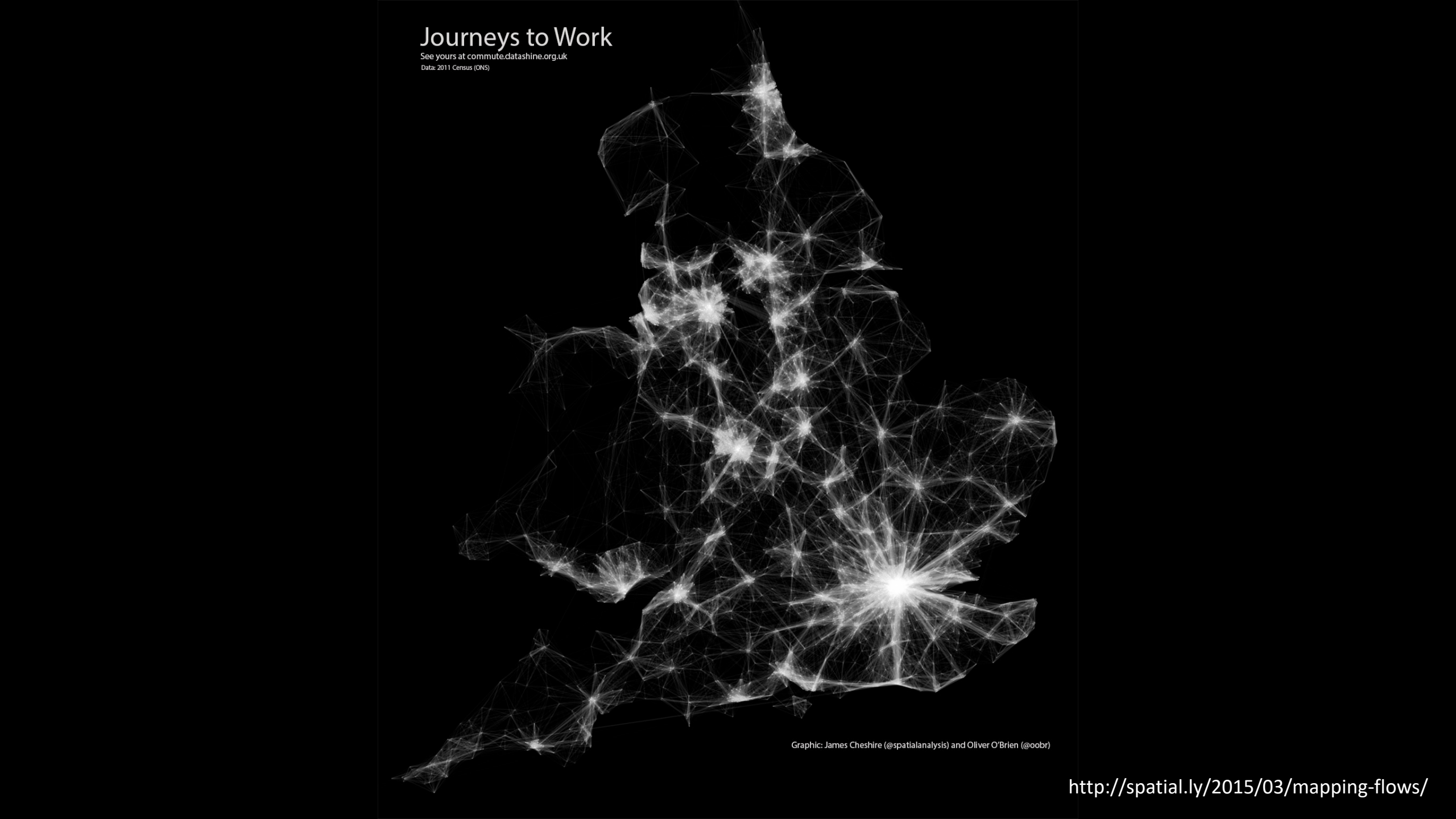




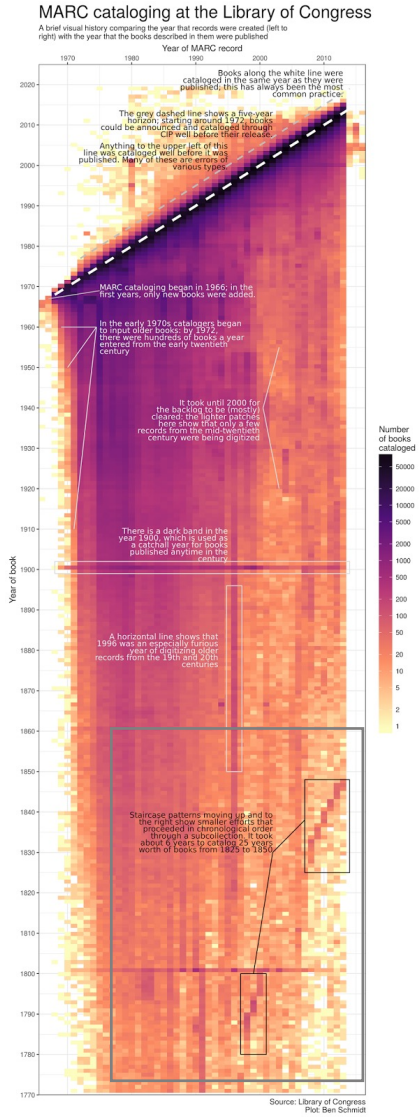


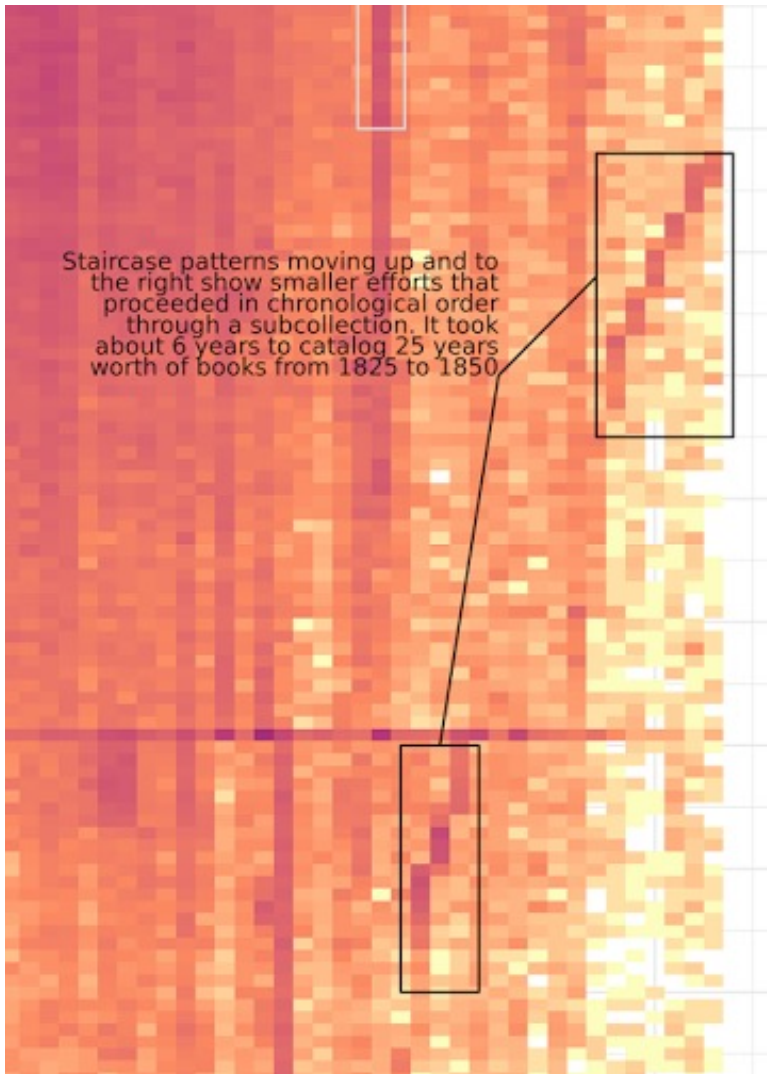


https://bbc.github.io/rcookbook/



http://spatial.ly/2015/03/mapping-flows/





http://sappingattention.blogspot.com/2017/05/a-brief-visual-history-of-marc.html

Working in RStudio

Get workshop files

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

On GitHub:

- Click green “Code” button and select “Download ZIP”
- Unzip files on your laptop
  - Windows: Double-click, then look for “Extract Files” at the top
  - Mac: Double-click

In RStudio:

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

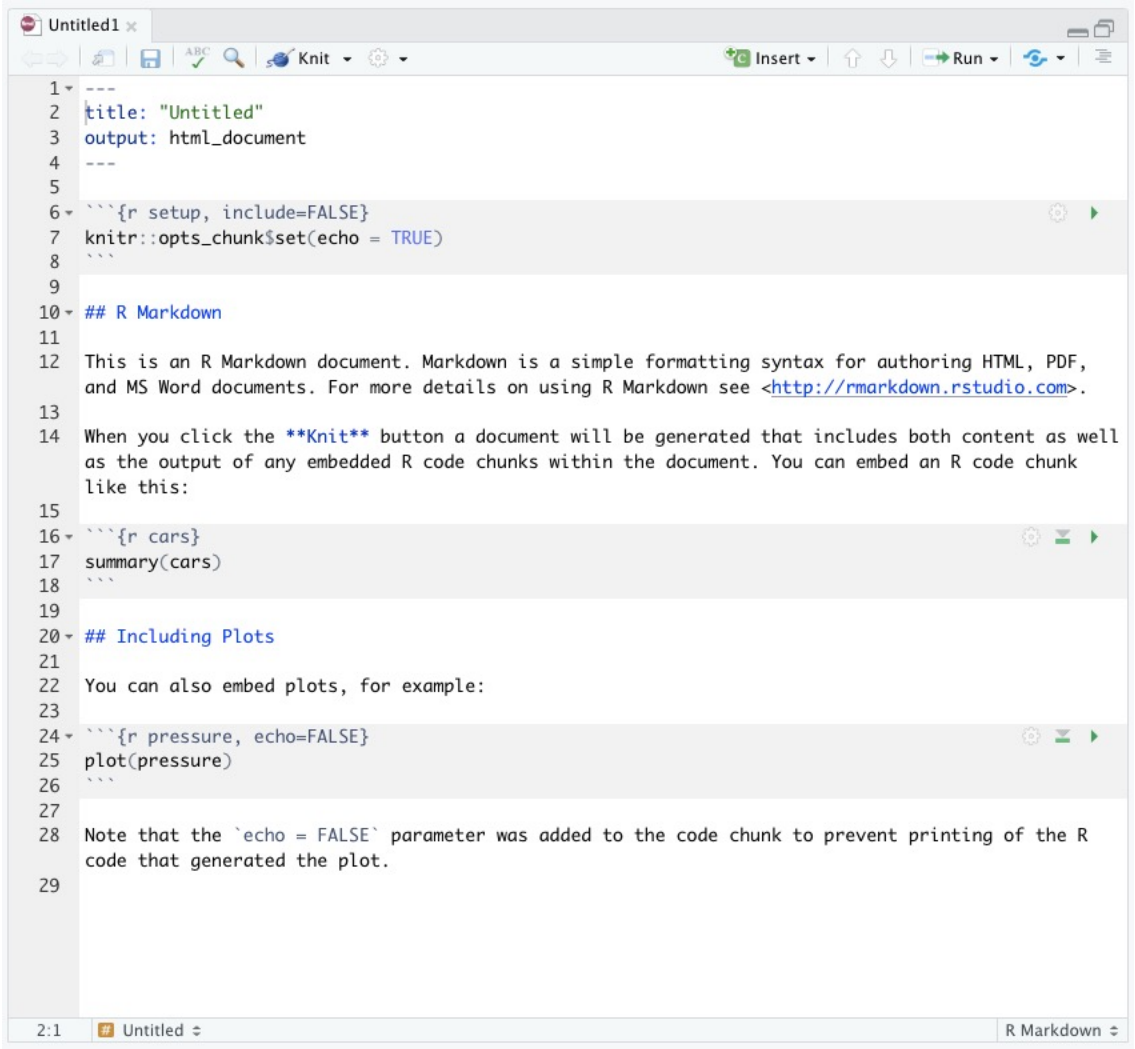
Why R Markdown?

- 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.

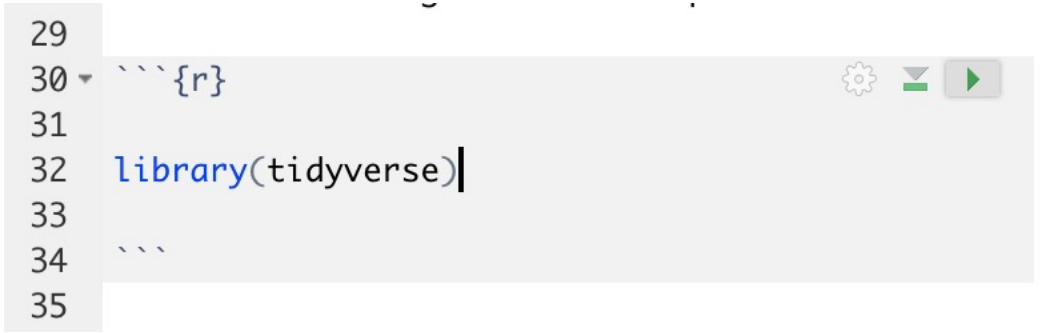
R Markdown files

- First few lines are called YAML header, set up some properties of file
- R code goes inside code chunks
- Text in Markdown syntax goes in between code chunks
- Use the “play” button to run individual code chunks
- Knit or run all to run the entire document



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

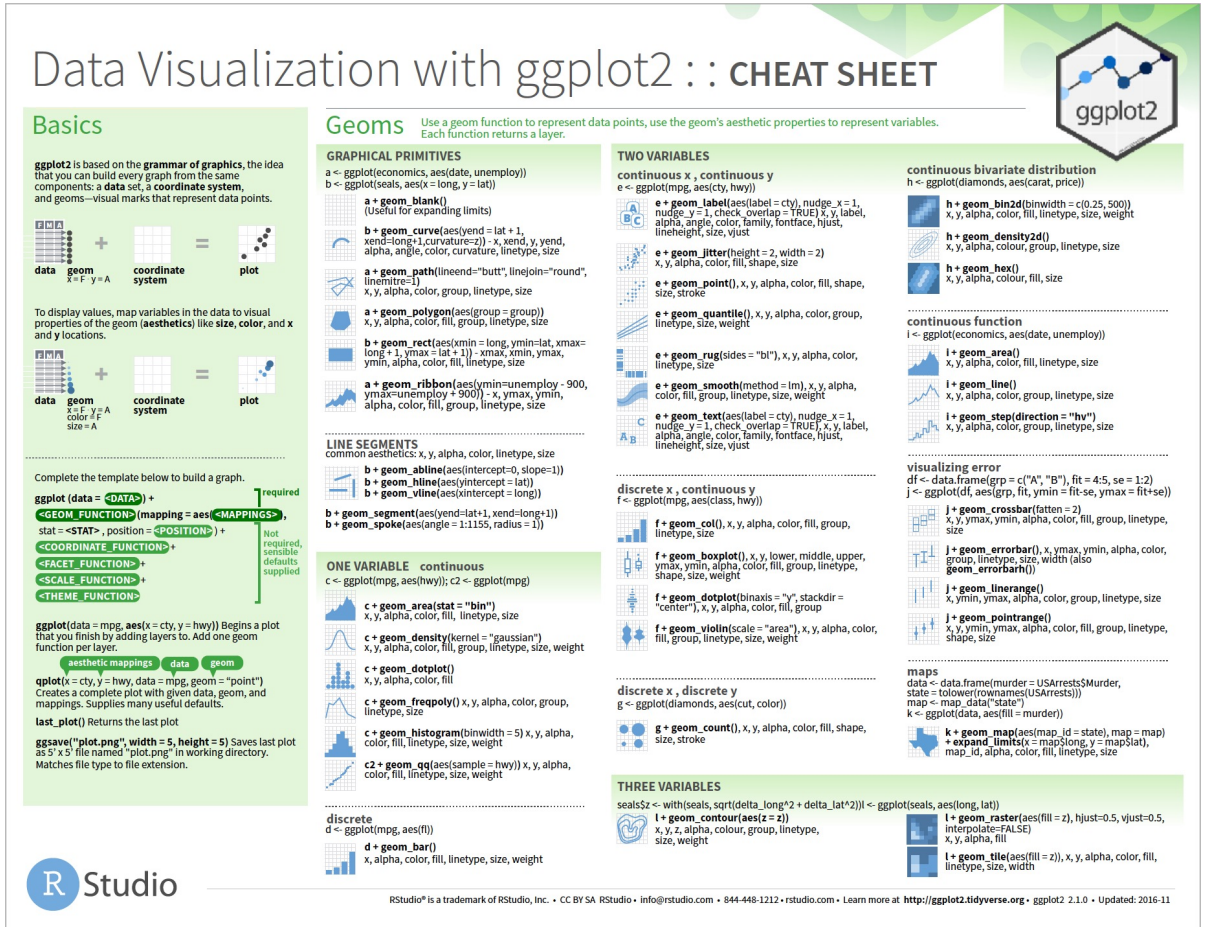


ggplot2 Cheat Sheet

Help →

Cheatsheets →

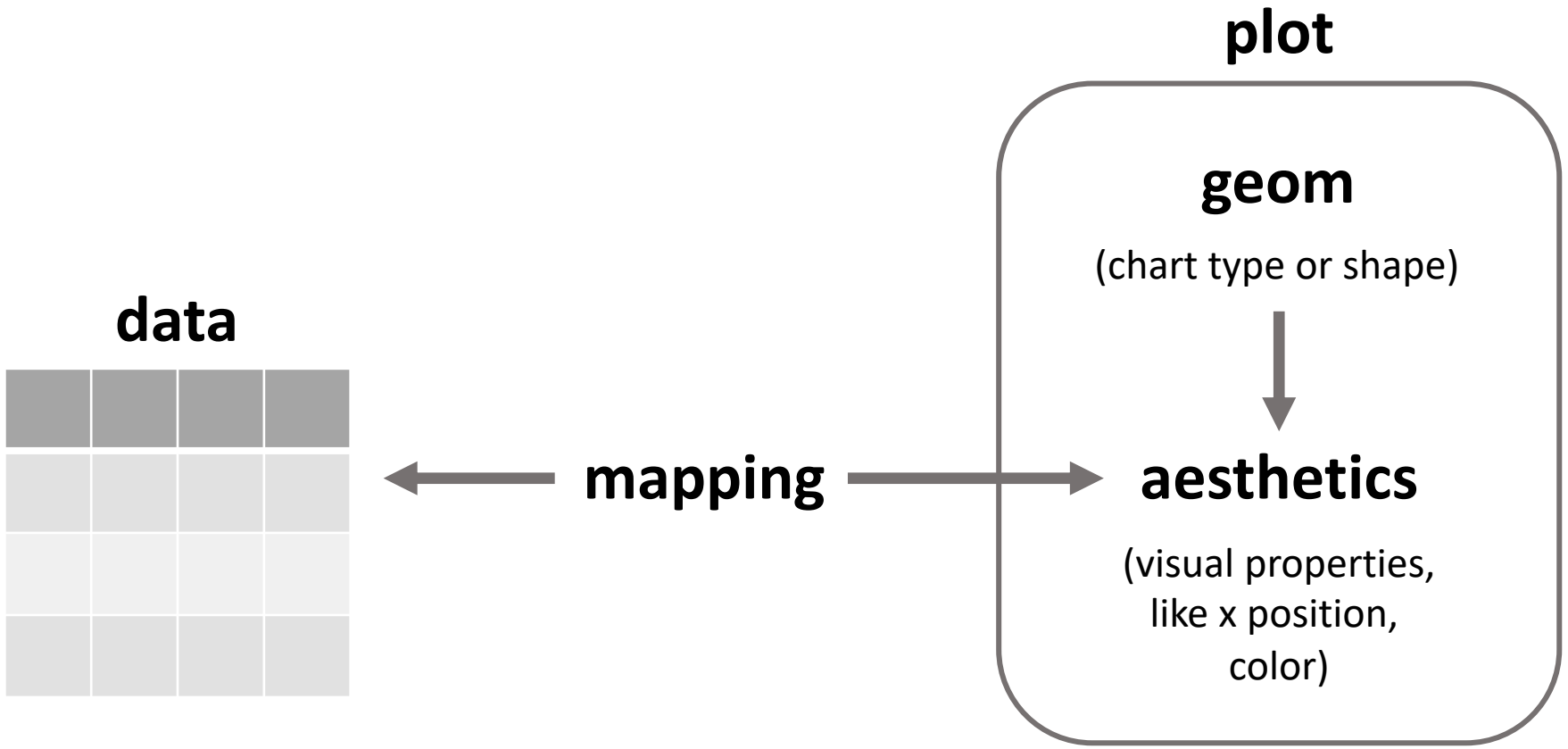
Data Visualization with ggplot2



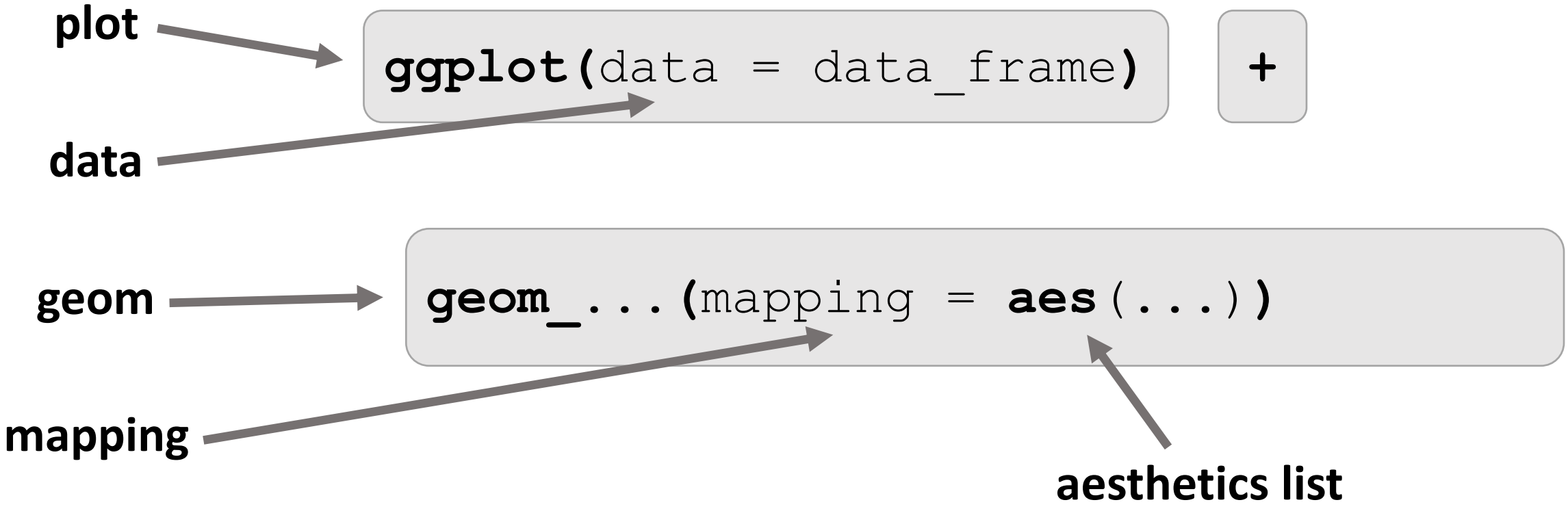
https://www.rstudio.com/resources/cheatsheets/#ggplot2

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"

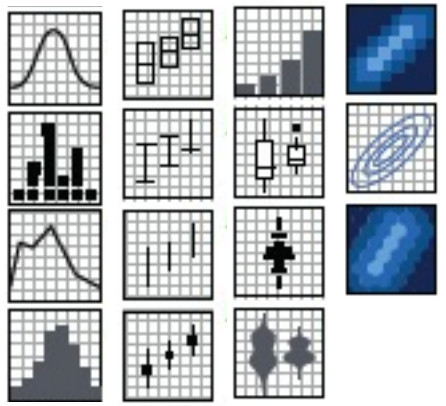
Petal.Width	Petal.Length	Species
0.3	1.4	setosa
1.3	4.0	versicolor
2.1	5.7	virginica

```
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.



http://bit.ly/ggplot2-cheatsheet

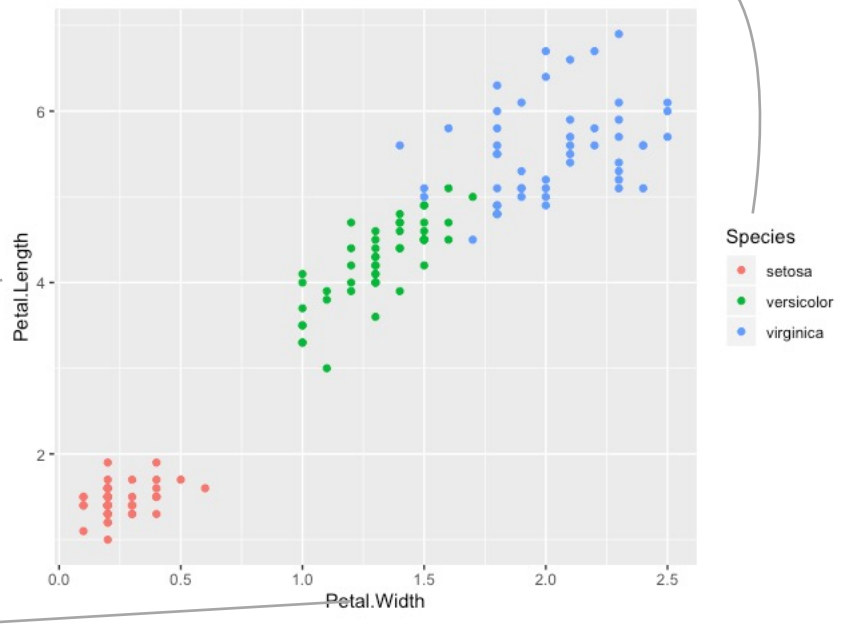
3. Map variables to aesthetics

"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))
```

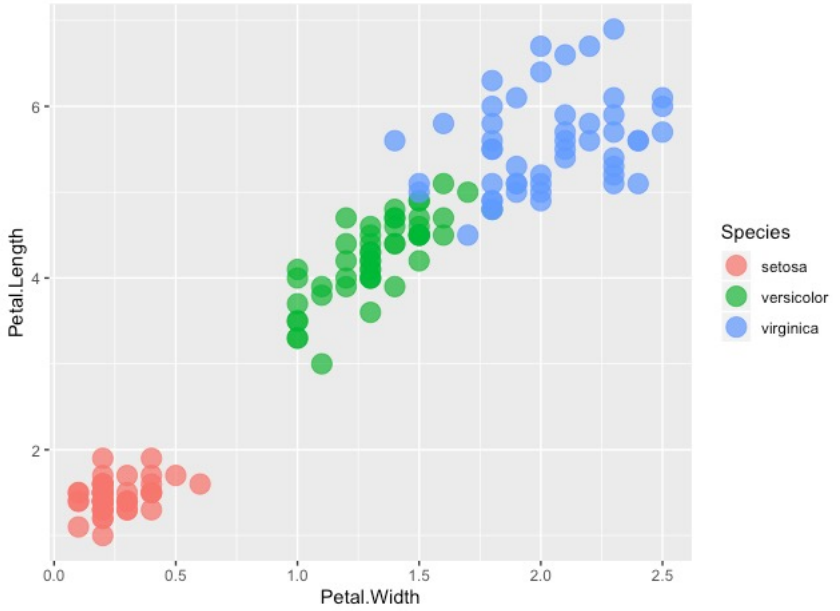


4. Add non-variable adjustments

"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)
```



Debugging code

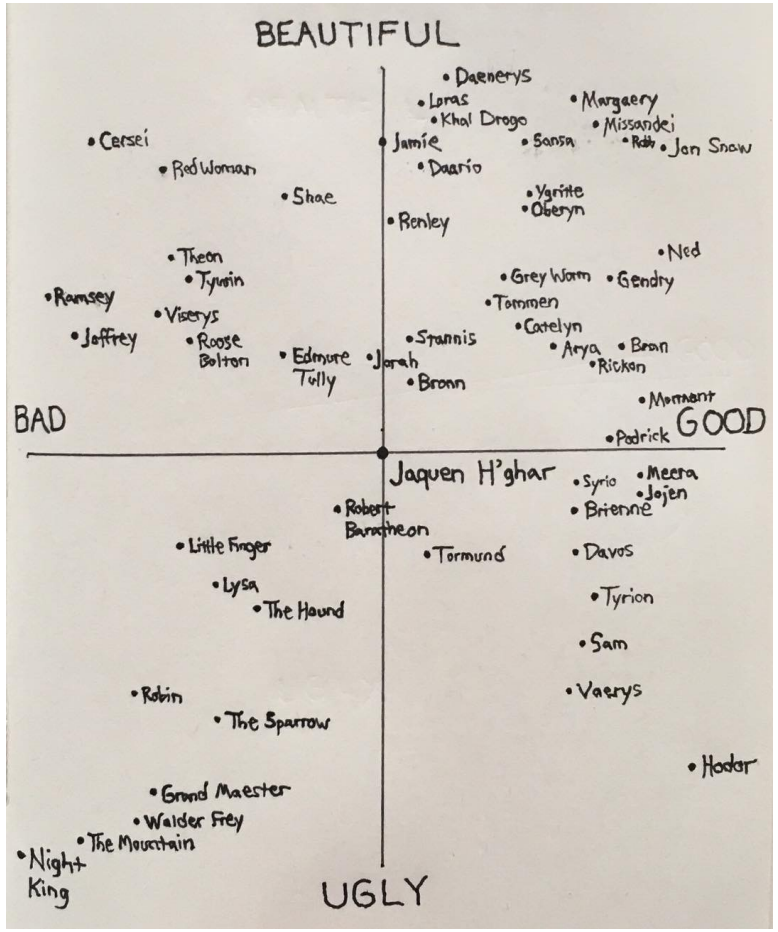
- Start simple
- If you see an error:
  - read error message for hints
  - check for problems with spelling/punctuation marks
- Get code to run without errors
- Check result to see if it makes sense

- Add a small change
- Get code to run without errors
- Check result to see if it makes sense
- etc.

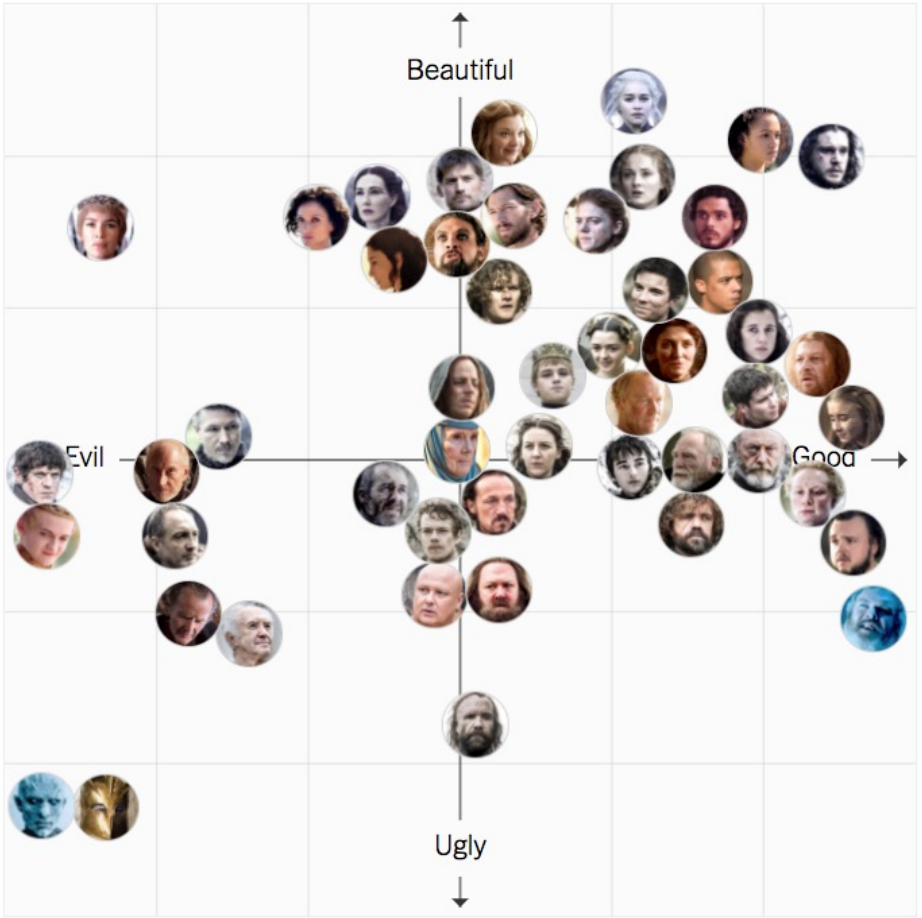
Exercise 1:
Game of Thrones character ratings

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

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)
```

+

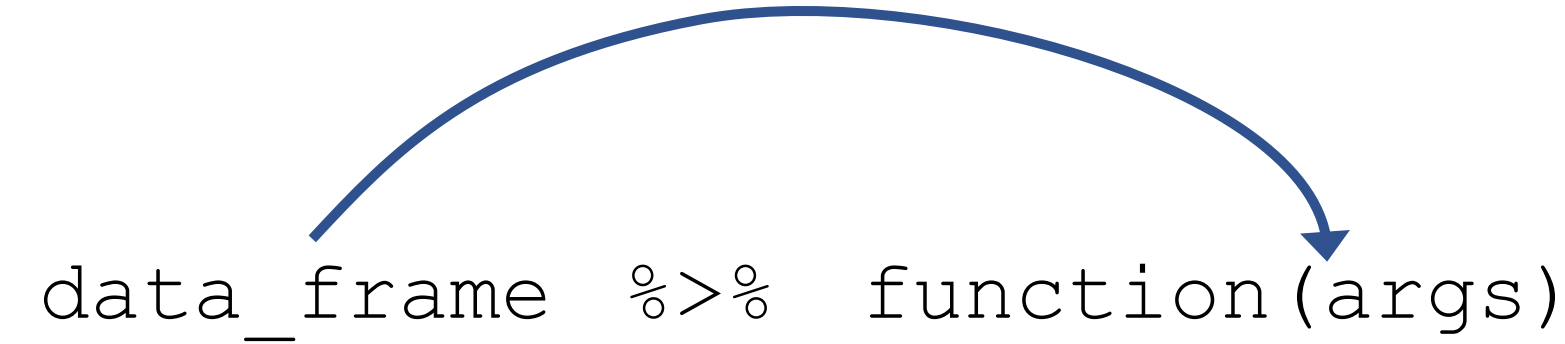
shape layer

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

Data aggregation

About %>%

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



drop_na

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

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

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

https://www.rstudio.com/resources/cheatsheets/ (Data Import with 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(sex)
```

sex	n
m	23
f	45

```
data %>% count(sex, marital_status)
```

sex	marital_status	n
m	married	18
m	unmarried	5
f	married	31
f	unmarried	14

https://www.rstudio.com/resources/cheatsheets/ (Data Transformation Cheatsheet)

Exercise 2:
Git Experience data

https://osf.io/57tb8/

Working with text variables

Text variables

In R, "character" variables

Gender	Age	Household Income	Education
Response	Response	Response	Response
Male	18-29		High school degree
Male	18-29	$0 - $24,999	Bachelor degree
Male	18-29	$0 - $24,999	High school degree
Male	18-29	$100,000 - $149,999	Some college or Associate degree
Male	18-29	$100,000 - $149,999	Some college or Associate degree
Male	18-29	$25,000 - $49,999	Bachelor degree
Male	18-29		High school degree
Male	18-29		High school degree
Male	18-29	$0 - $24,999	Some college or Associate degree
Male	18-29	$25,000 - $49,999	Some college or Associate degree
Male	18-29	$25,000 - $49,999	Bachelor degree
Male	30-44	$50,000 - $99,999	Graduate degree
Male	18-29		High school degree
Male	18-29	$0 - $24,999	Some college or Associate degree
Male	18-29	$50,000 - $99,999	Bachelor degree

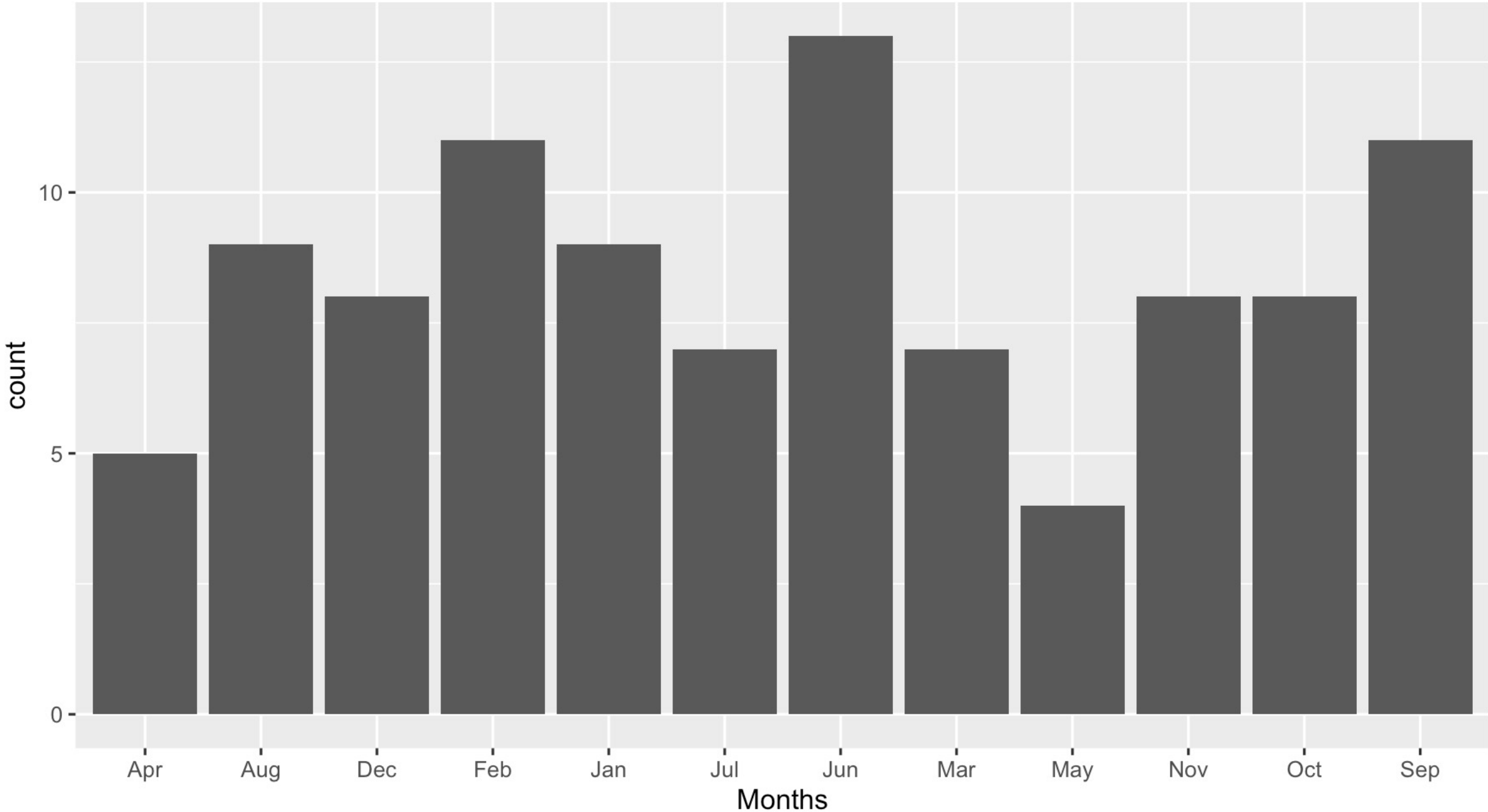
Problems with text variables: Ordering

Factors

- Default ordering for categories: **alphabetical**
- Converting to factor allows you to:
  - Specify “levels” for a categorical variable
  - Specify the order of those levels
  - Specify whether the factor is “ordered”

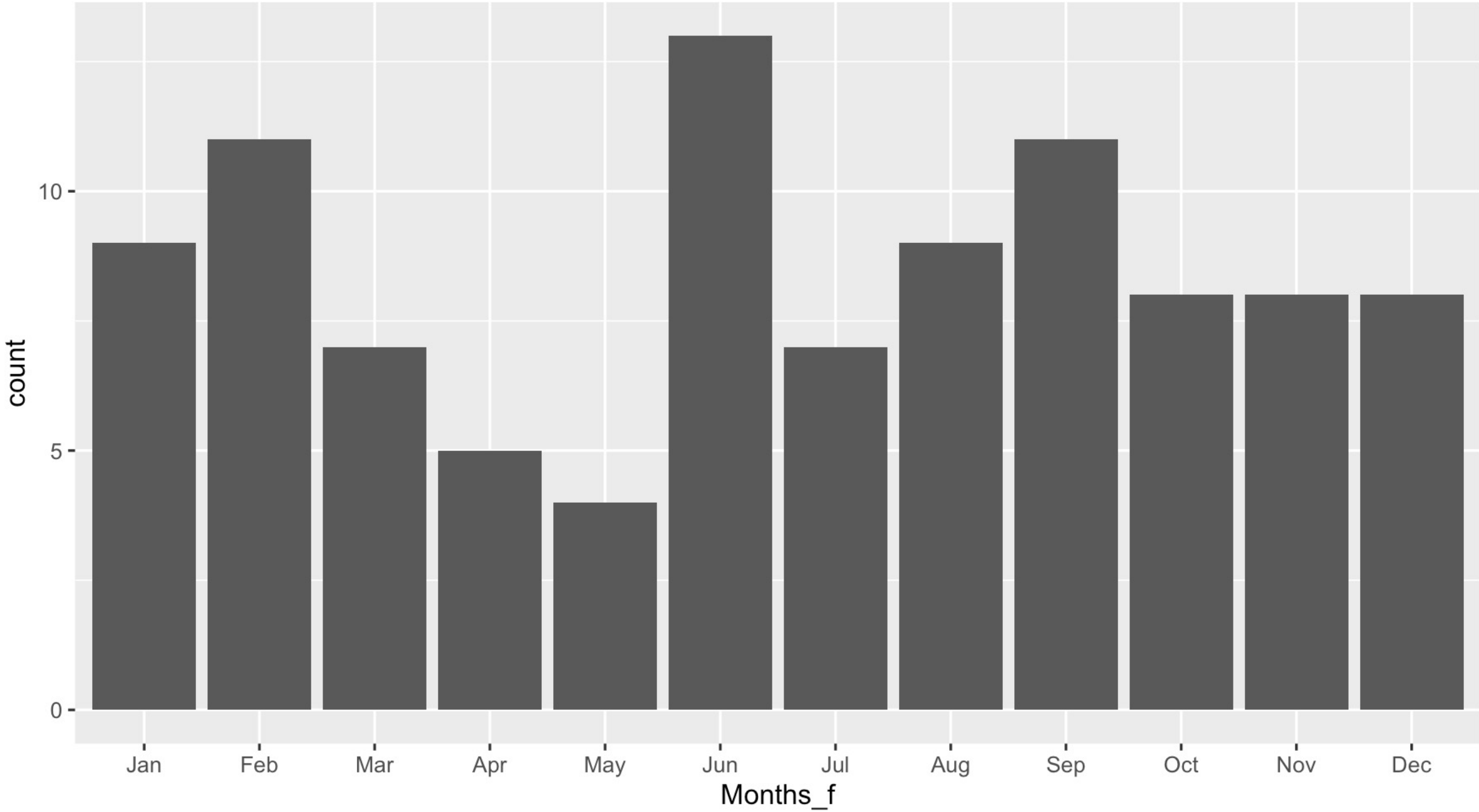
https://r4ds.had.co.nz/factors.html

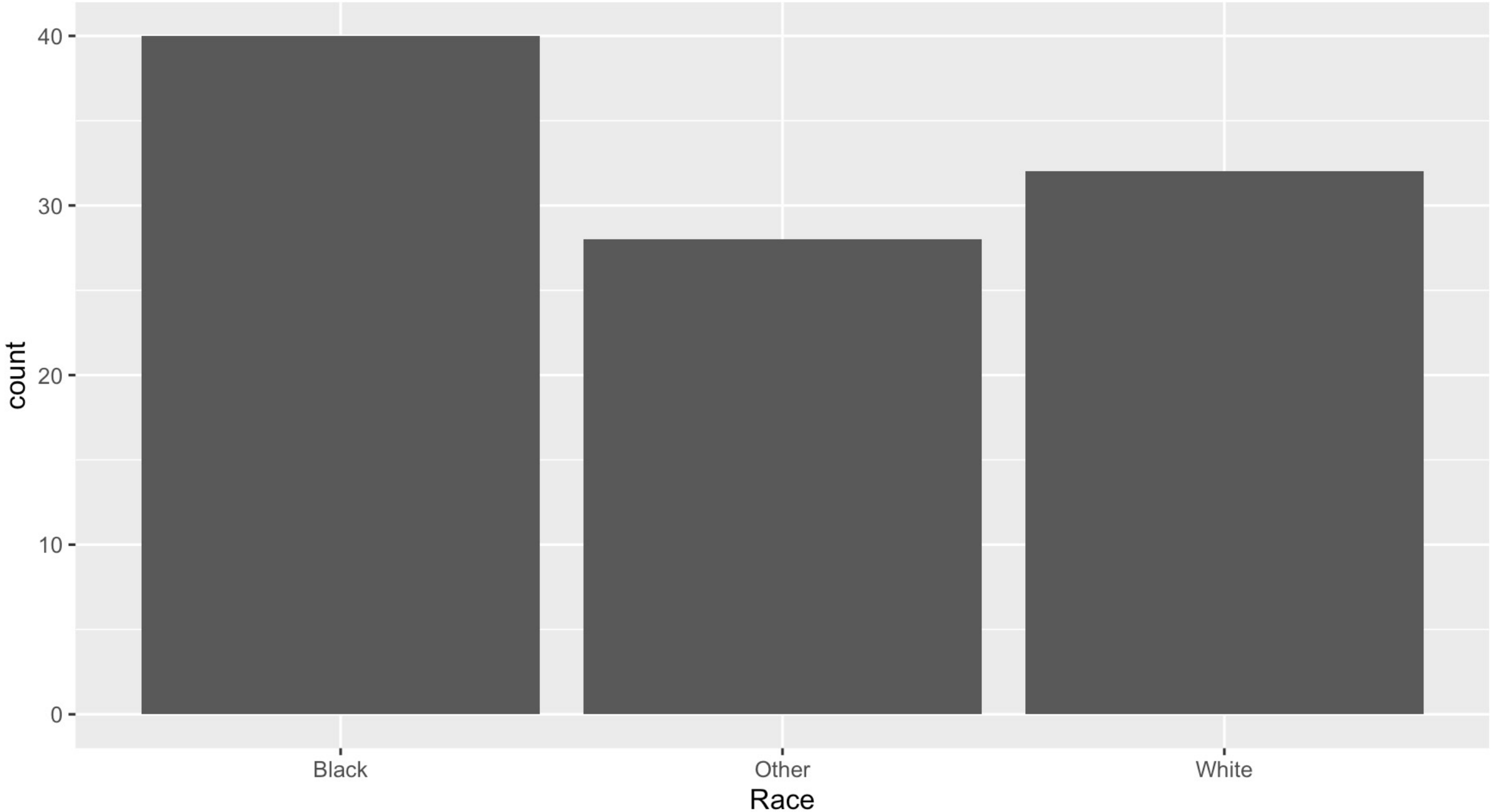
```
> x1 <- c("Dec", "Apr", "Jan",
"Mar")

> factor(x1)
[1] Dec Apr Jan Mar
Levels: Apr Dec Jan Mar

> month_levels <- c( "Jan", "Feb",
"Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec" )

> y1 <- factor(x1,
               levels = month_levels)
> y1
[1] Dec Apr Jan Mar
Levels: Jan Feb Mar Apr May Jun Jul
Aug Sep Oct Nov Dec
```


count
10
5
0
Apr
Aug
Dec
Feb
Jan
Jul
Jun
Mar
May
Nov
Oct
Sep
Months

Order by meaning

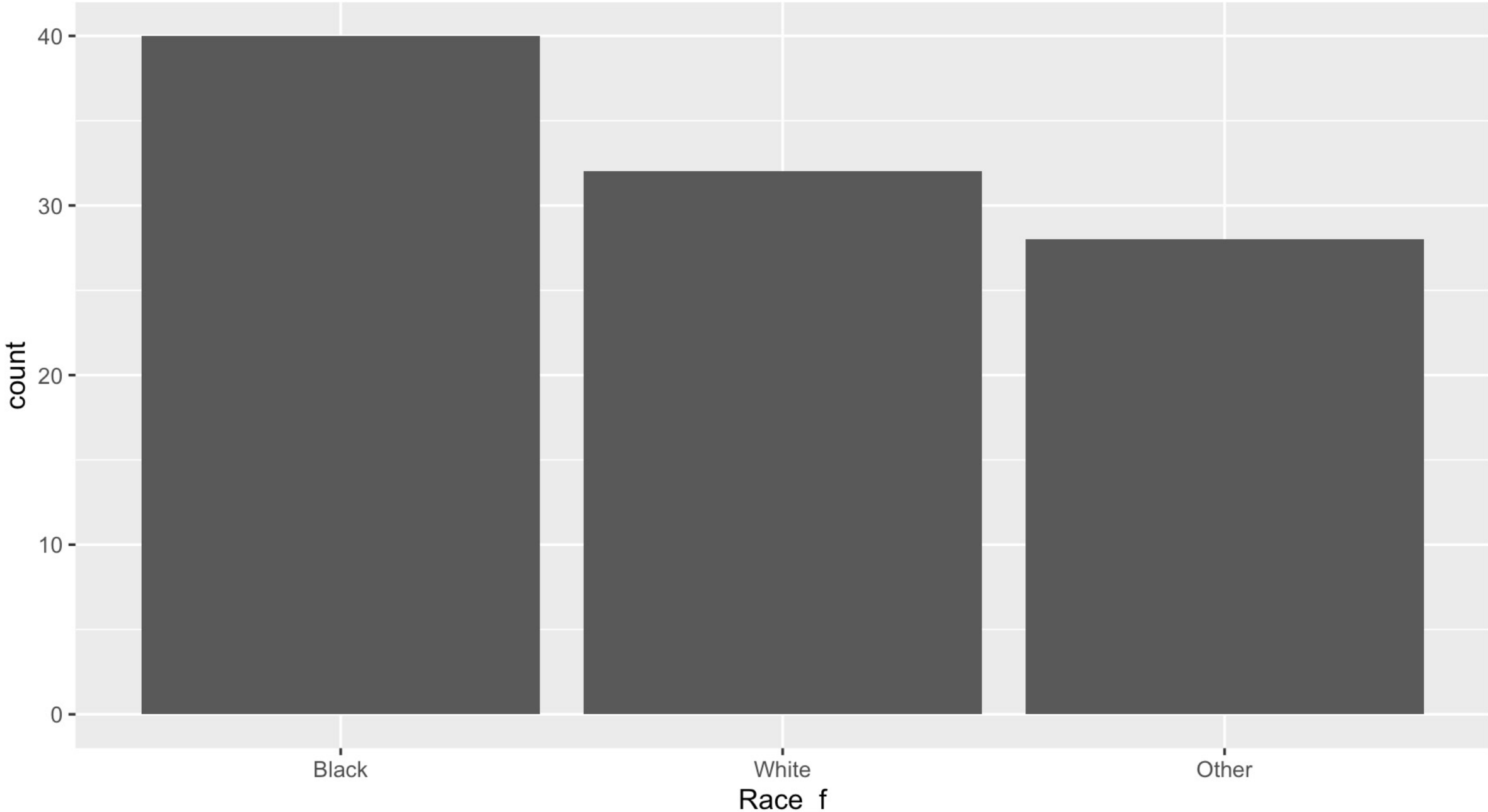
```
month_levels <- c( "Jan", "Feb", "Mar", "Apr",
"May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
"Dec" )


data <- data %>%
    mutate(Months_f = Months %>%
                      as_factor() %>%
                      fct_relevel(month_levels))
```


count
10
5
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Months_f


40
30
20
10
0
count
Black
Other
White
Race

Order by value (using forcats)

```
demo <- data %>%
     mutate(Race_f = Race %>%
                        as_factor() %>%
                        fct_infreq())


ggplot(data,
         aes(Race %>%
                  as_factor() %>%
                  fct_infreq())) +
     geom_bar()
```


40
30
20
10
0
count
Black
White
Other
Race_f

forcats package: helpful functions

- `as_factor(char_var)`:
  convert a character variable to a factor
- `fct_infreq(factor)`:
  take factor levels and set the order according to (inverse) category frequency
- `fct_reorder(factor, num_var)`:
  sort factor levels by a second, numerical variable (like a pre-calculated count or average)

https://www.rstudio.com/resources/cheatsheets/#forcats

Note about read.csv (base R)

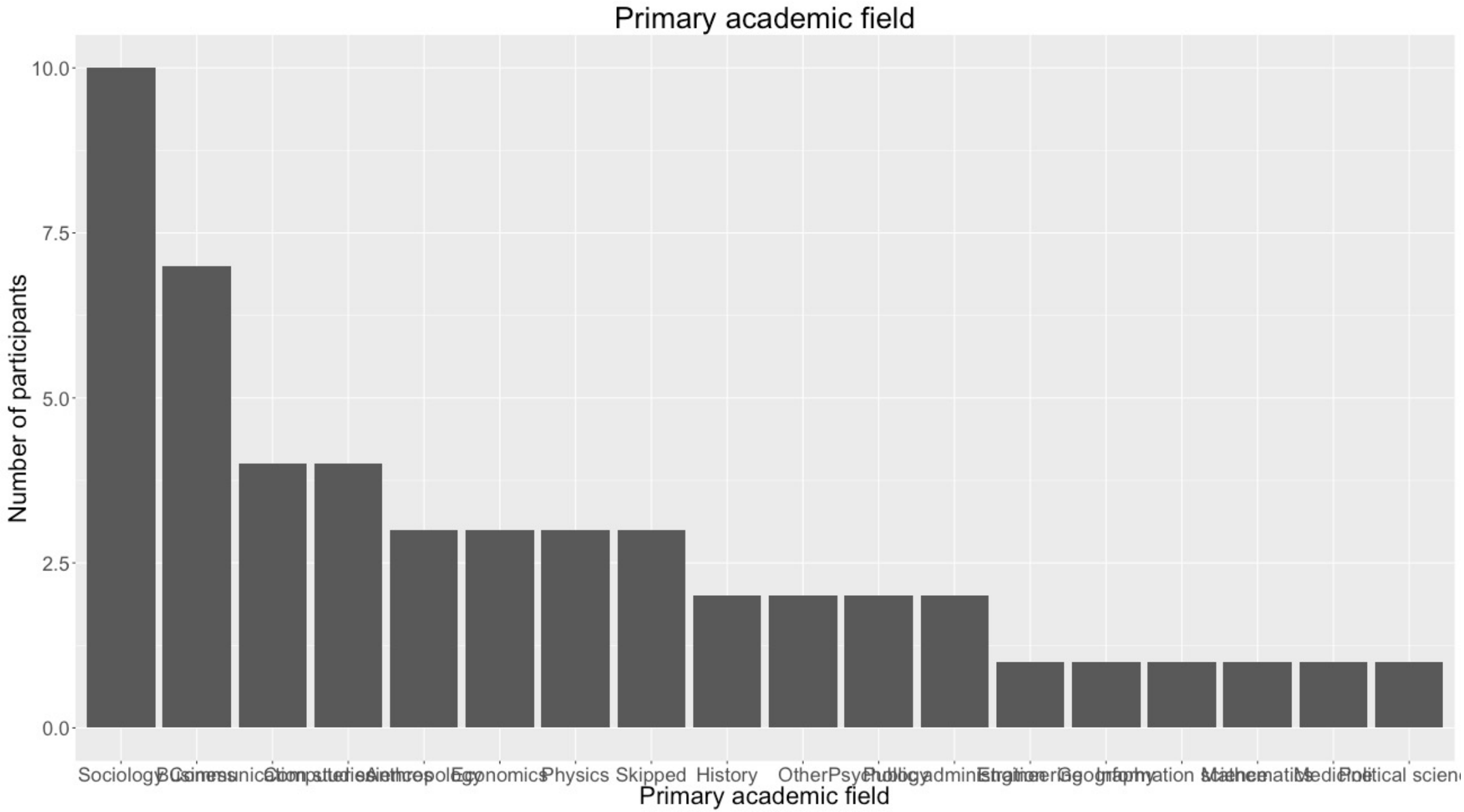
- Converts string variables to factors by default
- Can either:
  - Include `stringsAsFactors=FALSE`
  - Use `read_csv()` instead

Factoring resources

From Amelia McNamara:

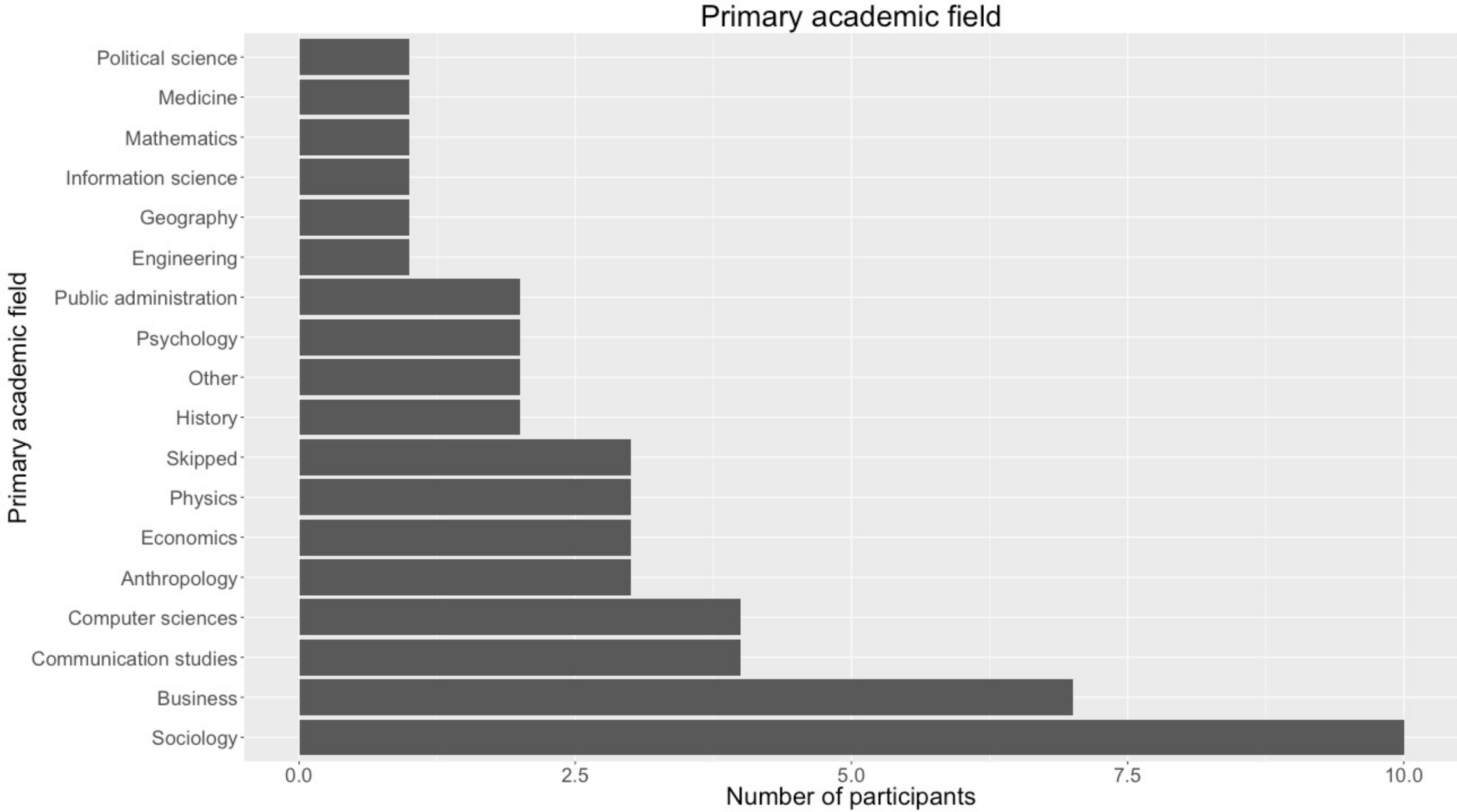
- RStudioConf 2019 slides:
  Working with Categorical Data in R Without Losing Your Mind
- Wrangling Categorical Data in R article
- Wrangling Categorical Data in R repository

Problems with text variables: Long category names


Primary academic field
Number of participants
10.0
7.5
5.0
2.5
0.0
Sociology
Business
Computer science
Anthropology
Economics
Physics
Skipped
History
Other
Psychology
Public administration
Engineering
Geography
Information science
Mathematics
Medicine
Political scien
Primary academic field

In ggplot2, have to flip the axes

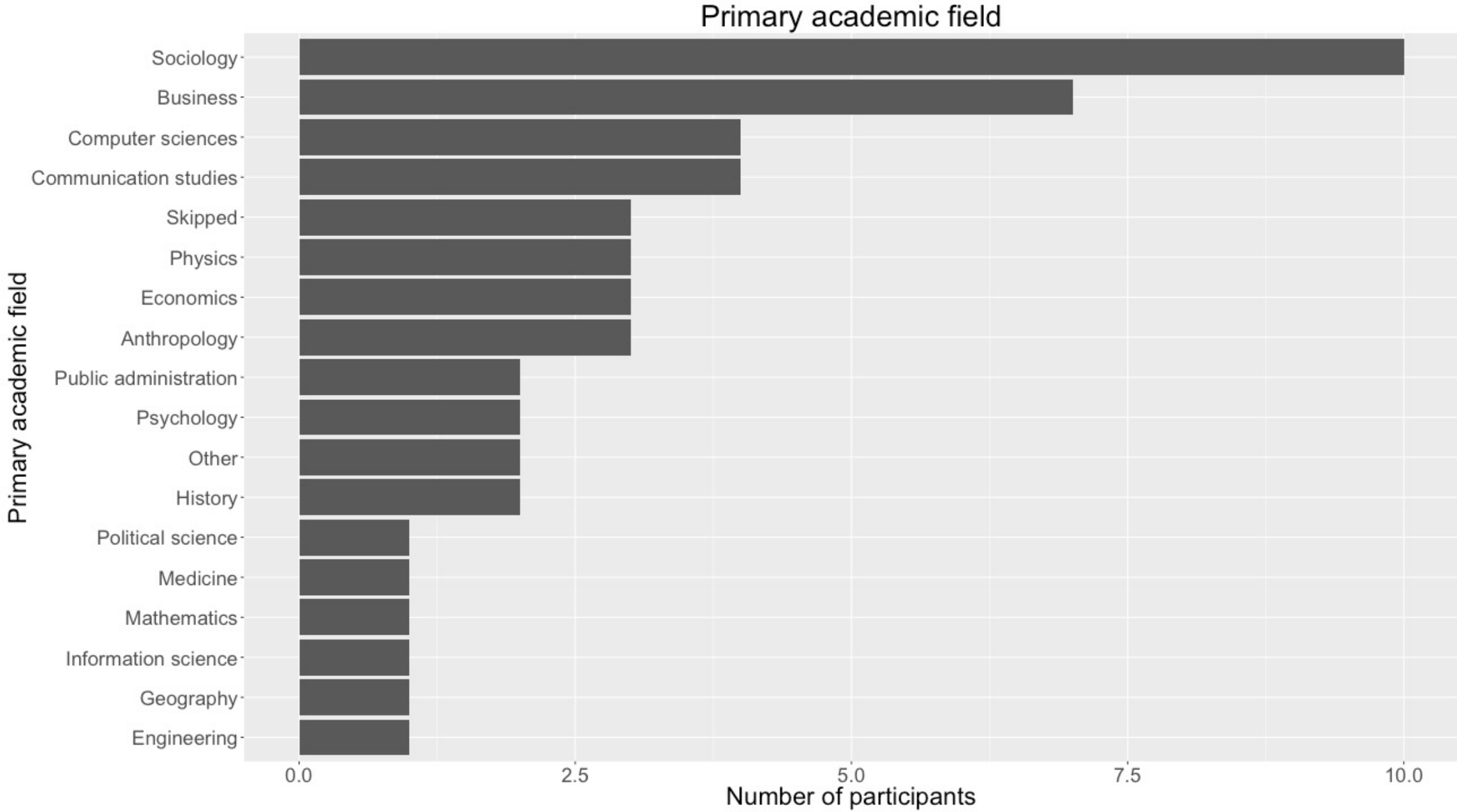
```
+ coord_flip()


or


ggplot(df, aes(y=cat_variable)) +
    geom_bar()
```


Primary academic field
Political science
Medicine
Mathematics
Information science
Geography
Engineering
Public administration
Psychology
Other
History
Skipped
Physics
Economics
Anthropology
Computer sciences
Communication studies
Business
Sociology
Primary academic field
0.0
2.5
5.0
7.5
10.0
Number of participants

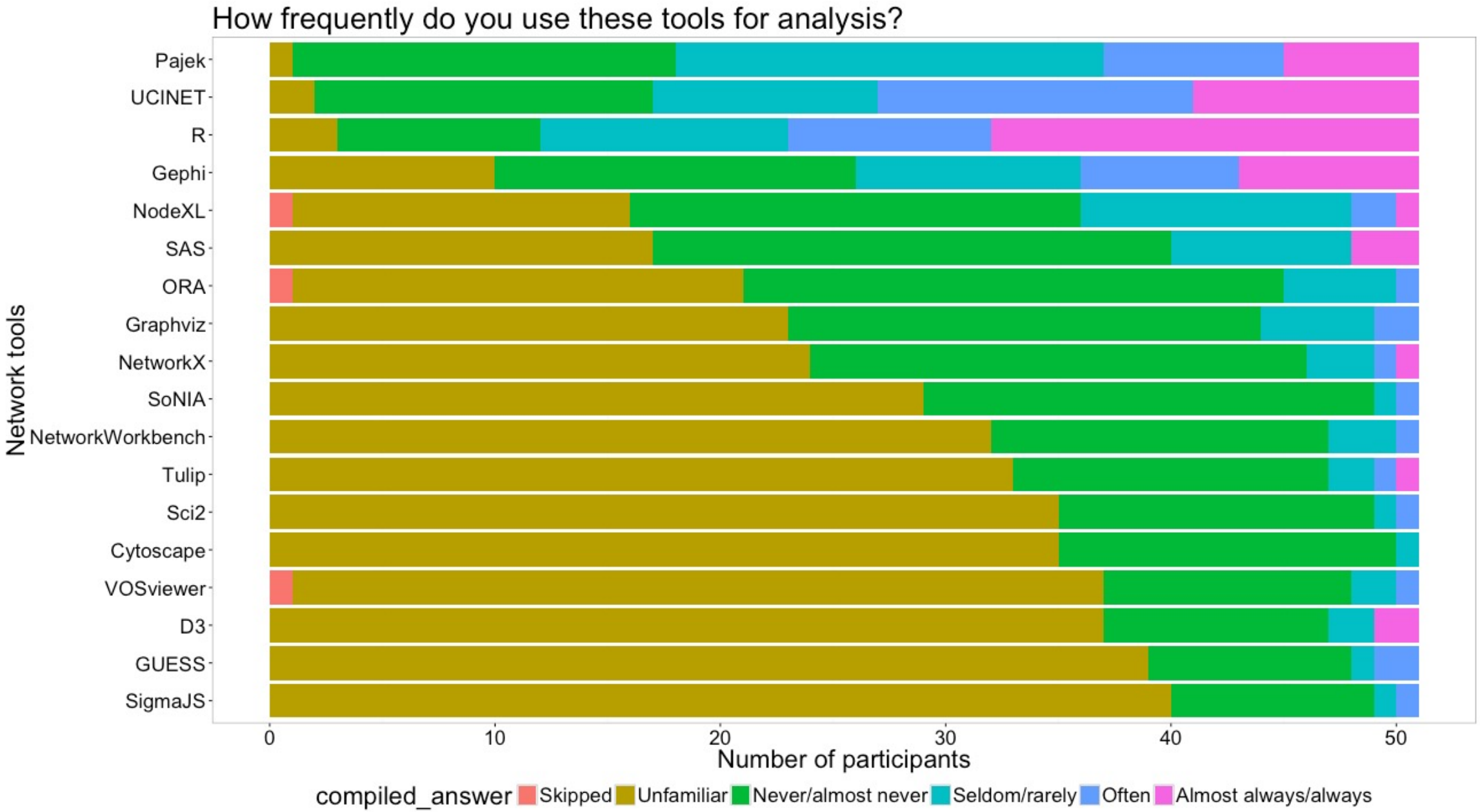
When you flip axes, you sort the other way

```
academic_field %>%
    as_factor() %>%
    fct_infreq() %>%
    fct_rev()
```

Have to reverse the order of the levels

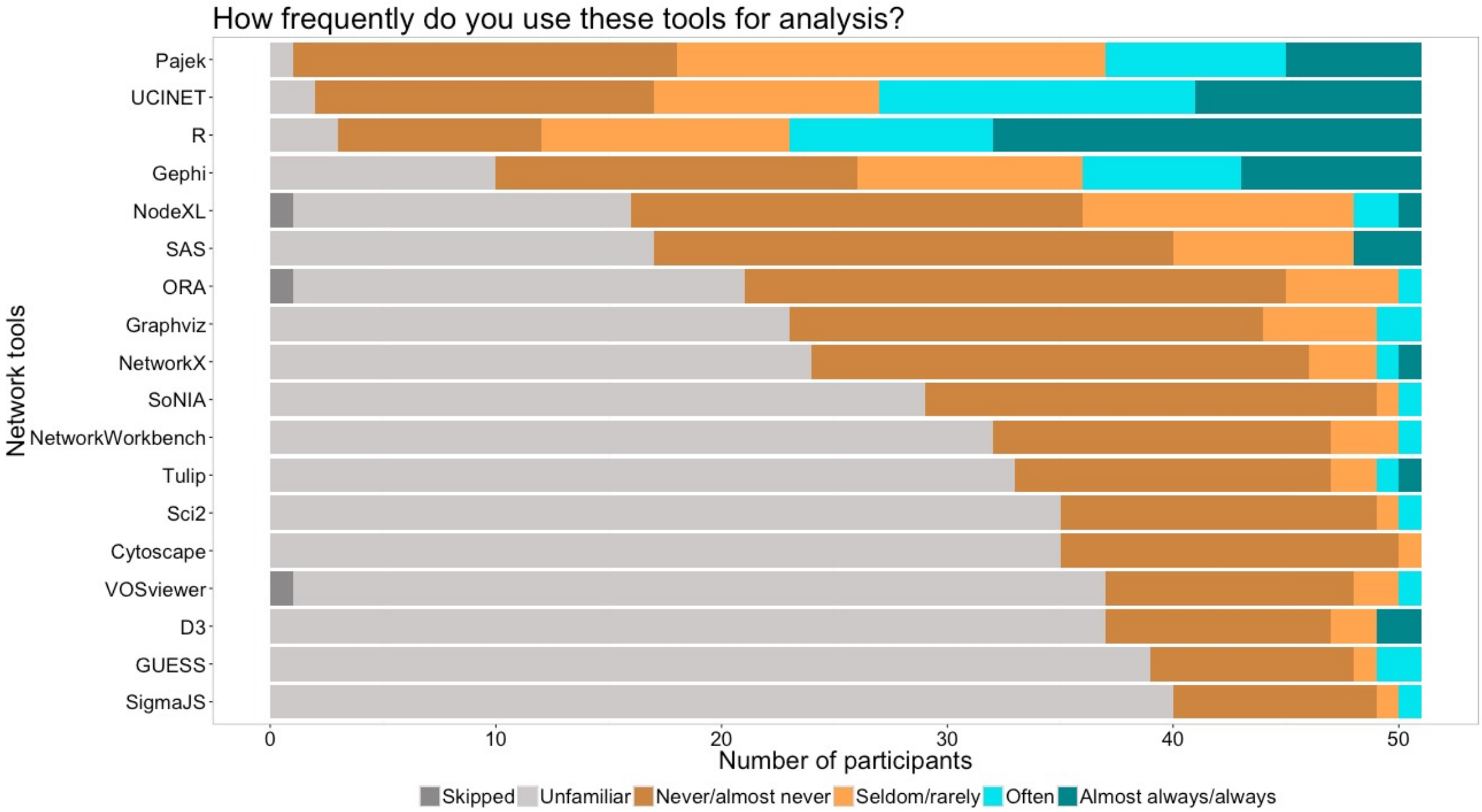

Primary academic field
Sociology
Business
Computer sciences
Communication studies
Skipped
Physics
Economics
Anthropology
Public administration
Psychology
Other
History
Political science
Medicine
Mathematics
Information science
Geography
Engineering
Primary academic field
0.0
2.5
5.0
7.5
10.0
Number of participants

Problems with text variables: Arbitrary colors


How frequently do you use these tools for analysis?
Network tools
Pajek
UCINET
R
Gephi
NodeXL
SAS
ORA
Graphviz
NetworkX
SoNIA
NetworkWorkbench
Tulip
Sci2
Cytoscape
VOSviewer
D3
GUESS
SigmaJS
0
10
20
30
40
50
Number of participants
compiled_answer
Skipped
Unfamiliar
Never/almost never
Seldom/rarely
Often
Almost always/always

Select colors manually, or use alternate palette

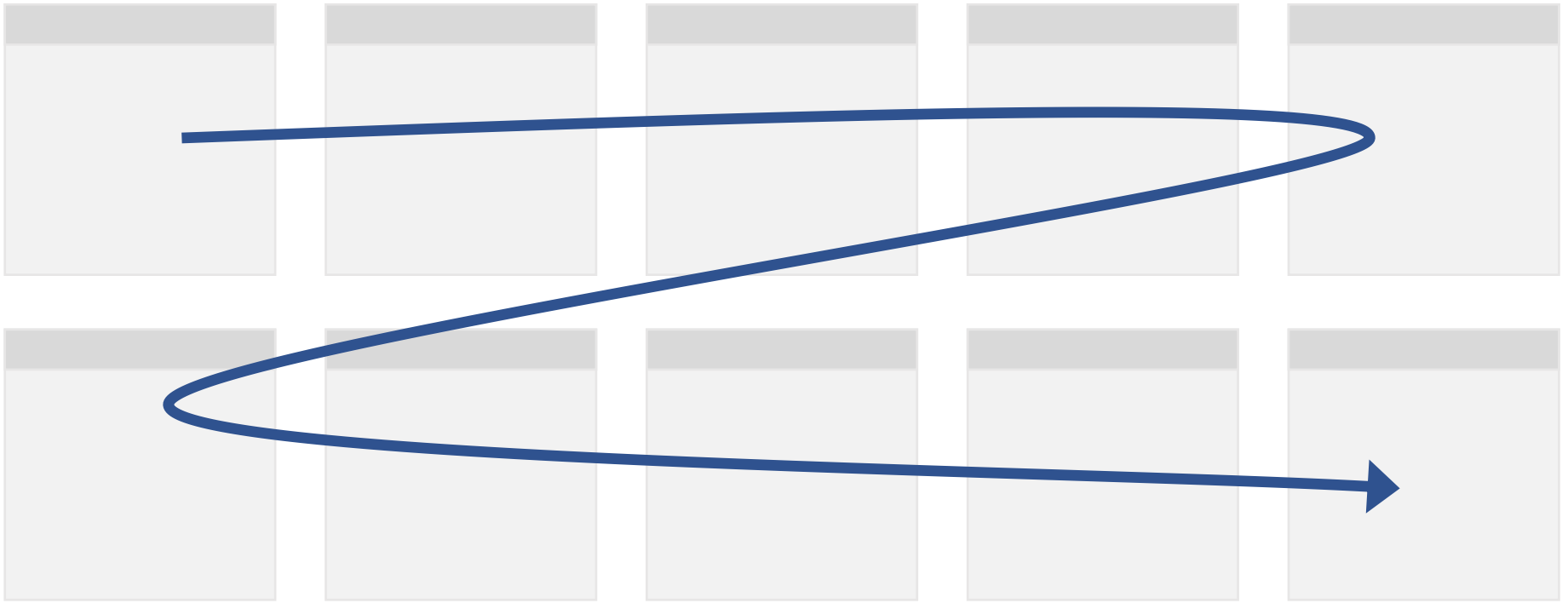
```
scale_fill_manual(
    values=c("snow4","snow3",
             "tan3","tan1",
             "turquoise2","turquoise4"))


# Also see package RColorBrewer
scale_fill_brewer(palette="BrBG")
```


How frequently do you use these tools for analysis?
Network tools
Pajek
UCINET
R
Gephi
NodeXL
SAS
ORA
Graphviz
NetworkX
SoNIA
NetworkWorkbench
Tulip
Sci2
Cytoscape
VOSviewer
D3
GUESS
SigmaJS
0
10
20
30
40
50
Number of participants
Skipped
Unfamiliar
Never/almost never
Seldom/rarely
Often
Almost always/always

Creating repeated charts

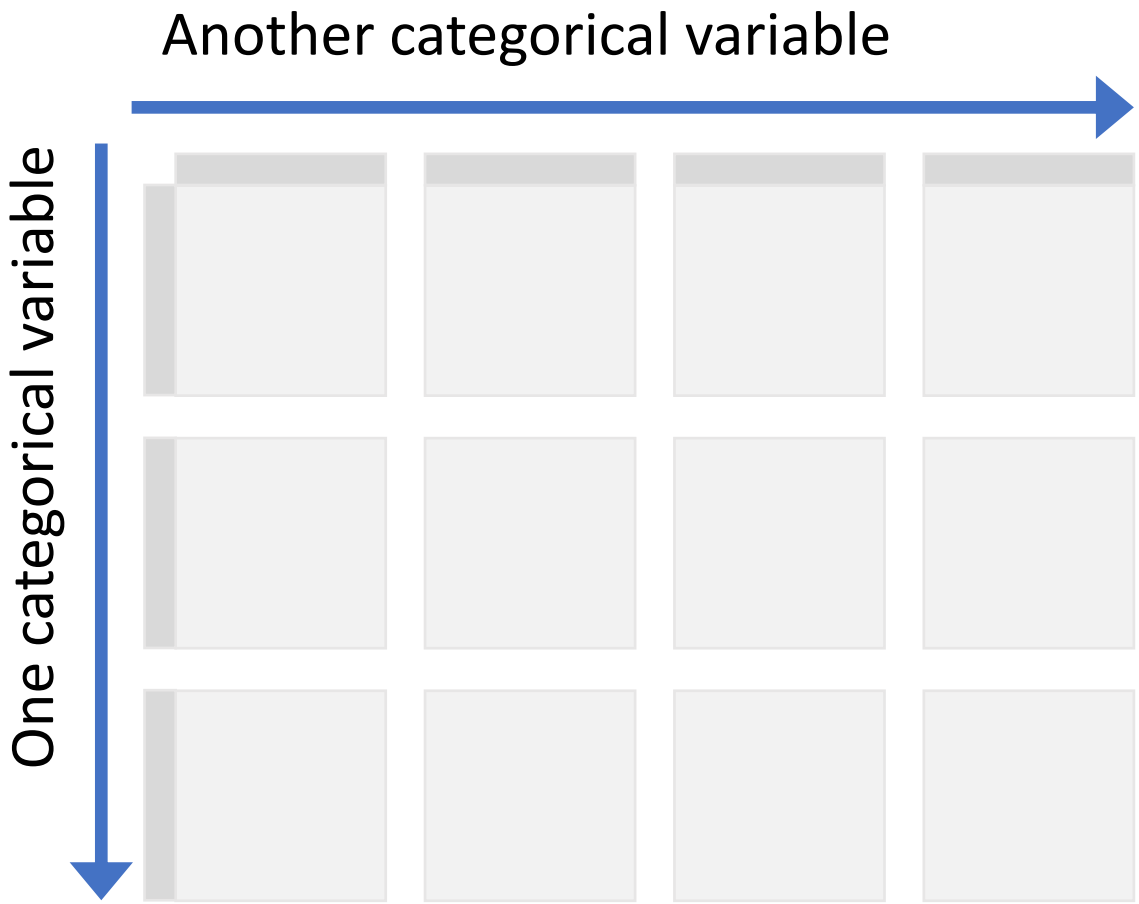
facet_wrap()

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



facet_grid()

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



Exercise 3: Inclusiveness Index

https://belonging.berkeley.edu/inclusivenessindex

Helpful data manipulation

filter

Select a subset of rows

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

same as

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

https://www.rstudio.com/resources/cheatsheets/#dplyr

select

Select a subset of columns (many options!)

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

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

https://www.rstudio.com/resources/cheatsheets/#dplyr

Exercise 4: Customizing charts

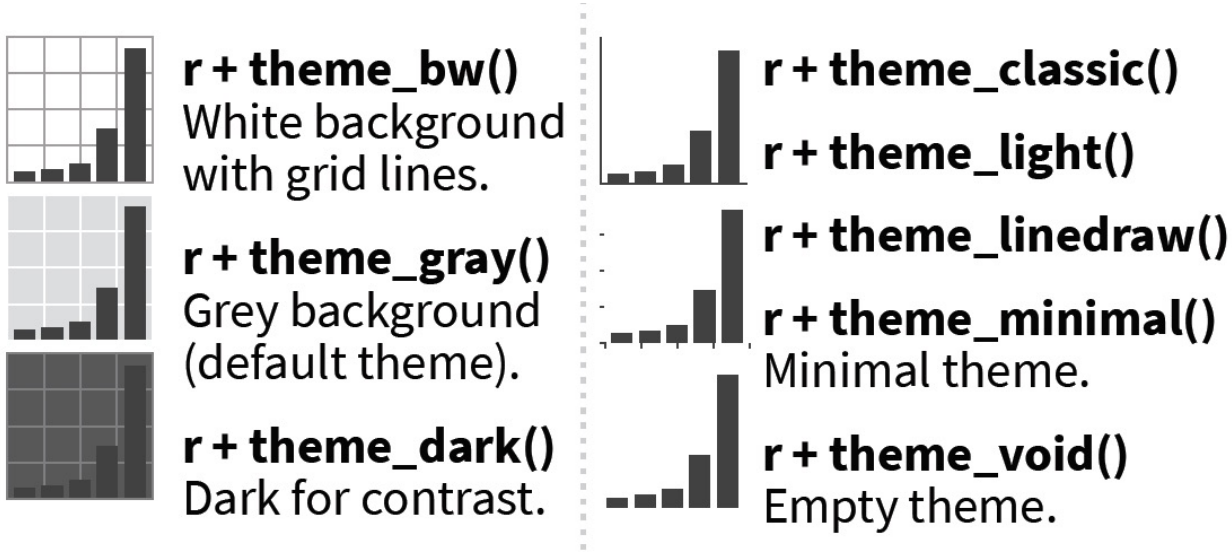
Scales

- 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



https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

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!**

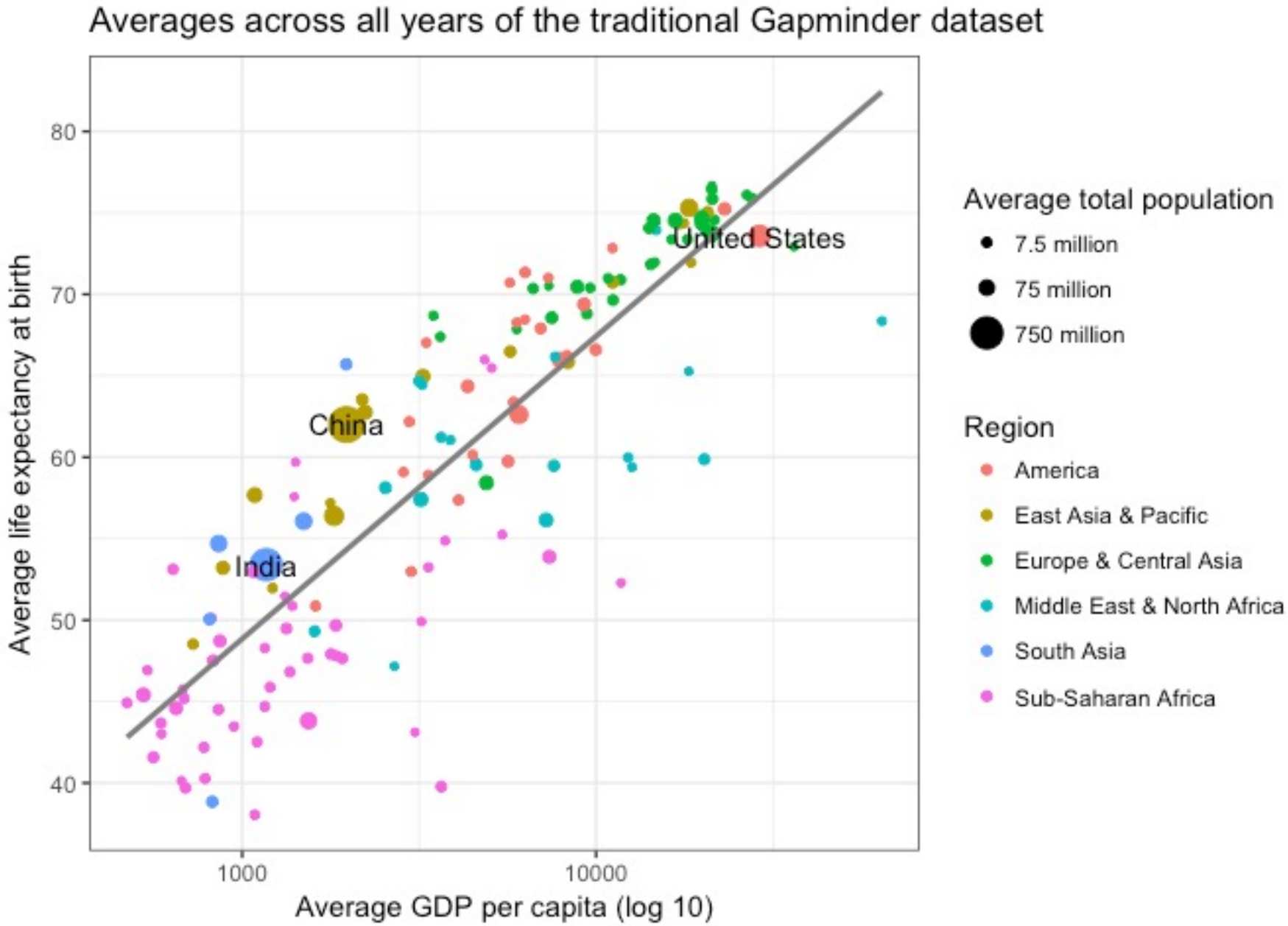
Advanced ggplot2 workshop

Workshop video

Workshop materials

Exercise 5: Gapminder Data

http://www.gapminder.org/


Averages across all years of the traditional Gapminder dataset
Average life expectancy at birth
80
70
60
50
40
1000
10000
Average GDP per capita (log 10)
China
India
United States
Average total population
7.5 million
75 million
750 million
Region
America
East Asia & Pacific
Europe & Central Asia
Middle East & North Africa
South Asia
Sub-Saharan Africa

Accessibility

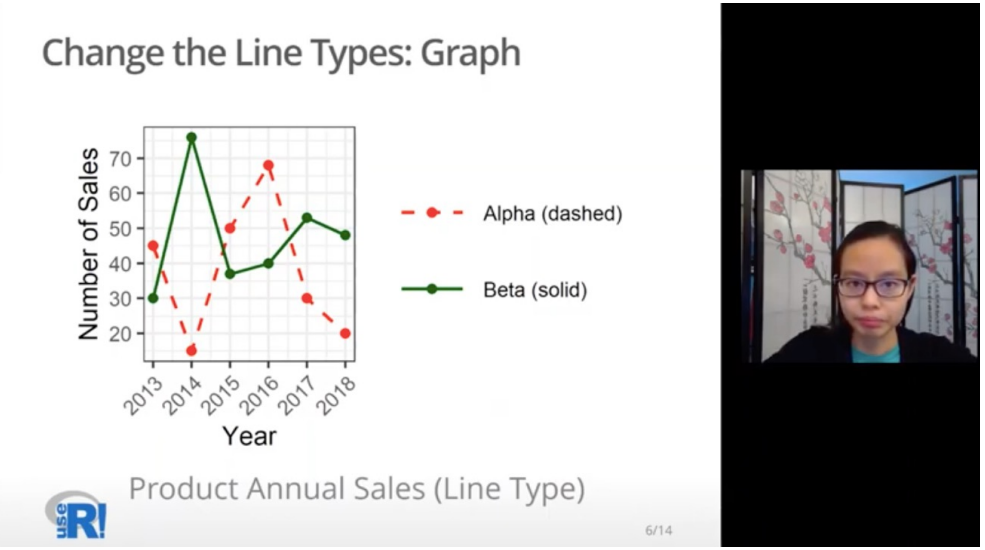
Low Vision

- Large text
  - “output-examples” file
- High color contrast
  - Both marks/text on background and labels on marks
  - Check with savonliquide package

Color Vision Deficiency

Dual encoding (never just color)

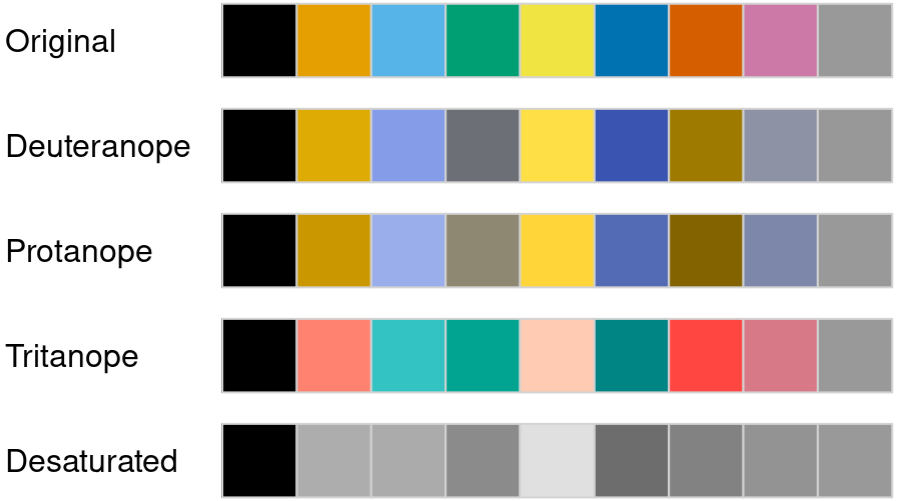
- Line color – also vary line type
- Point color – also vary point shape



https://www.youtube.com/watch?v=mbi_JVC1arM

Color palettes

- colorspace package



http://colorspace.r-forge.r-project.org/index.html

Alternative Text for Screen Readers

In R, RMarkdown

- fig.alt in code chunk (new, just for HTML output)
- fig.cap in code chunk as backup
- embedded images:
  ![alt text or image title](path/to/image)
- New: ggplot2 v3.3.4 adds alt option in labs(), with plans to propagate to Rmd, Shiny

Writing good alt text for visualizations

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

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

https://nightingaledvs.com/writing-alt-text-for-data-visualization/

Longer descriptions:
savonliquide package

Converting graphics to sound, touch, text

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

Accessible Data Science for the Blind Using R

Accessibility Resources

- savonliquide package
- Making betteR figures: Accessibility and Universal Design
- Highlights from the DVS accessibility fireside chat

Advanced topics:
Mapping, saving charts out

Mapping resources

- tigris for downloading TIGER/Line shapefiles
- sf (simple features) for spatial tables
  - Spatial Data Science book
  - Spatial Data Science in the tidyverse slides
  - Spatial Data Science in the tidyverse video

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
  http://ggplot2.tidyverse.org/
- R Graphics Cookbook (recipes for plots)
  http://www.cookbook-r.com/Graphs/index.html
- R for Data Science (online book that includes ggplot2)
  http://r4ds.had.co.nz/
- ggplot2: Elegant Graphs for Data Analysis (book by Hadley Wickham)
  http://ggplot2.org/book/
- ggplot2 cheatsheet (also in RStudio)
  http://bit.ly/ggplot2-cheatsheet
- 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

ggplot2: Chart quirks

See “templates” file

Chart components/slots

Bar chart, for example:

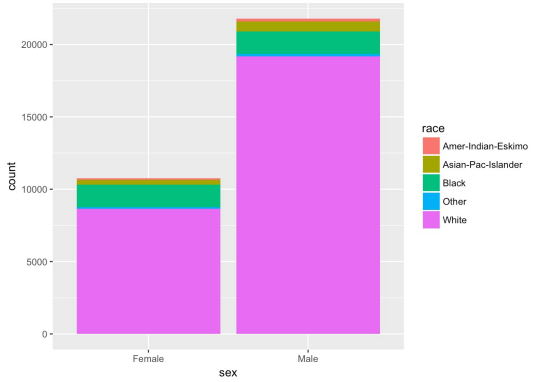
- x
  category (the names of the bars)

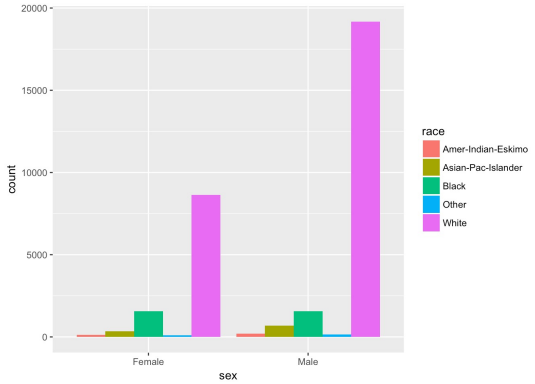
- y (optional)
  default is count, but can also specify a number (the length of the bars)

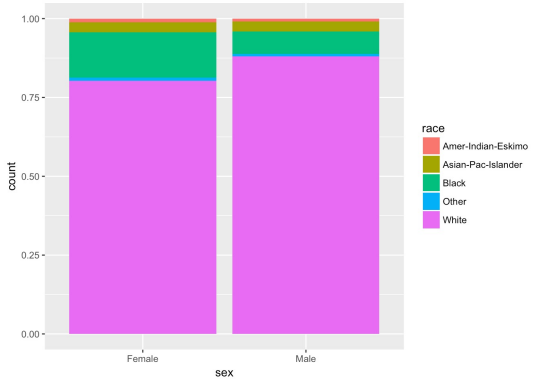
- color (optional)
  category (grouped or stacked bars)

Bar chart

- geom_bar() vs. geom_col()
- count vs. identity vs. summary
- categorical vs. continuous
- stack vs. dodge vs. fill
- bar vs. pie

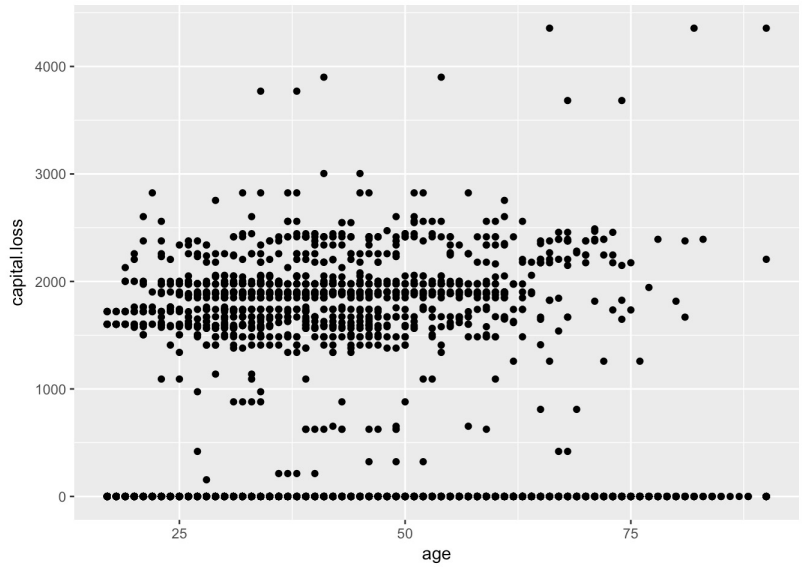


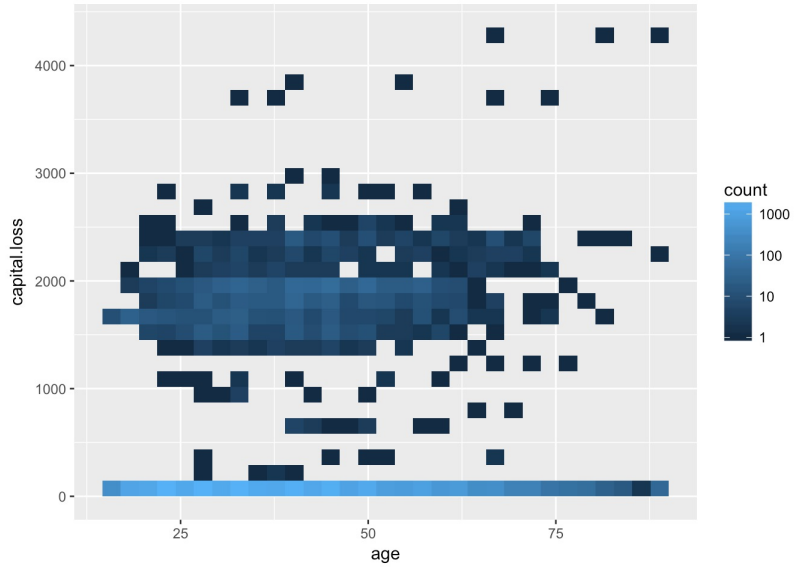




Scatter plot

- Overplotting
- point vs. bin2d





Line chart

- identity vs. summary
- line vs. smooth