

Visualization for Data Science in R

Angela Zoss

Data Matters 2018

<http://bit.ly/RVis2018>

Objectives/Outline

Day 1: Static visualizations

- Visualization and data science
- Basic ggplot2 syntax
- Charts for exploration
- Charts for communication
- Advanced topics: mapping, saving charts out

Day 2: Interactivity

- 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
- maps
- mapproj
- plotly
- flexdashboard
- shiny

Get workshop files

URL: <https://github.com/amzoss/RVis-DM2018>

With Git installed

In RStudio:

- Project → New project
- Version Control
- Git
 - Paste in GitHub URL
 - Project directory name:
RVis-DM2018
 - Subdirectory: you choose
- Create Project

Without Git installed

- Click green button to download ZIP
- Unzip files on your laptop

In RStudio:

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

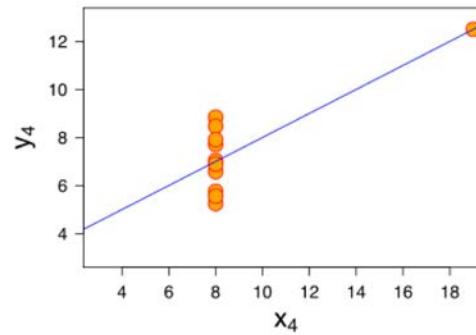
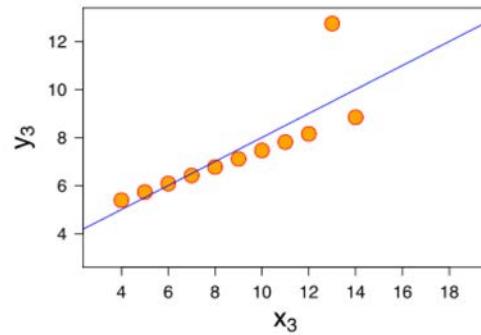
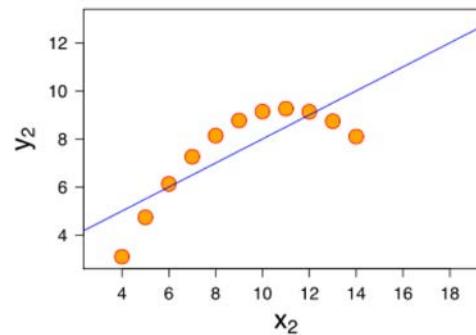
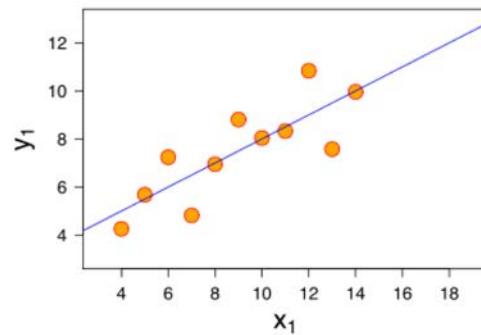
What does it mean to represent data visually?
Why do it?

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

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!

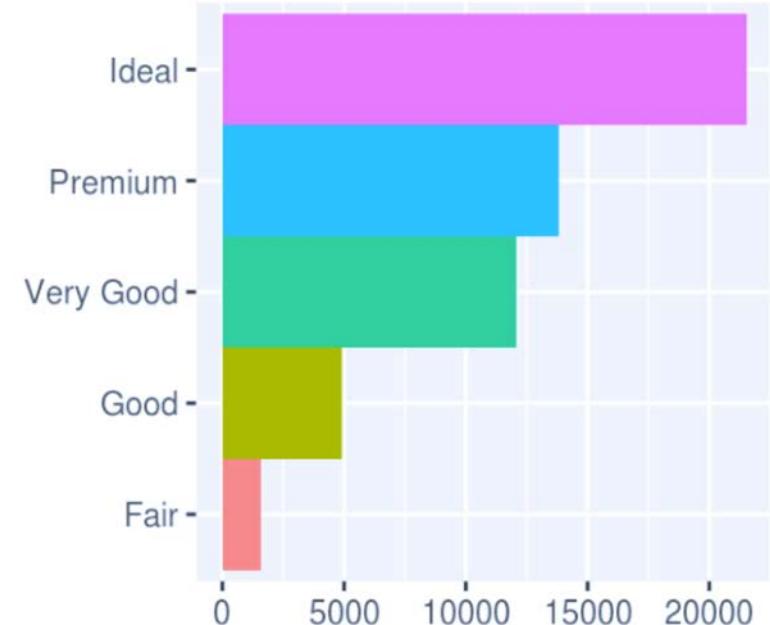
*“Your closest collaborator is **you** six months ago,
but you don’t reply to emails.”*

- *Mark Holder*

ggplot2

What is ggplot2?

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



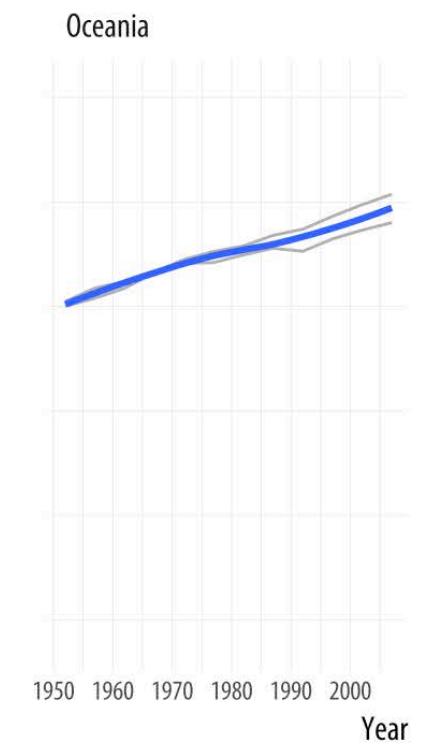
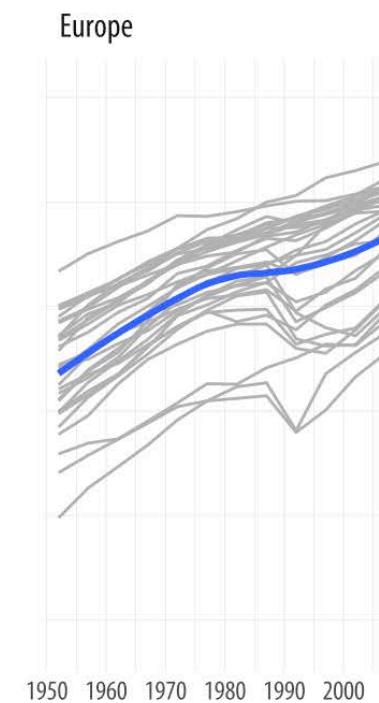
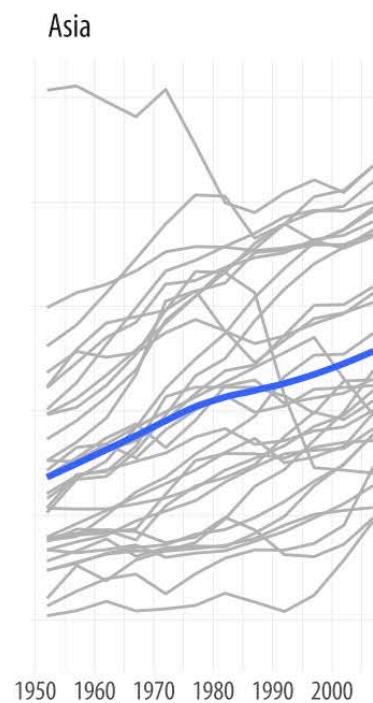
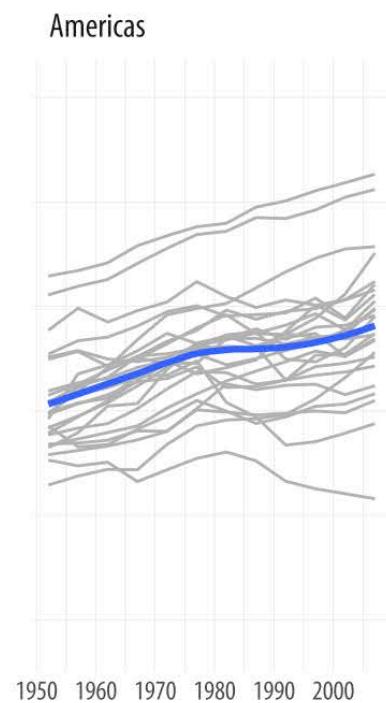
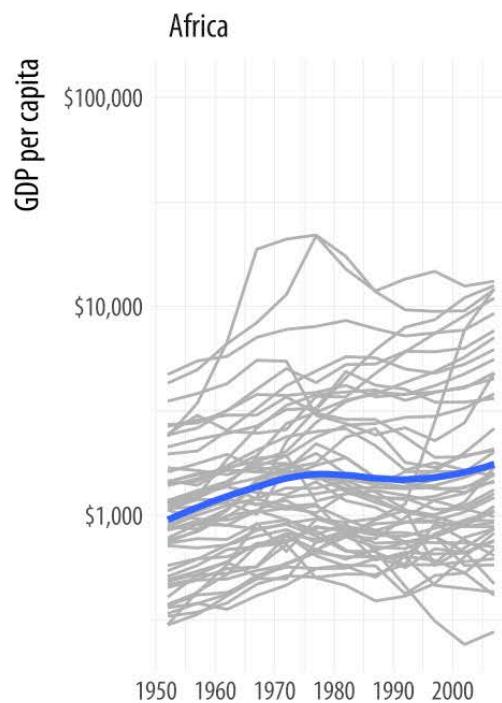
Grammar of graphics

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

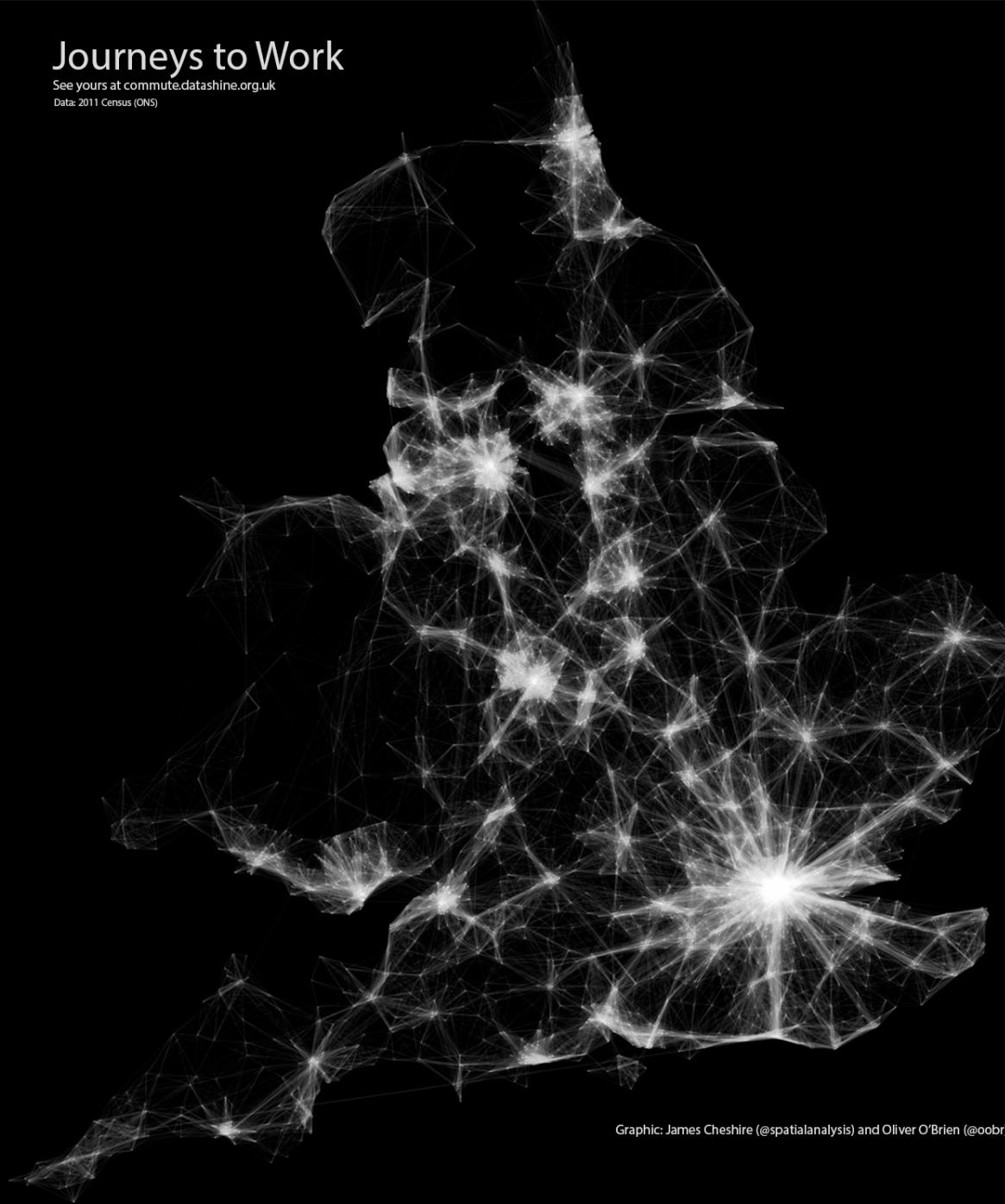


<http://socviz.co/groupfacettx.html>

Journeys to Work

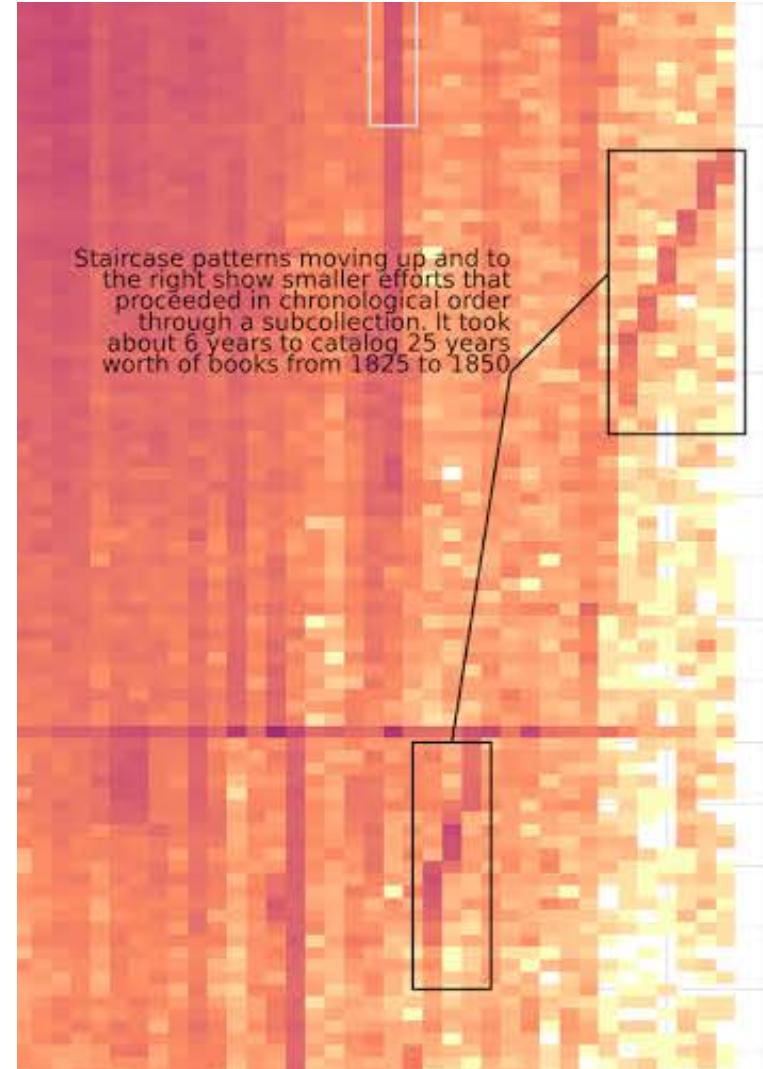
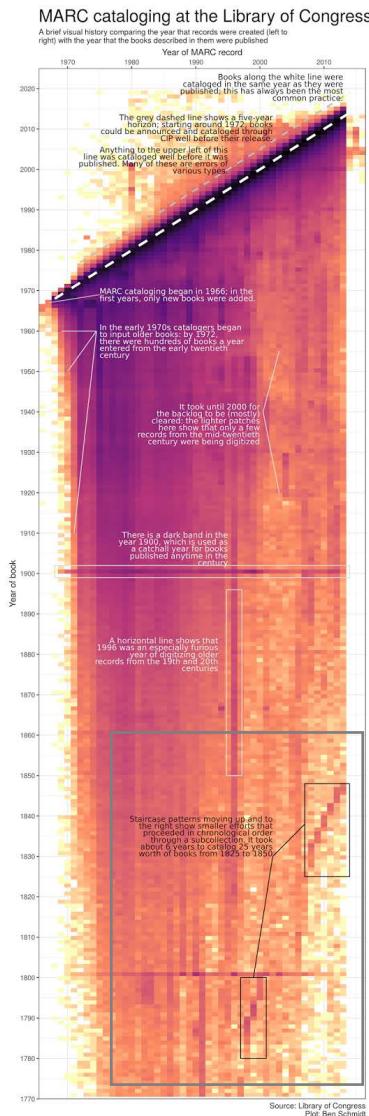
See yours at commute.datashine.org.uk

Data: 2011 Census (ONS)



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

<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

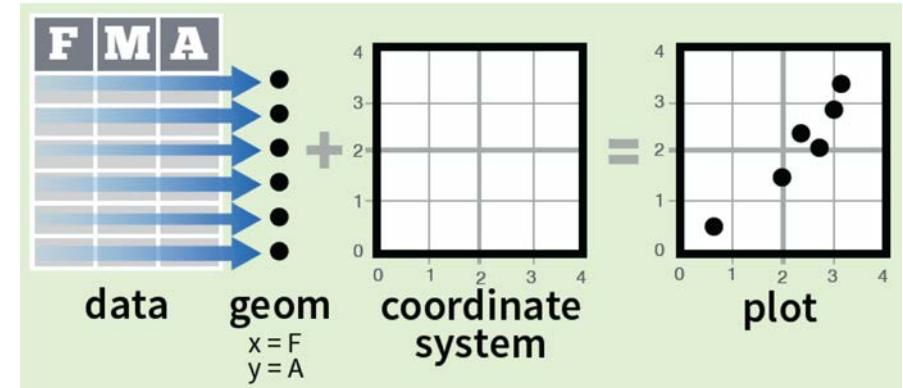
“Why I use ggplot2”, David Robinson

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

ggplot2: Elements

Basic elements in any ggplot2 visualization

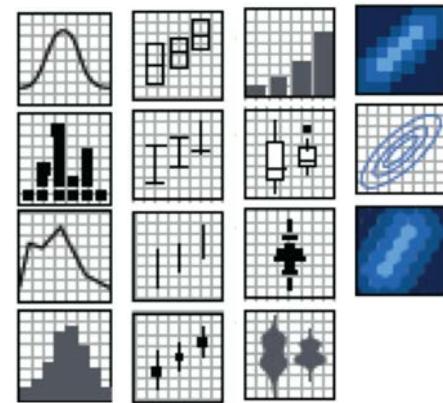
- **data**
- **aesthetics**
(variable mappings)
- **geom**
(chart type or shape)
- coordinate system
(the arrangement of the marks;
most geoms use default, cartesian)



<http://bit.ly/ggplot2-cheatsheet>

Types of geoms

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



<http://bit.ly/ggplot2-cheatsheet>

Note: some geoms also include data summary functions.
e.g., the “bar” geom will count data points in each category.

ggplot2: Basic syntax

Template for a simple plot

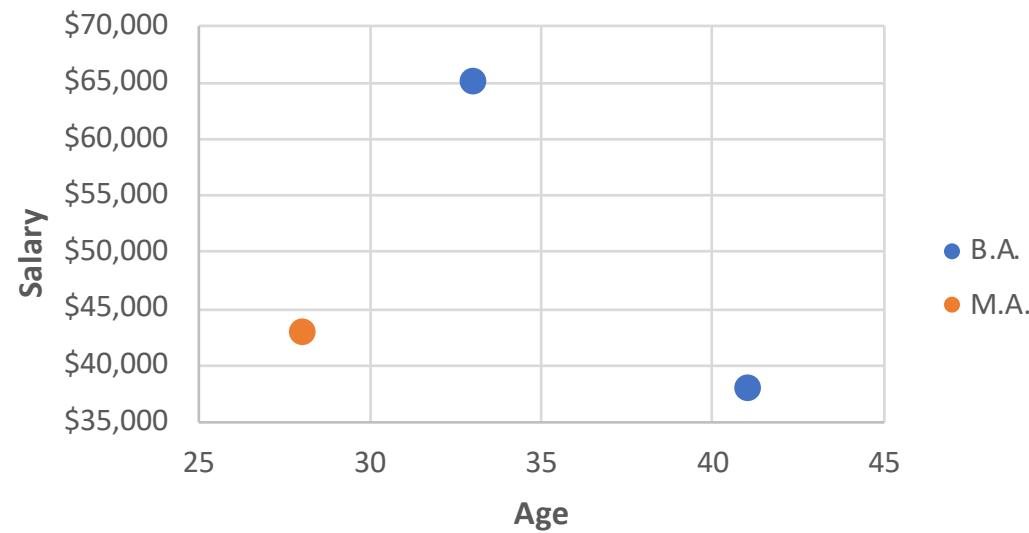
```
ggplot( data = data frame )
```

+

```
geom_... ( aes(variable mappings) ,  
          non-variable adjustments )
```

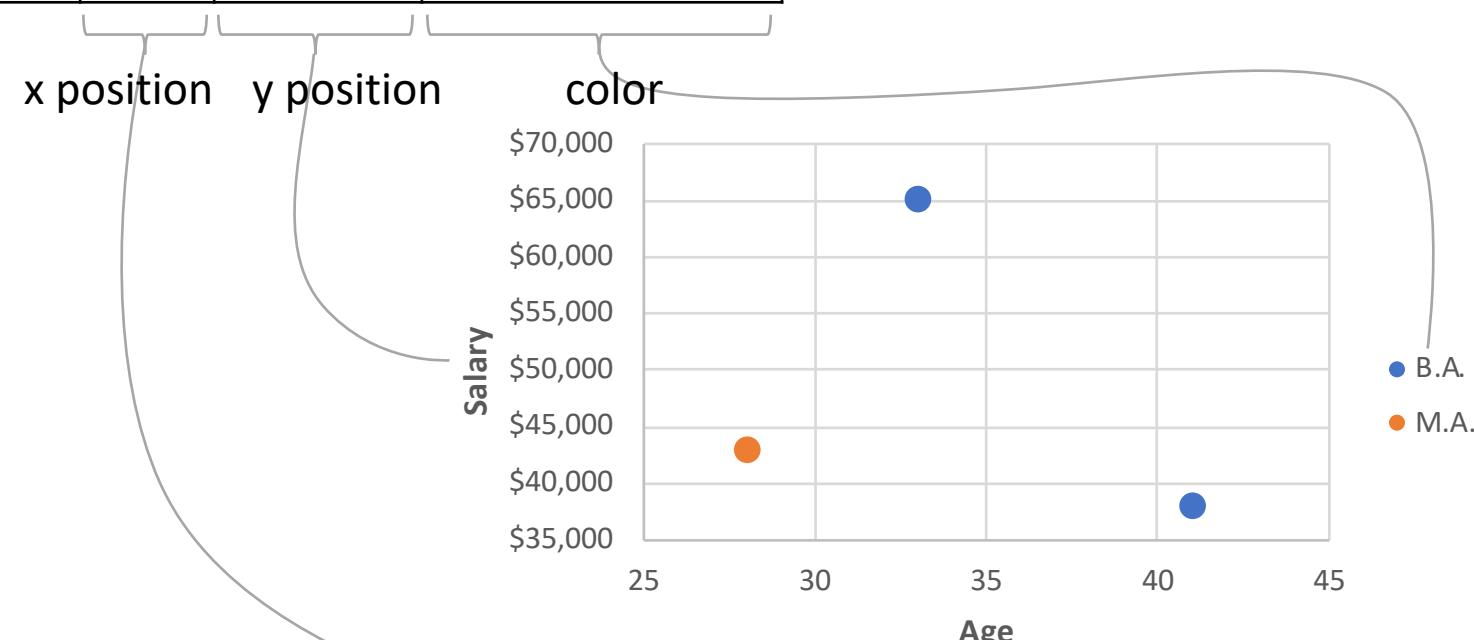
Aesthetic variable mappings

Name	Age	Salary	Highest Degree
Jane Smith	33	\$65,000	B.A.
Abby Jones	28	\$43,000	M.A.
Bridget Carden	41	\$38,000	B.A.



Aesthetic variable mappings

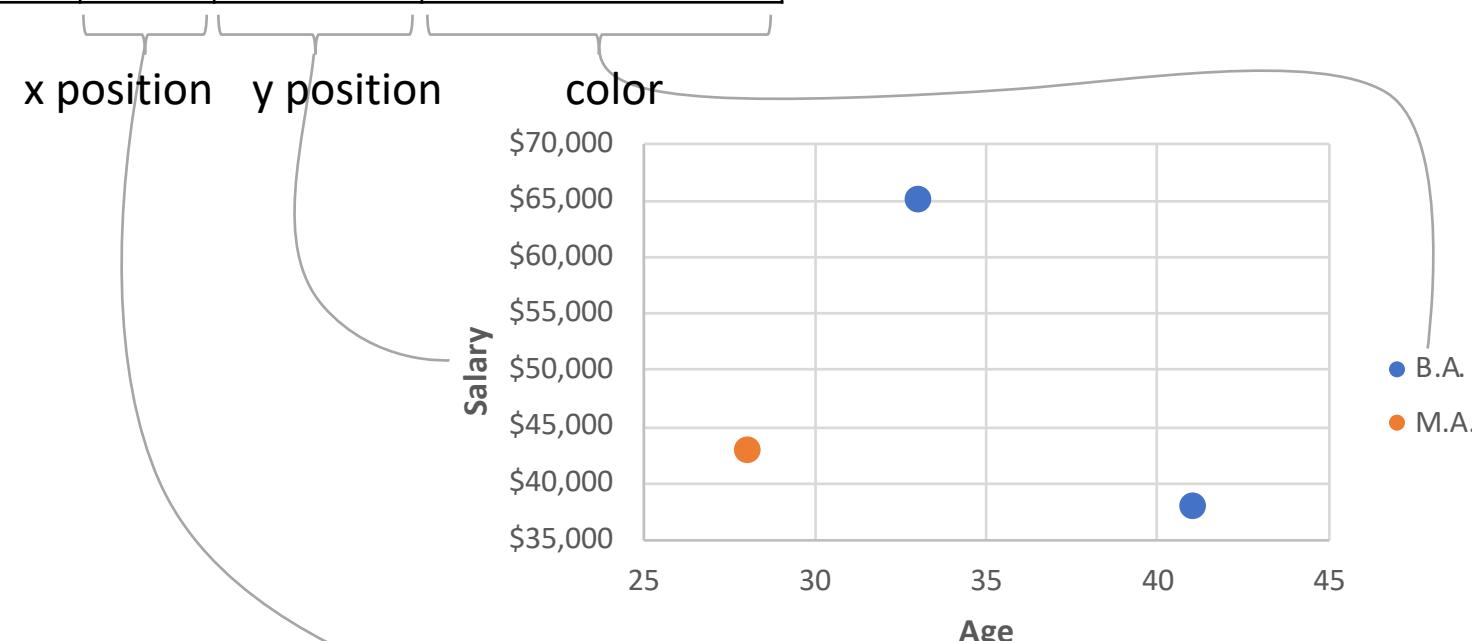
Name	Age	Salary	Highest Degree
Jane Smith	33	\$65,000	B.A.
Abby Jones	28	\$43,000	M.A.
Bridget Carden	41	\$38,000	B.A.



Aesthetic variable mappings

Name	Age	Salary	Highest Degree
Jane Smith	33	\$65,000	B.A.
Abby Jones	28	\$43,000	M.A.
Bridget Carden	41	\$38,000	B.A.

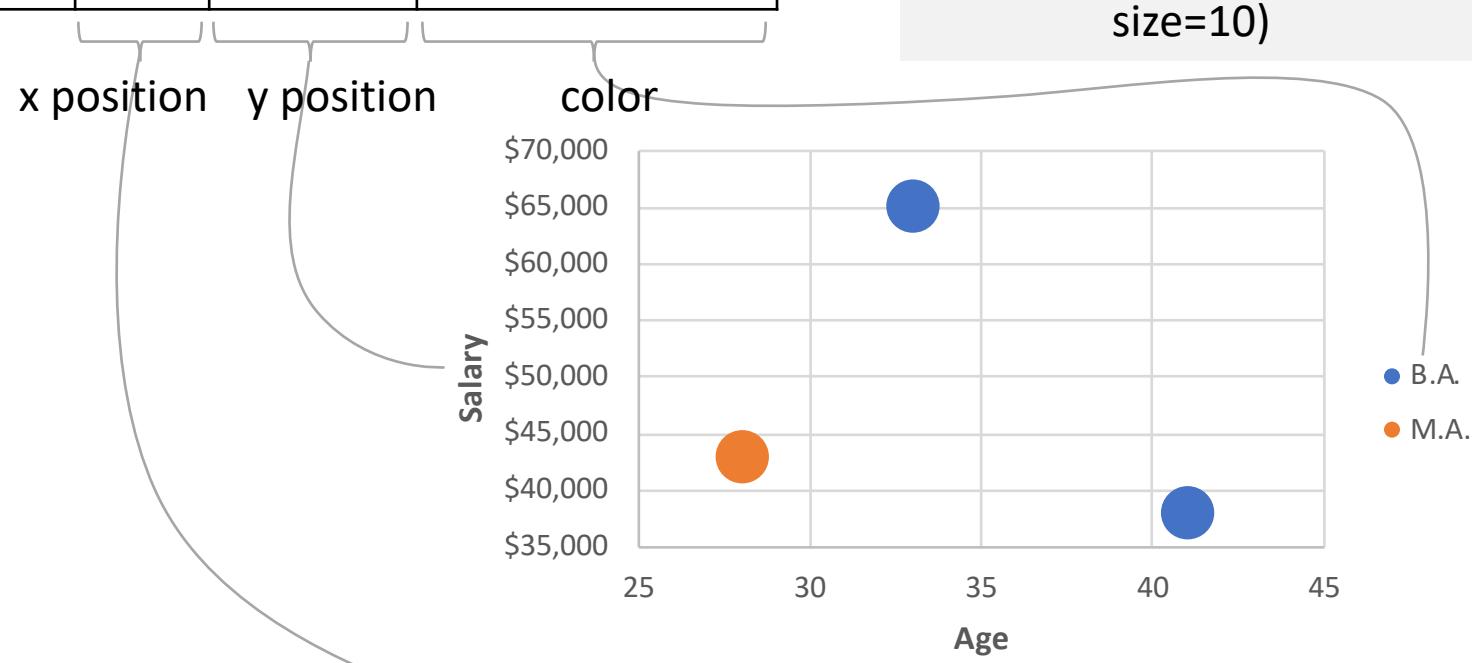
```
ggplot(data) +  
  geom_point(  
    aes(x=age,  
        y=salary,  
        color=degree))
```



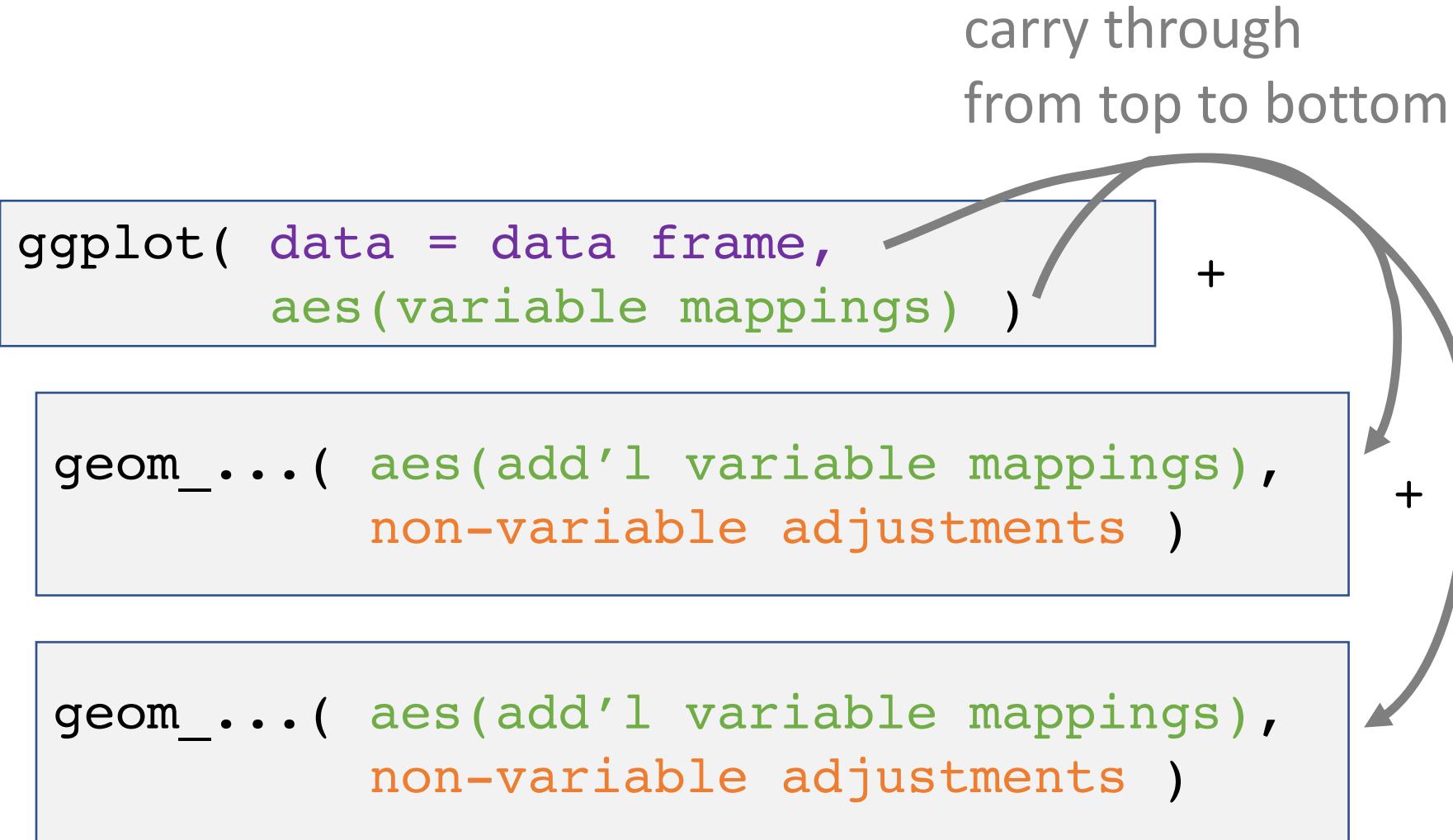
Non-variable adjustments

Name	Age	Salary	Highest Degree
Jane Smith	33	\$65,000	B.A.
Abby Jones	28	\$43,000	M.A.
Bridget Carden	41	\$38,000	B.A.

```
ggplot(data) +  
  geom_point(  
    aes(x=age,  
        y=salary,  
        color=degree),  
    size=10)
```



Template for a more complex plot



Using RStudio

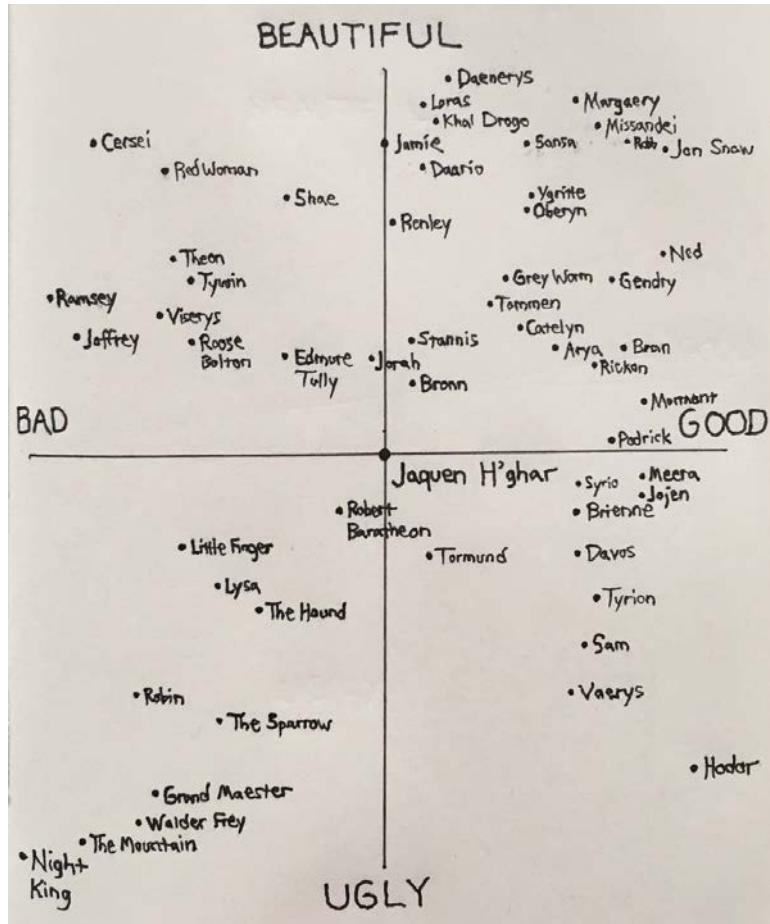
- Projects
- Rmarkdown
- Cheat sheets

<https://www.rstudio.com/resources/cheatsheets/#rmarkdown>

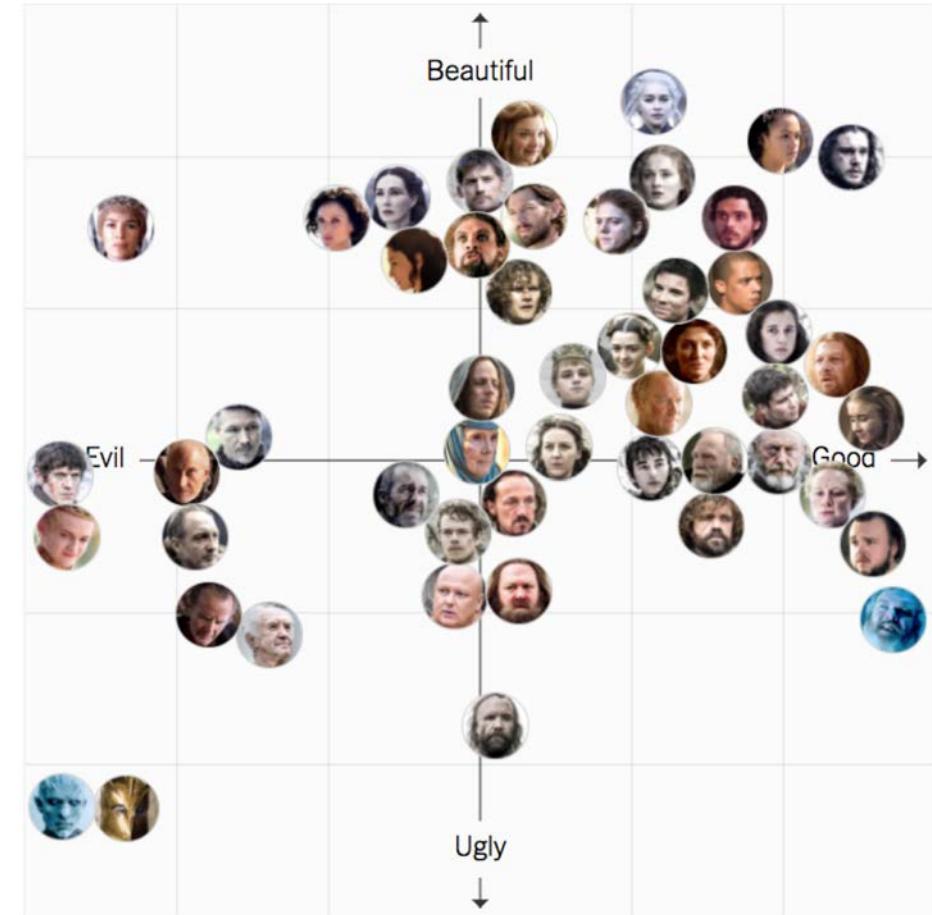
Dataset 1: Game of Thrones character ratings

[https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-
chart.html](https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-chart.html)

Game of Thrones character ratings



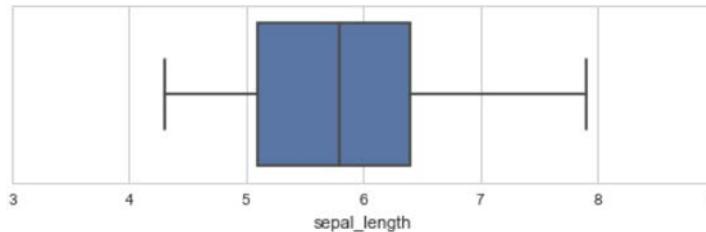
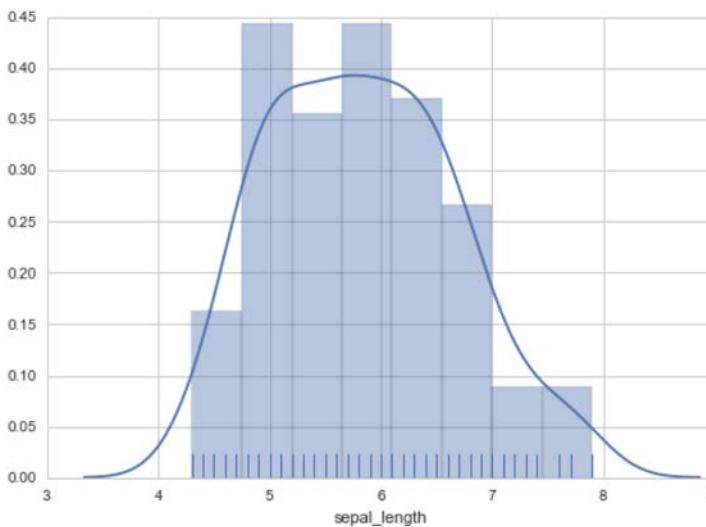
<https://www.instagram.com/p/BWnn-YogX1n/>



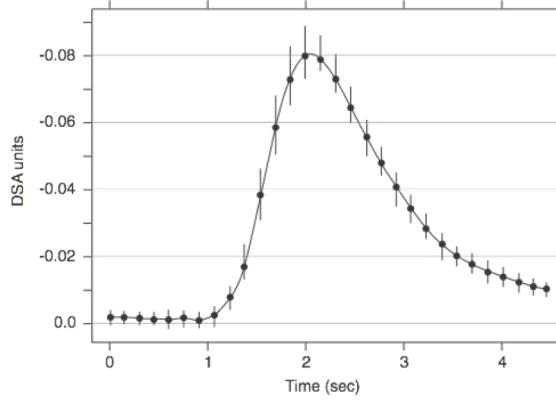
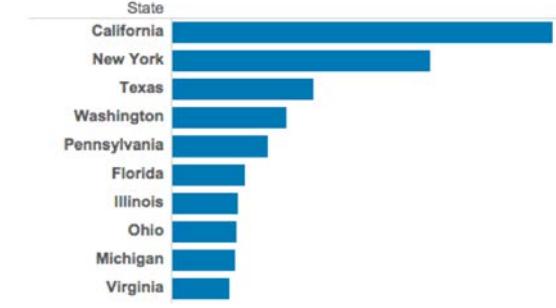
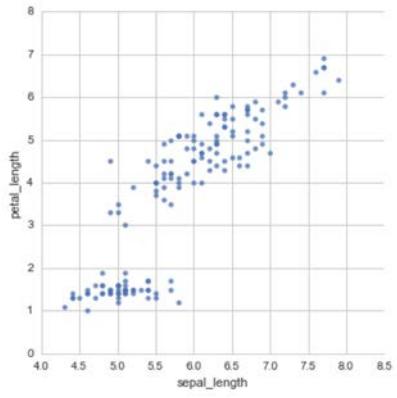
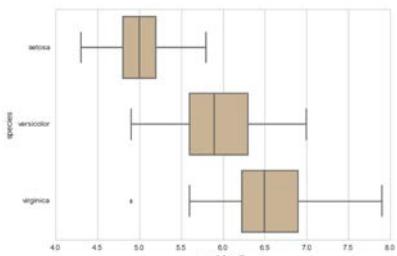
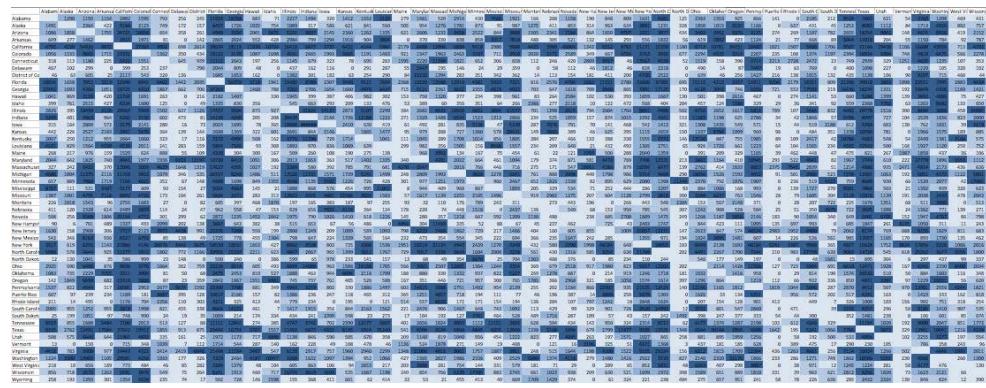
<https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-chart.html>

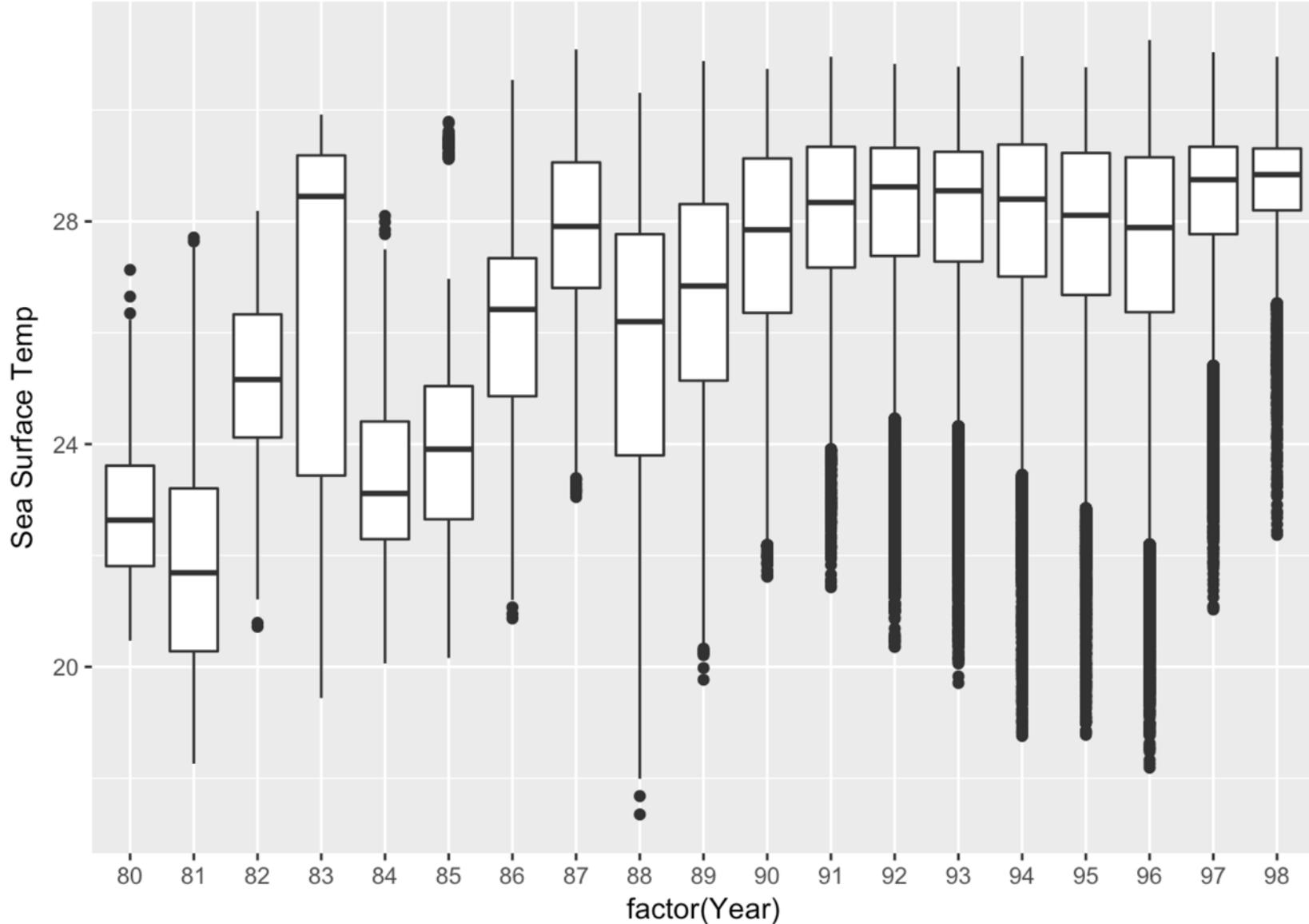
Data exploration

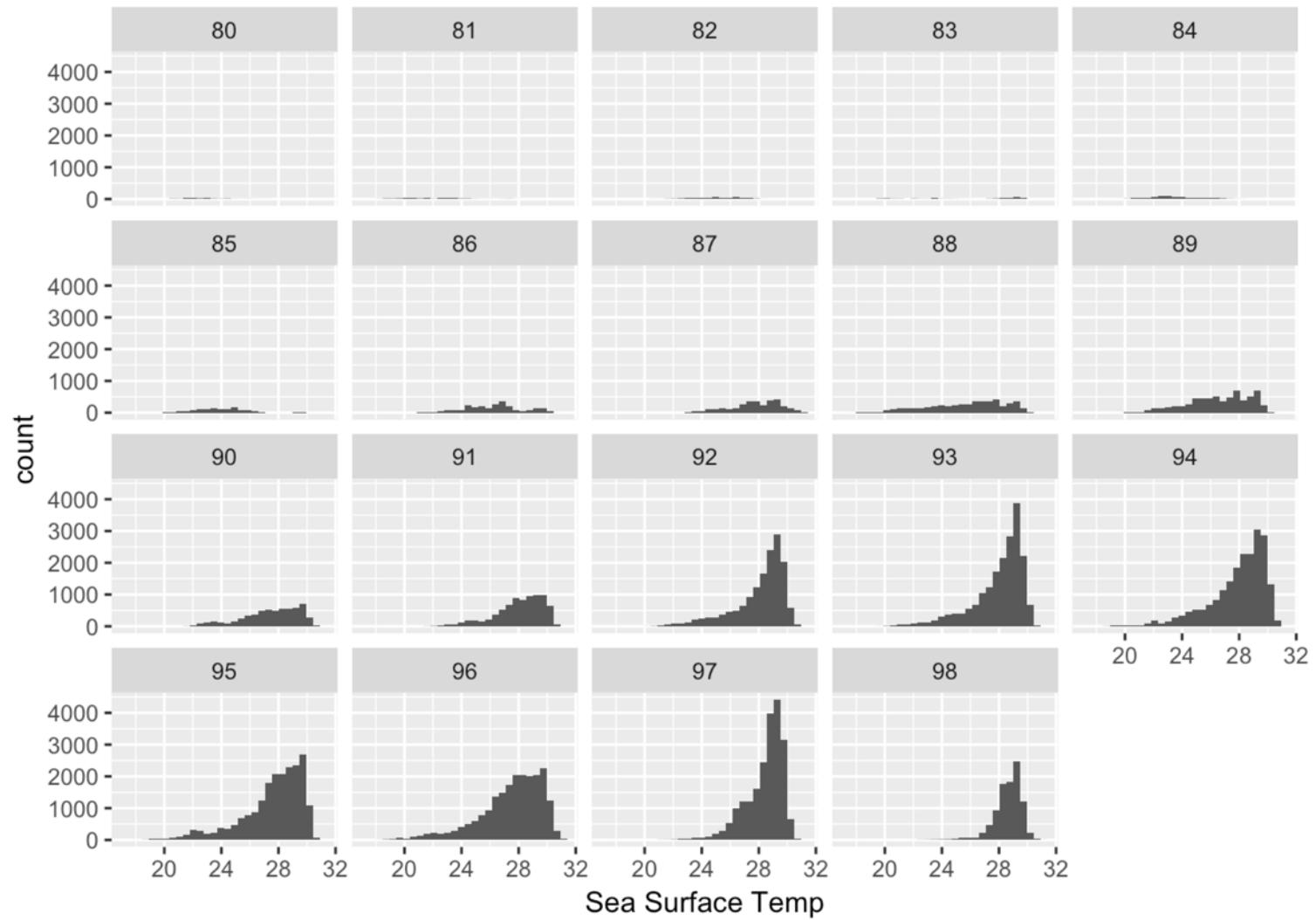
Exploring single variables



Exploring two variables

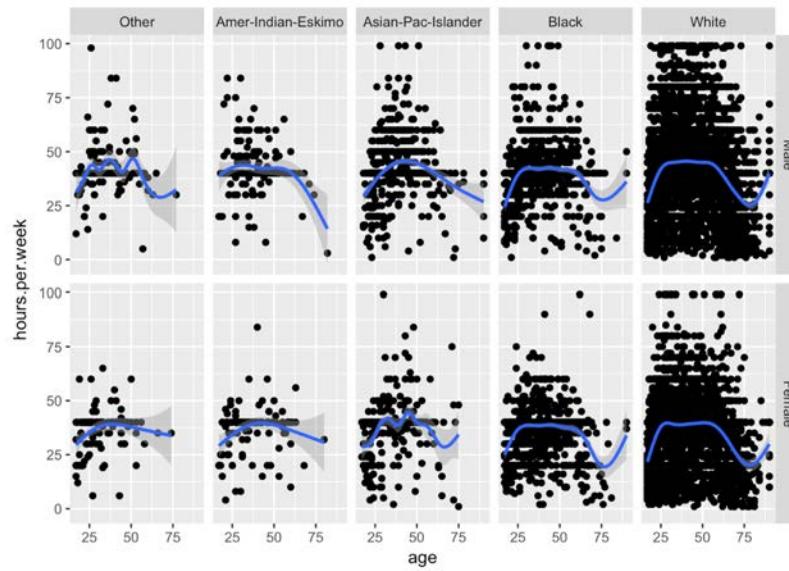
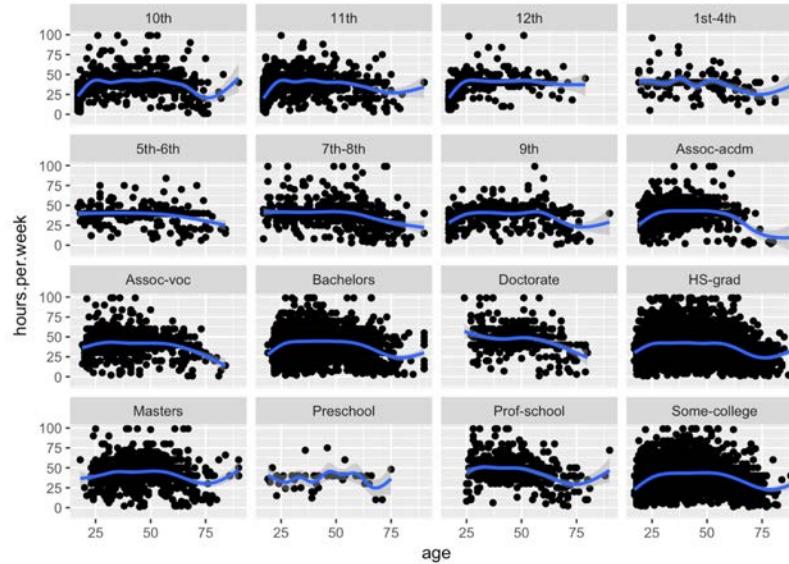




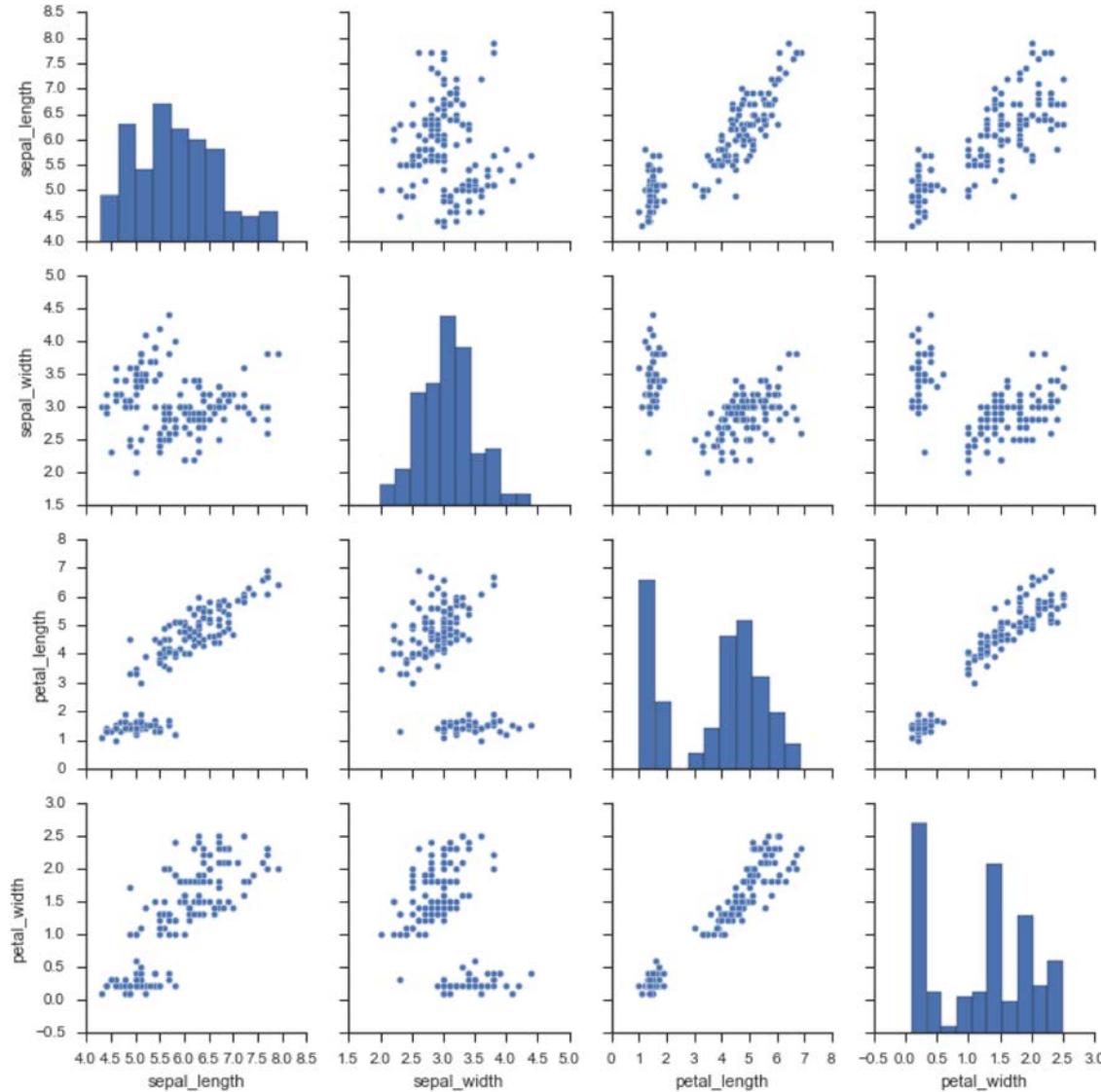


Facets

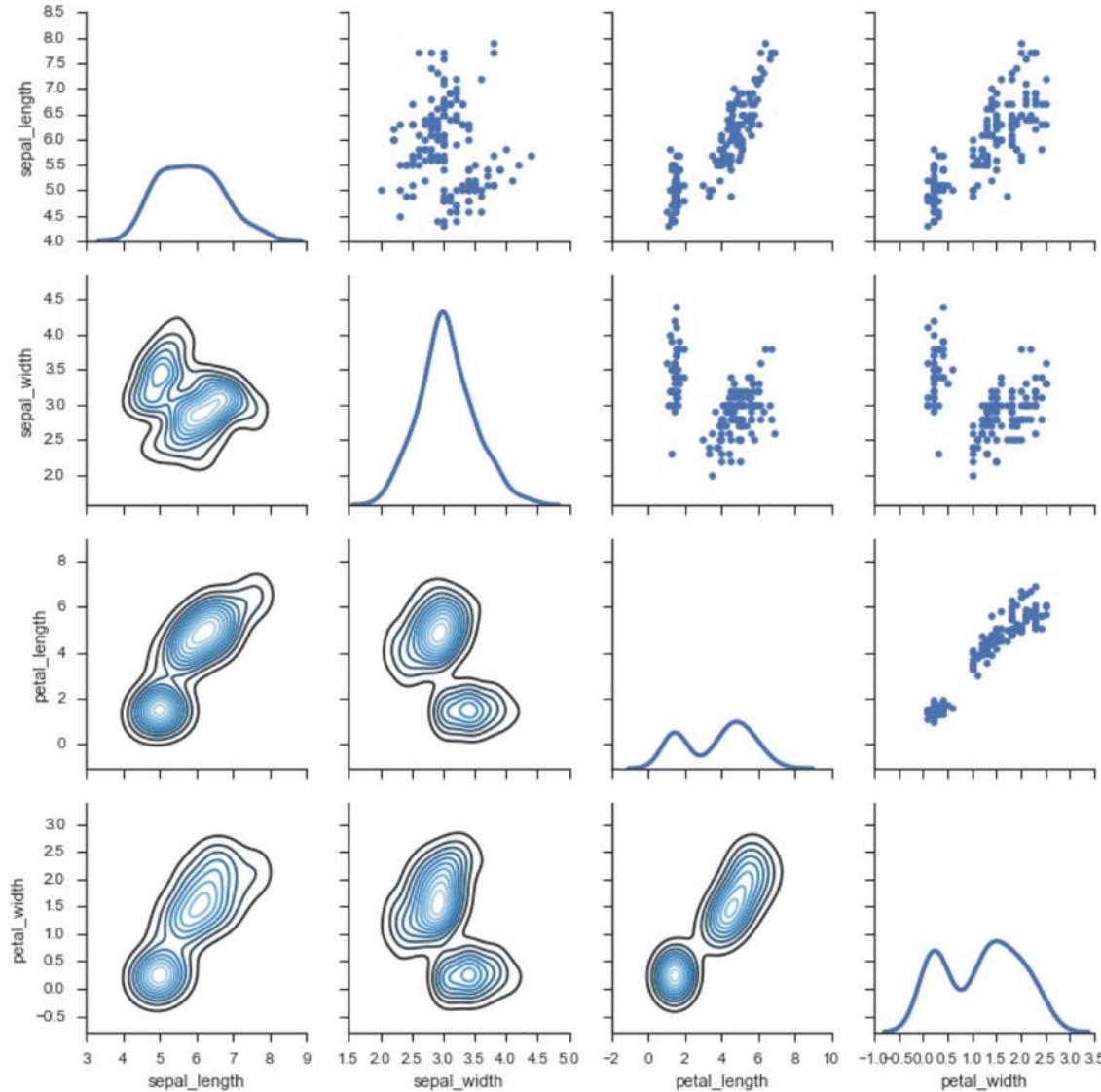
- grid vs. wrap



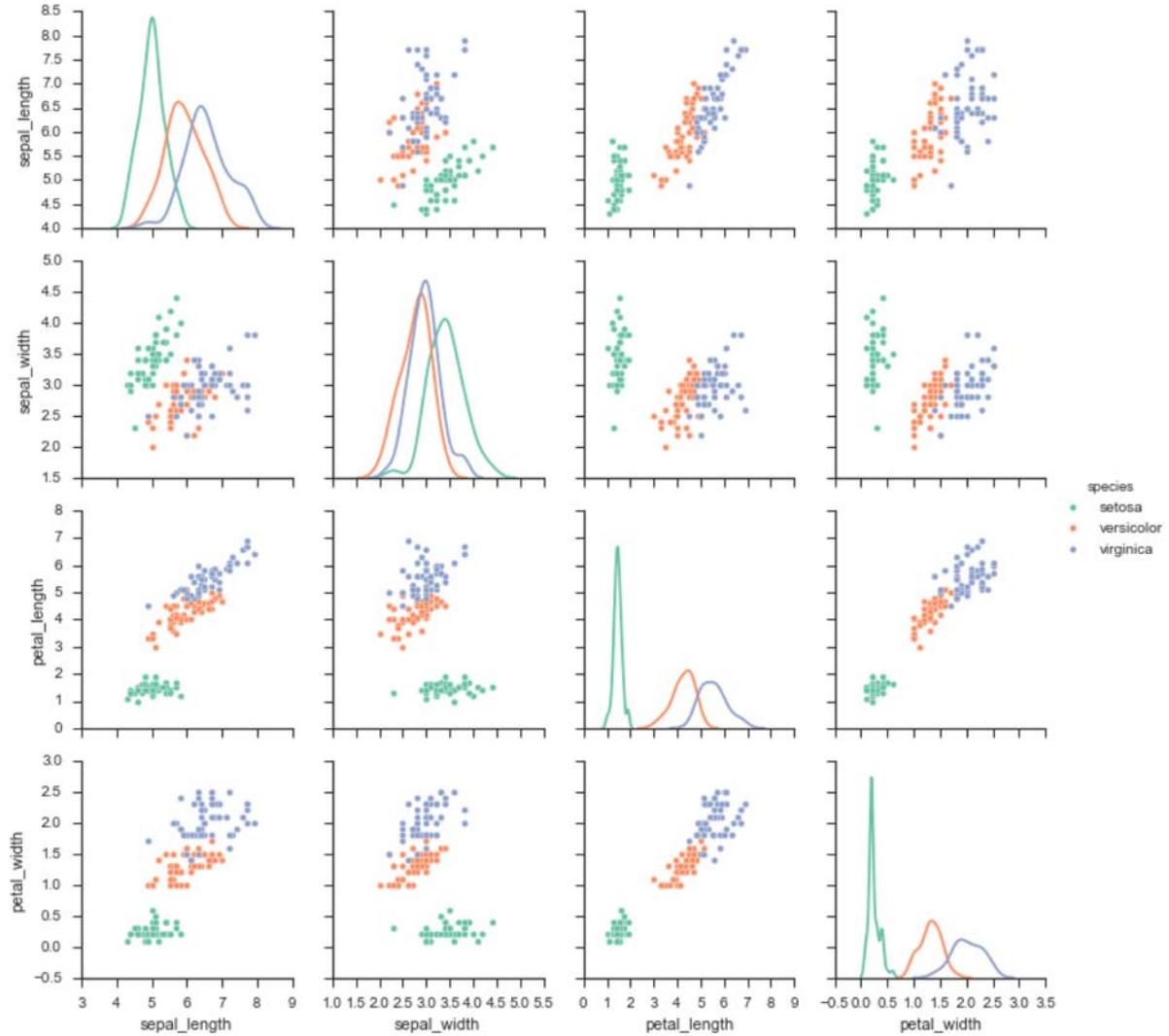
Exploring many variables, two at a time



Exploring many variables, two at a time



Exploring many variables, three at a time



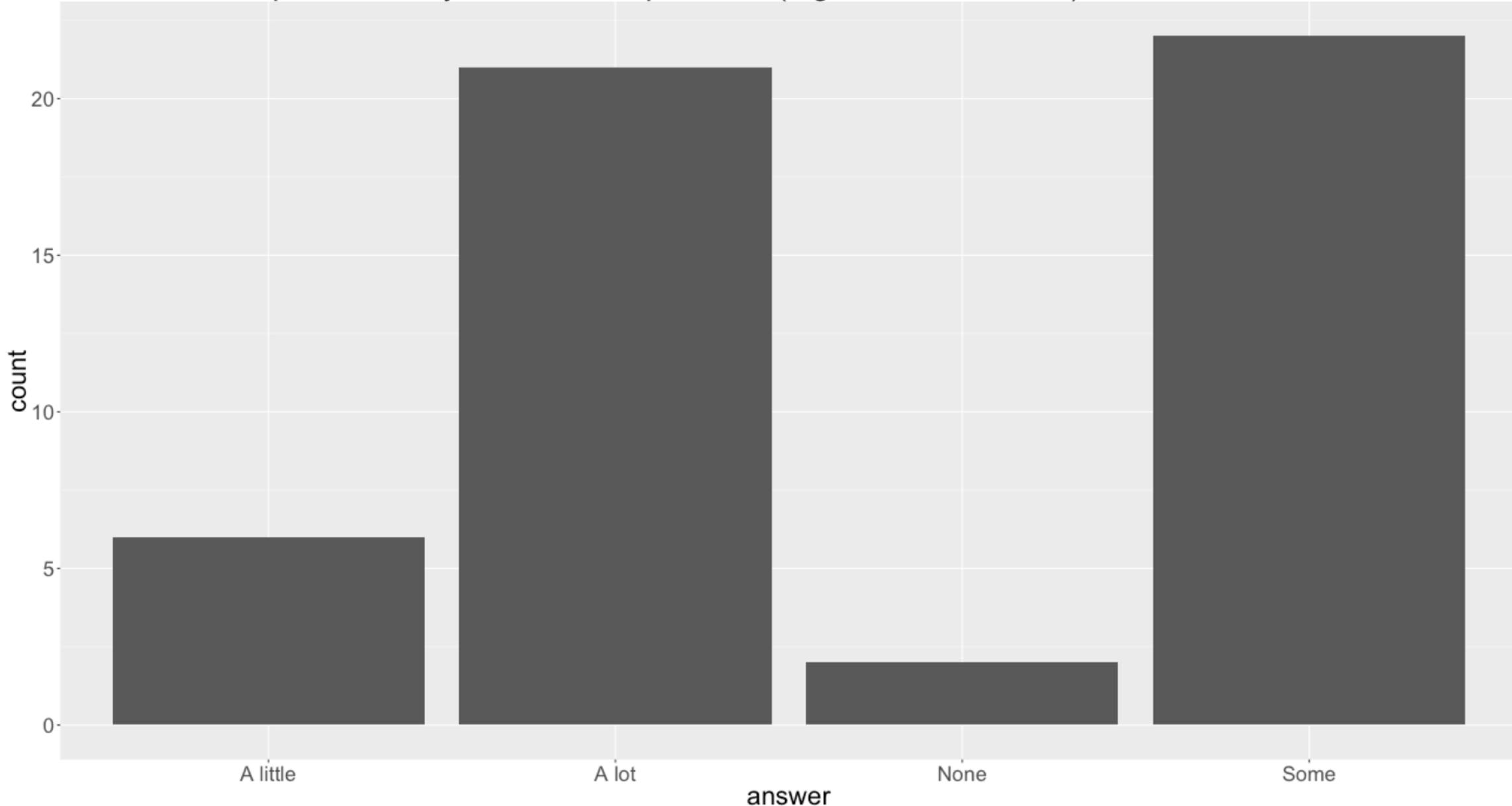
Dataset 2: Star Wars character data

<https://dplyr.tidyverse.org/reference/starwars.html>

Principles for Effective Visualizations

Principle 1: Order matters

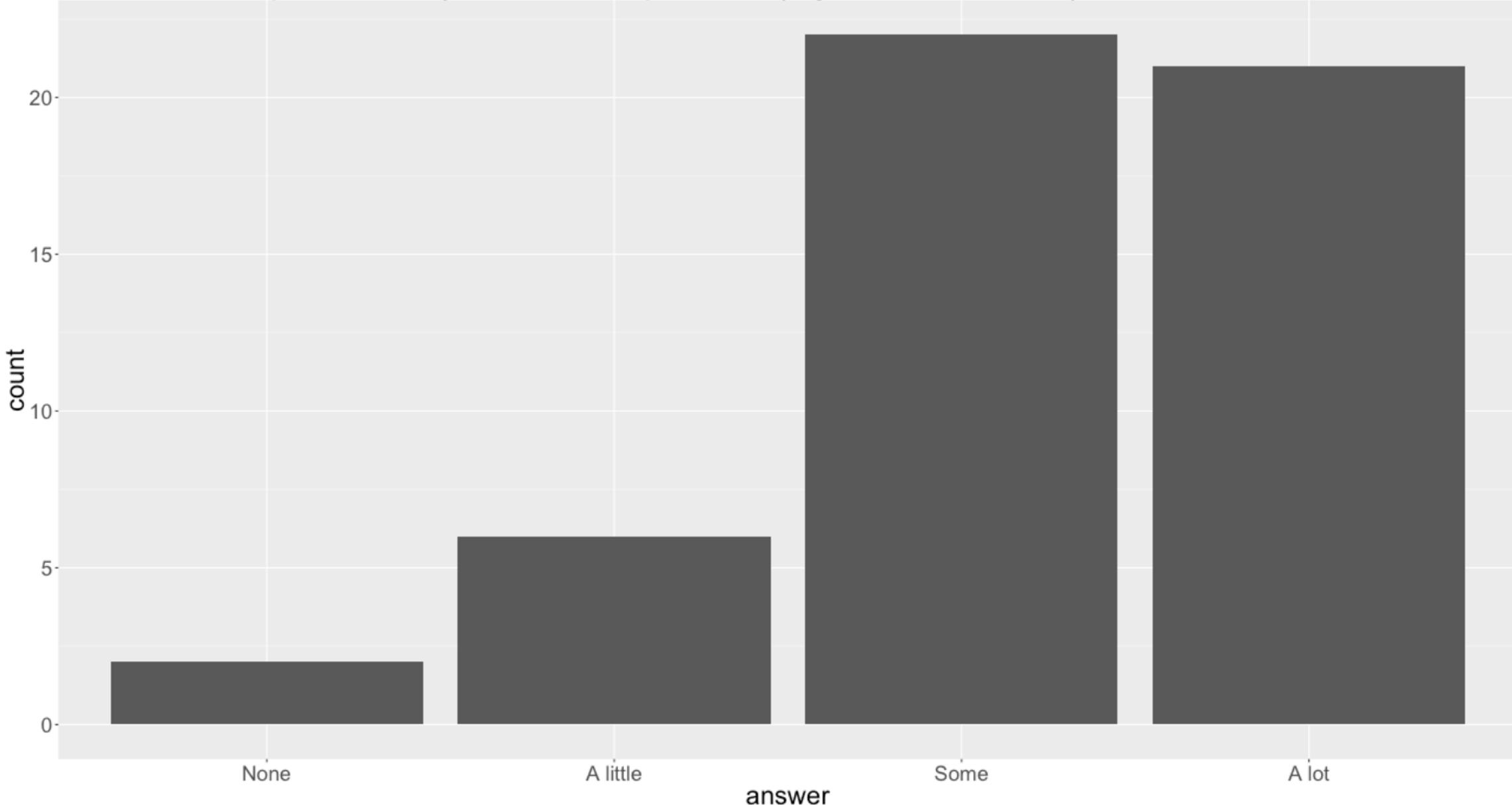
How much experience do you have as a producer (e.g., reader, follower) of network science research?



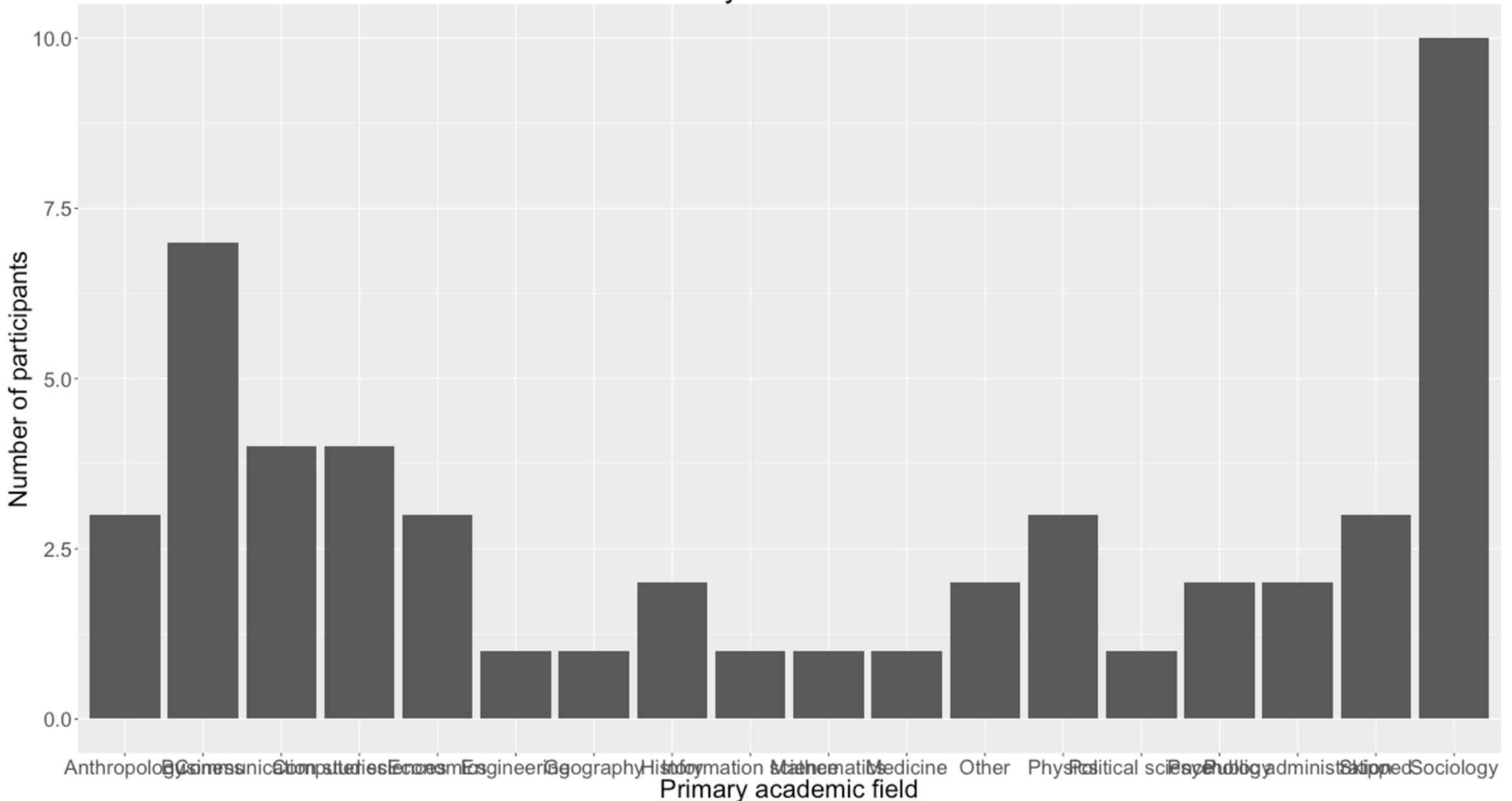
Order by meaning

```
data$answer <-  
  factor(data$answer,  
         levels=c("None", "A little", "Some", "A lot"),  
         ordered = TRUE)
```

How much experience do you have as a producer (e.g., reader, follower) of network science research?



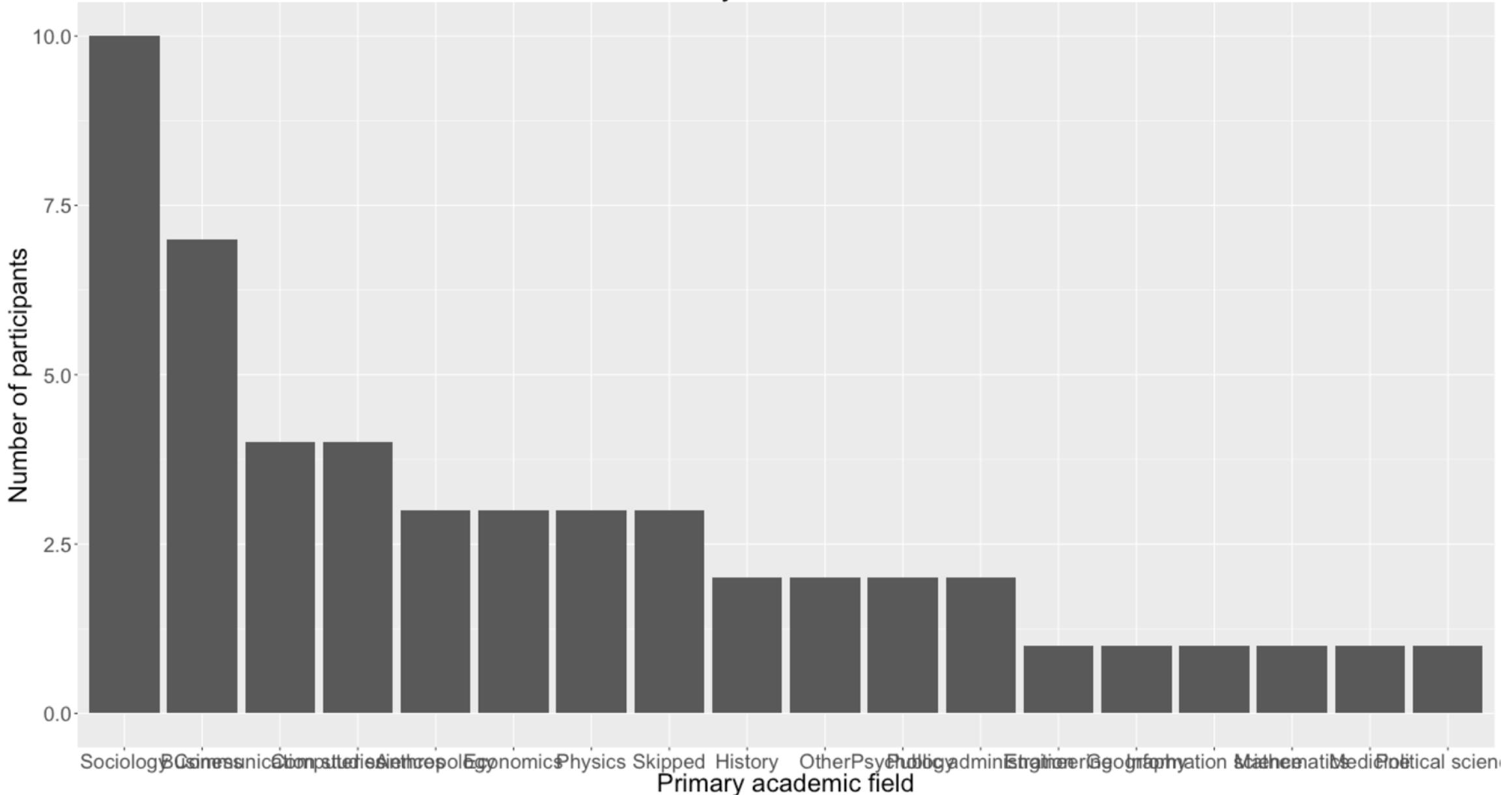
Primary academic field



Order by value

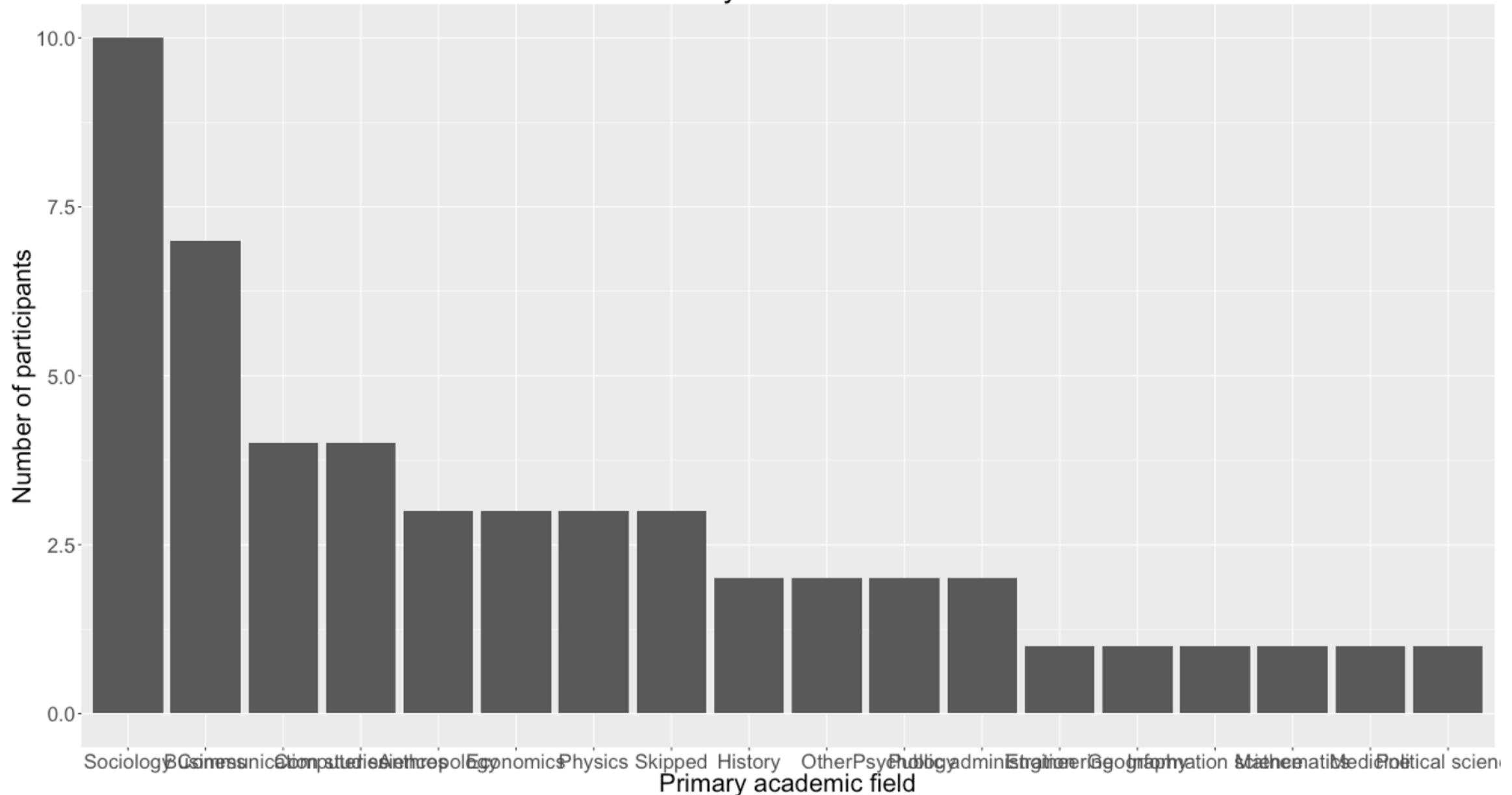
```
data$academic_field <-  
  factor(data$academic_field,  
         levels=names(  
           sort(  
             table(  
               data$academic_field),decreasing=TRUE))))
```

Primary academic field



Principle 2:
Put long categories on y-axis

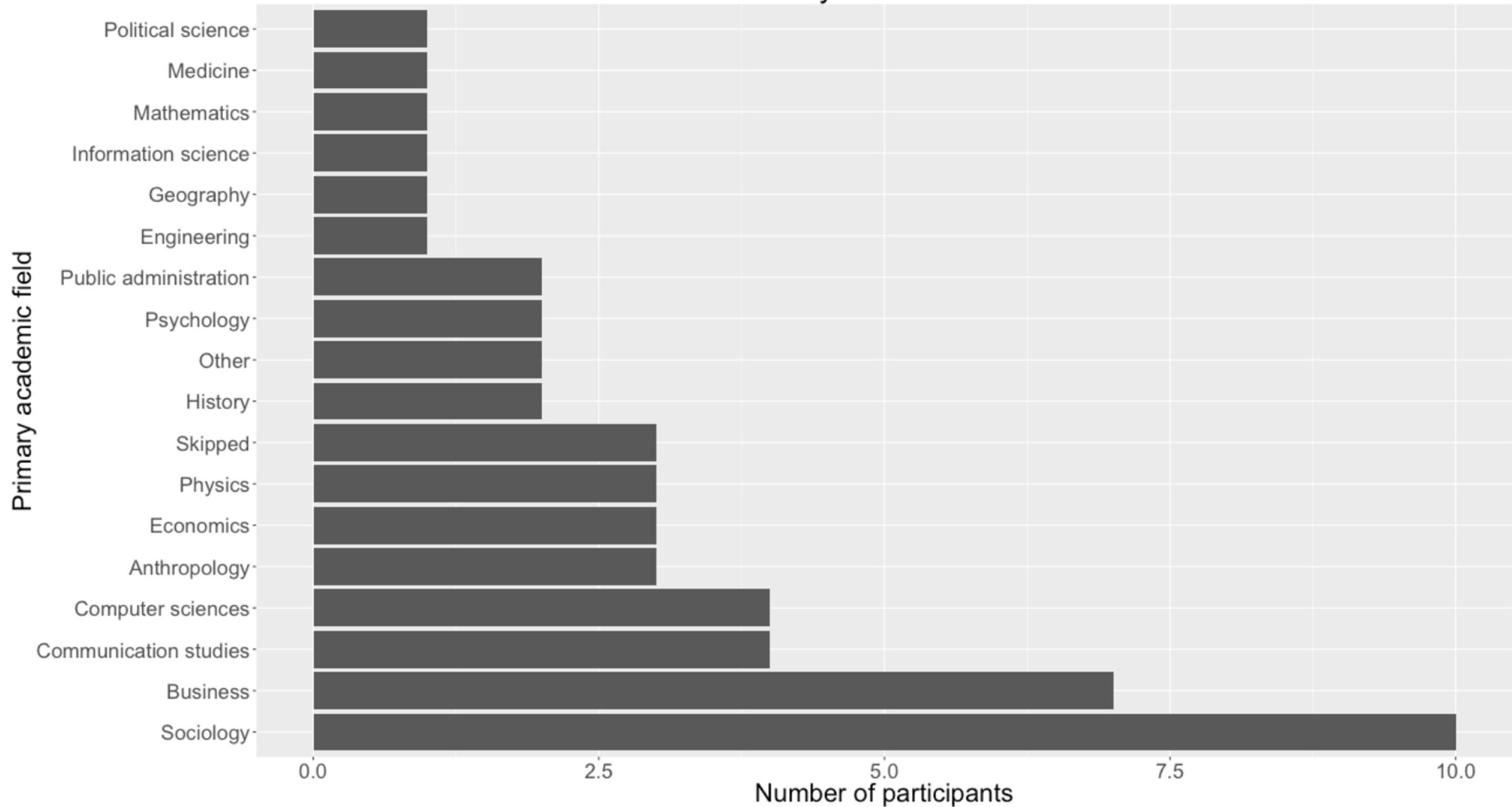
Primary academic field



Flip the axes

```
coord_flip()
```

Primary academic field

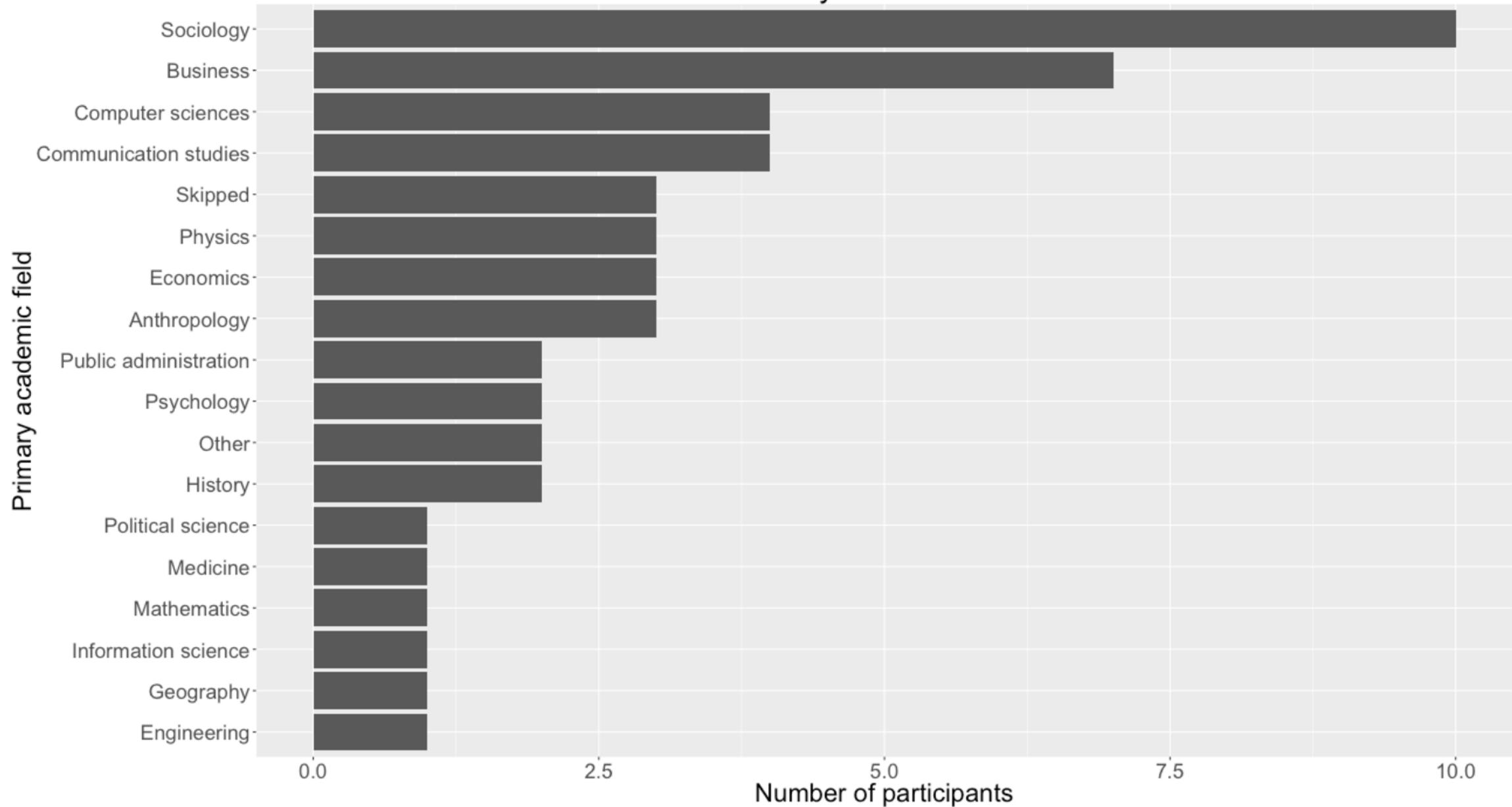


Oops!

```
data$academic_field <-  
  factor(data$academic_field,  
         levels=names(  
             sort(  
                 table(data$academic_field),  
                 decreasing=TRUE))))
```

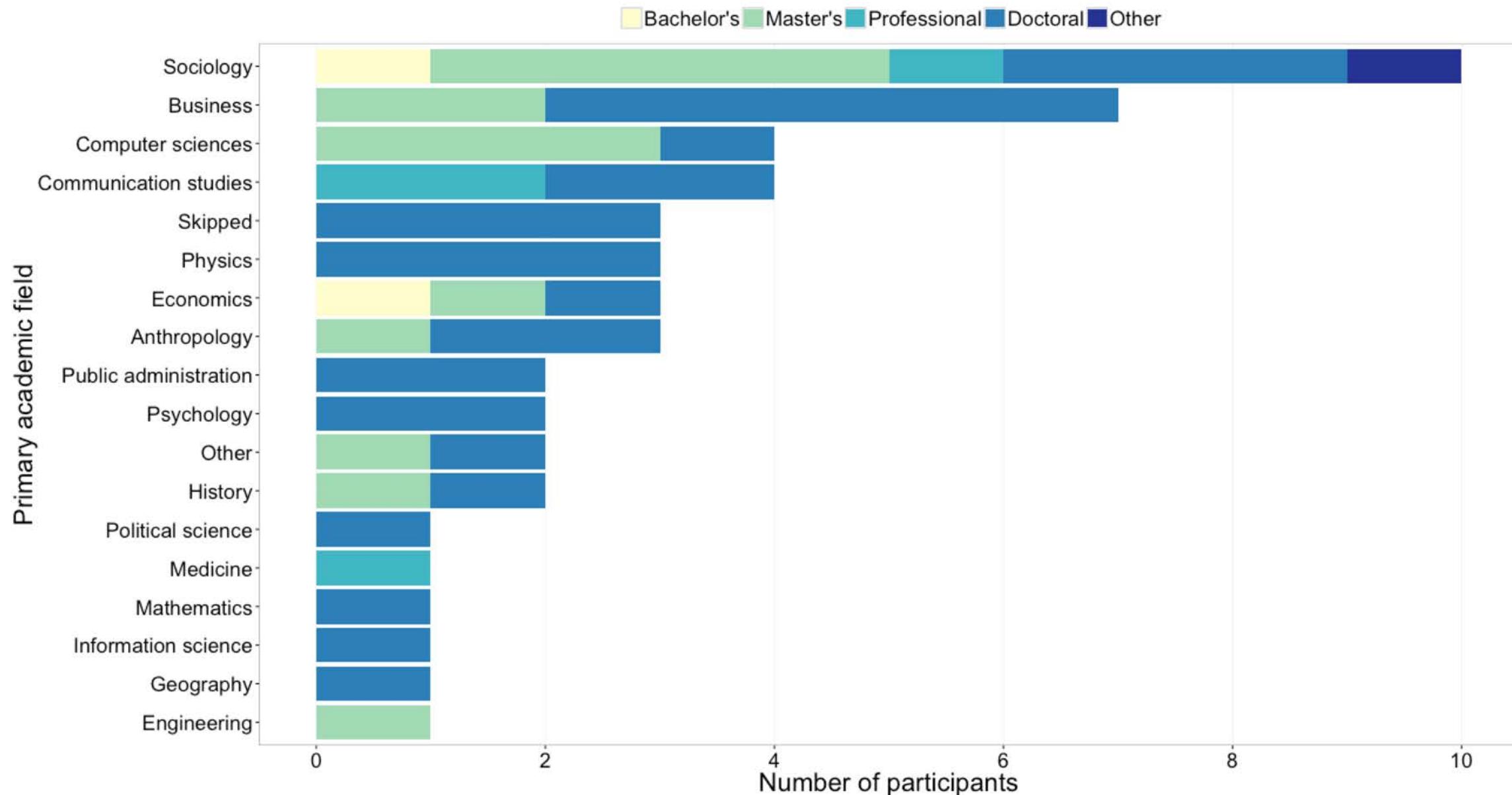
```
data$academic_field <-  
  factor(data$academic_field,  
         levels=names(  
             sort(  
                 table(data$academic_field)))))
```

Primary academic field

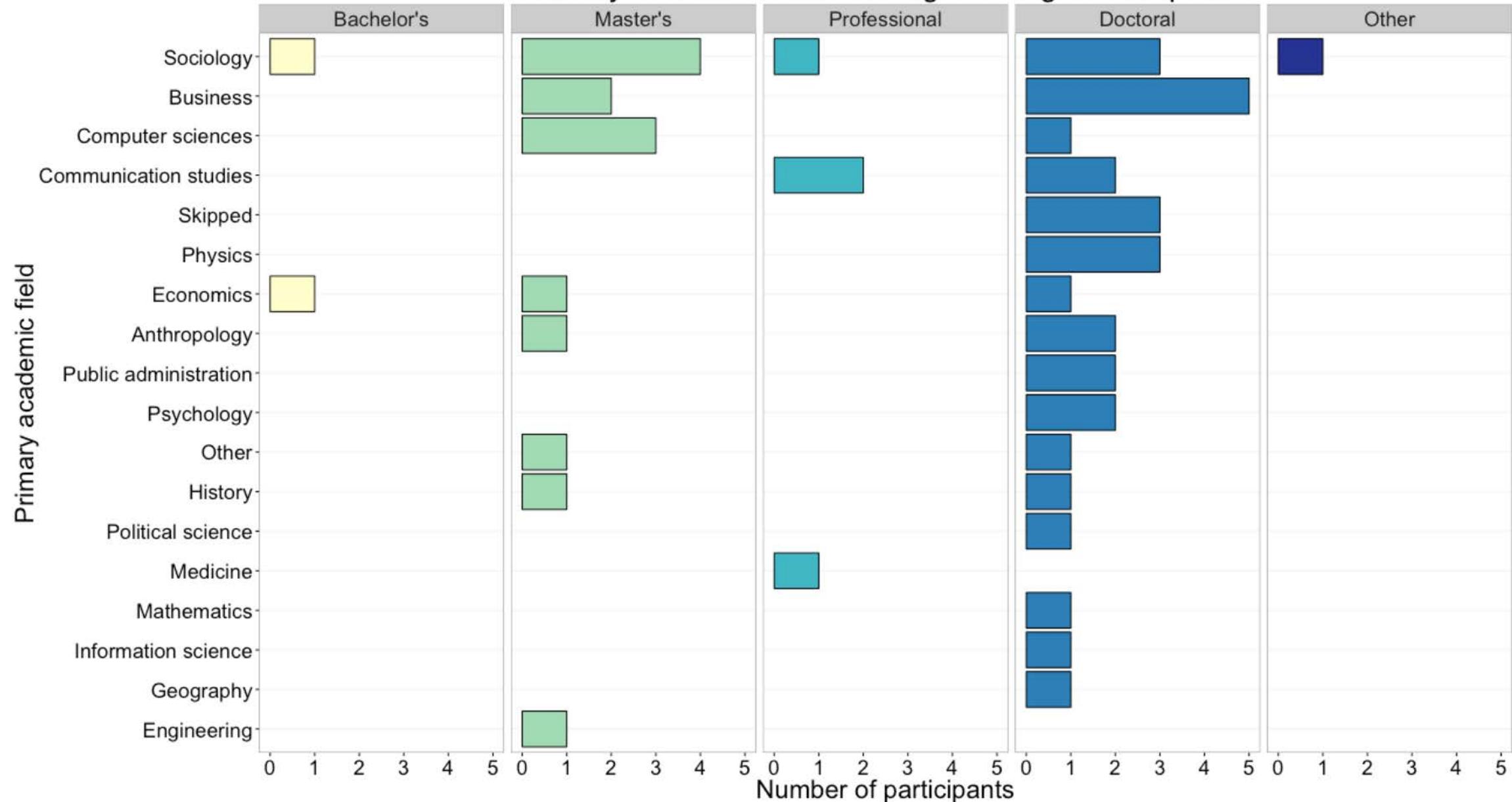


Principle 3: Pick a purpose

Primary academic field and highest degree completed



Primary academic field and highest degree completed



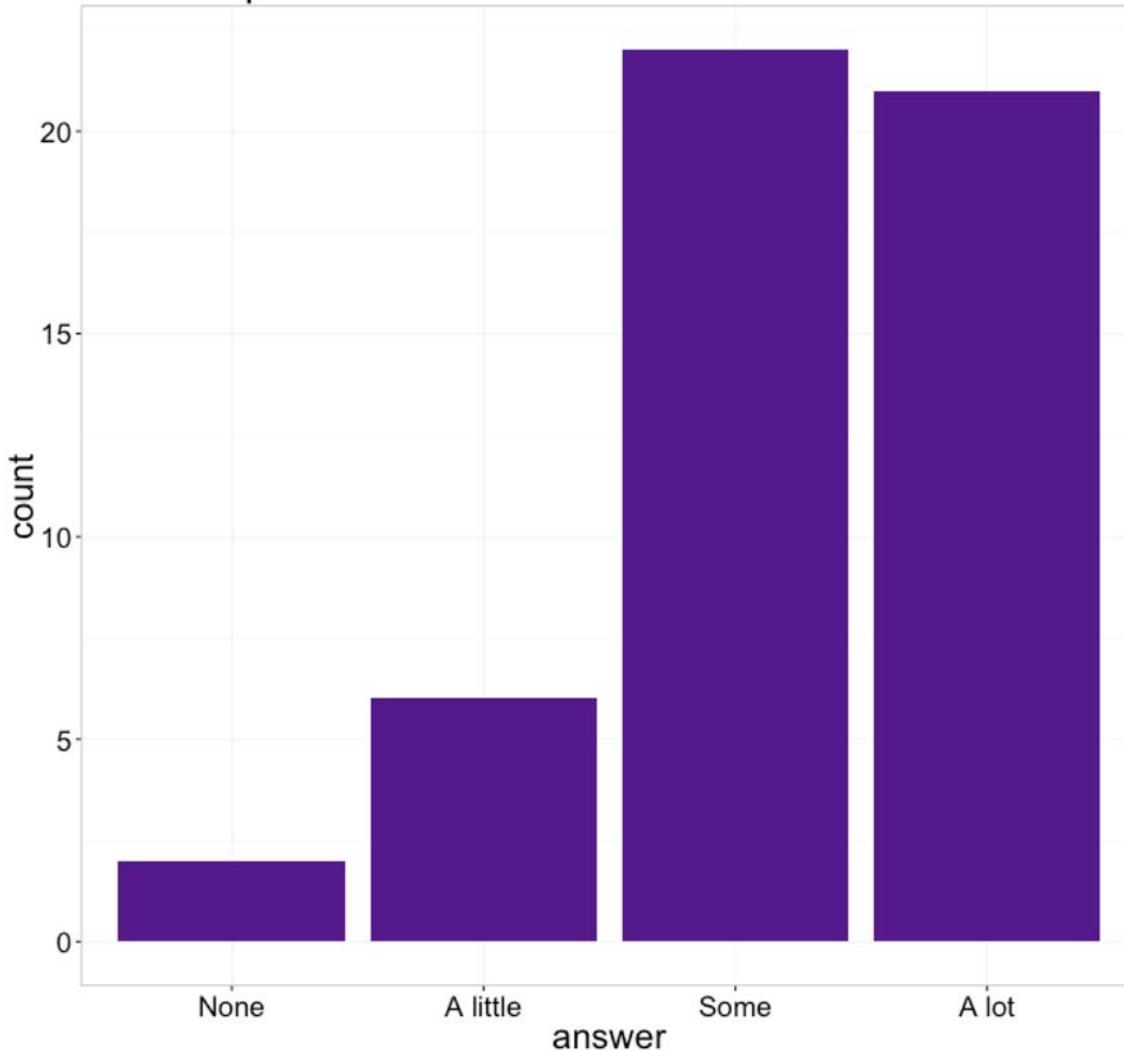
Different placement helps with different comparisons

```
fill=highest_degree
```

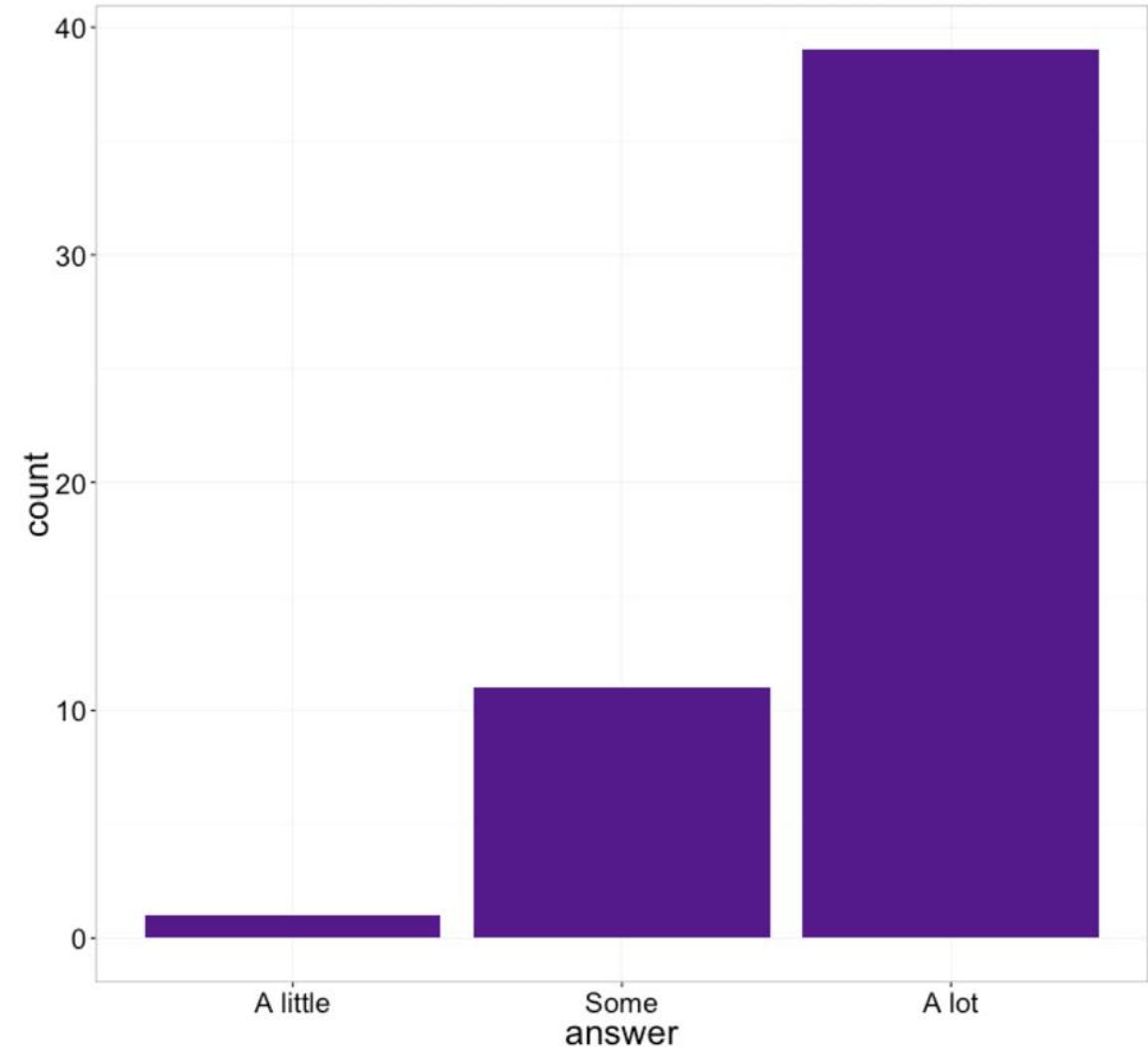
```
facet_grid(.~highest_degree)
```

Principle 4:
Keep scales consistent

How much experience do you have as a producer of network science research?



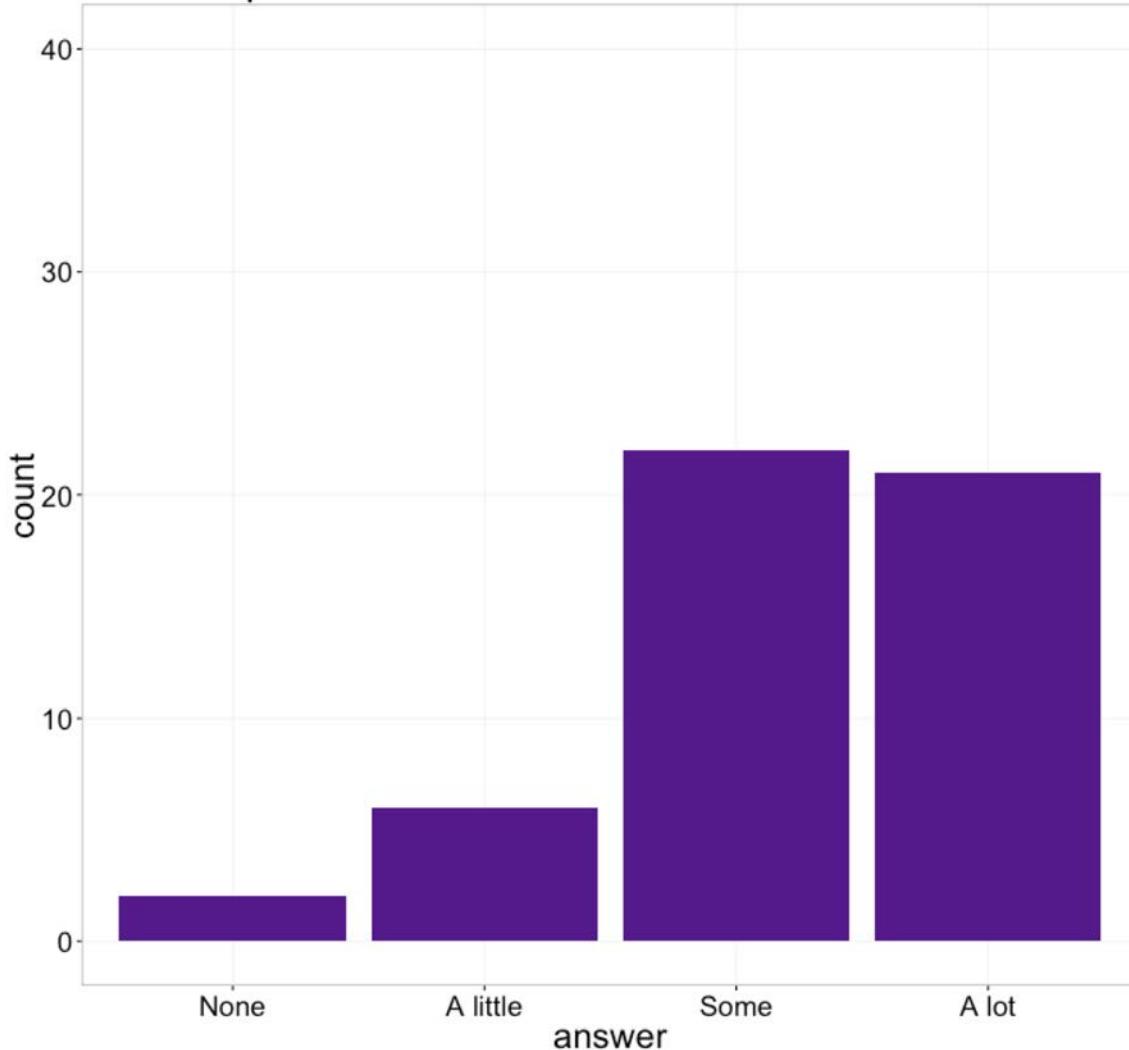
How much experience do you have as a consumer of network science research?



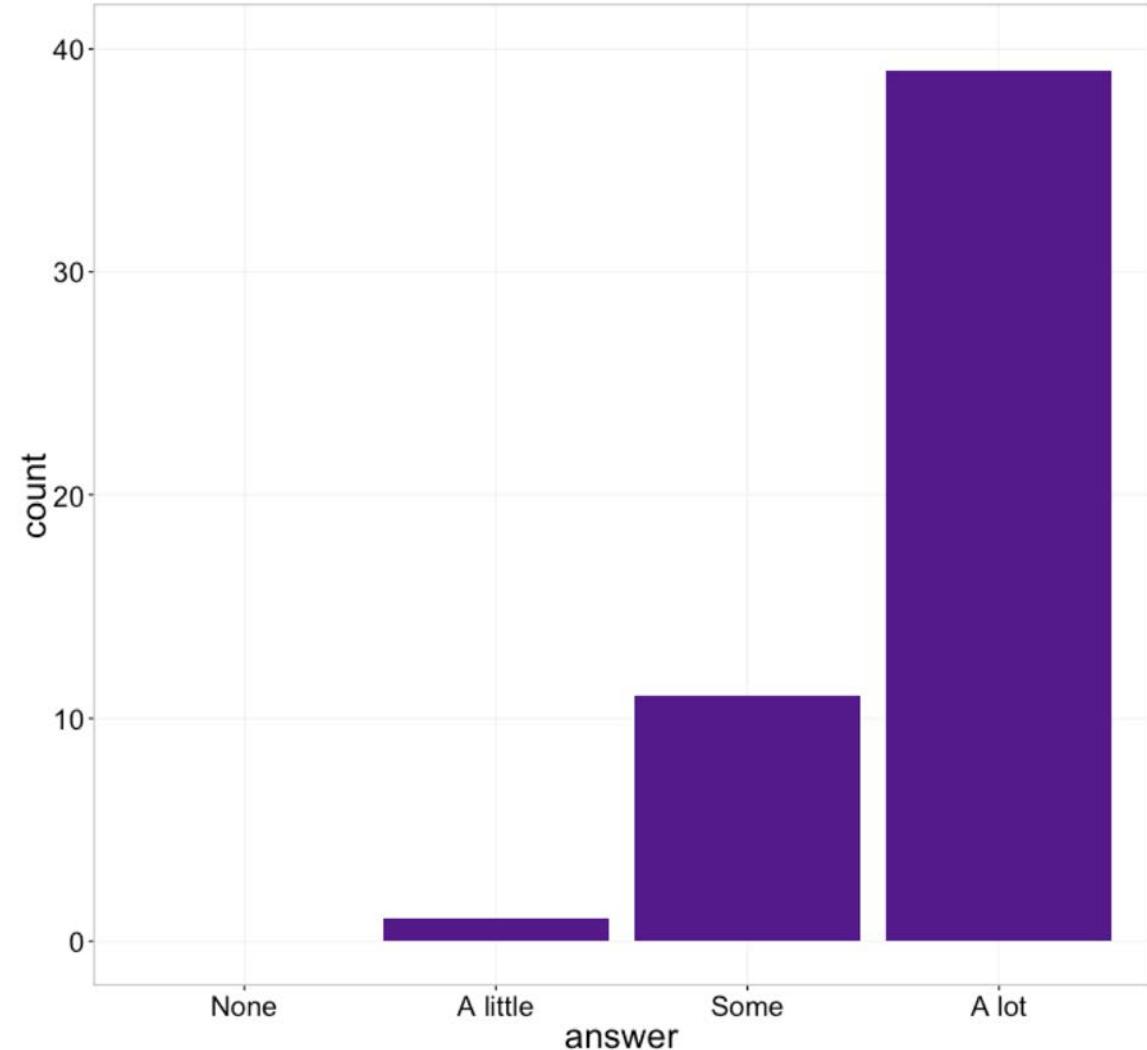
Keep all categories, manually set axes

```
scale_x_discrete(drop=FALSE)
scale_y_continuous(limits=c(0,40),
                   breaks=c(0,10,20,30,40),
                   minor_breaks=NULL)
```

How much experience do you have as a producer of network science research?

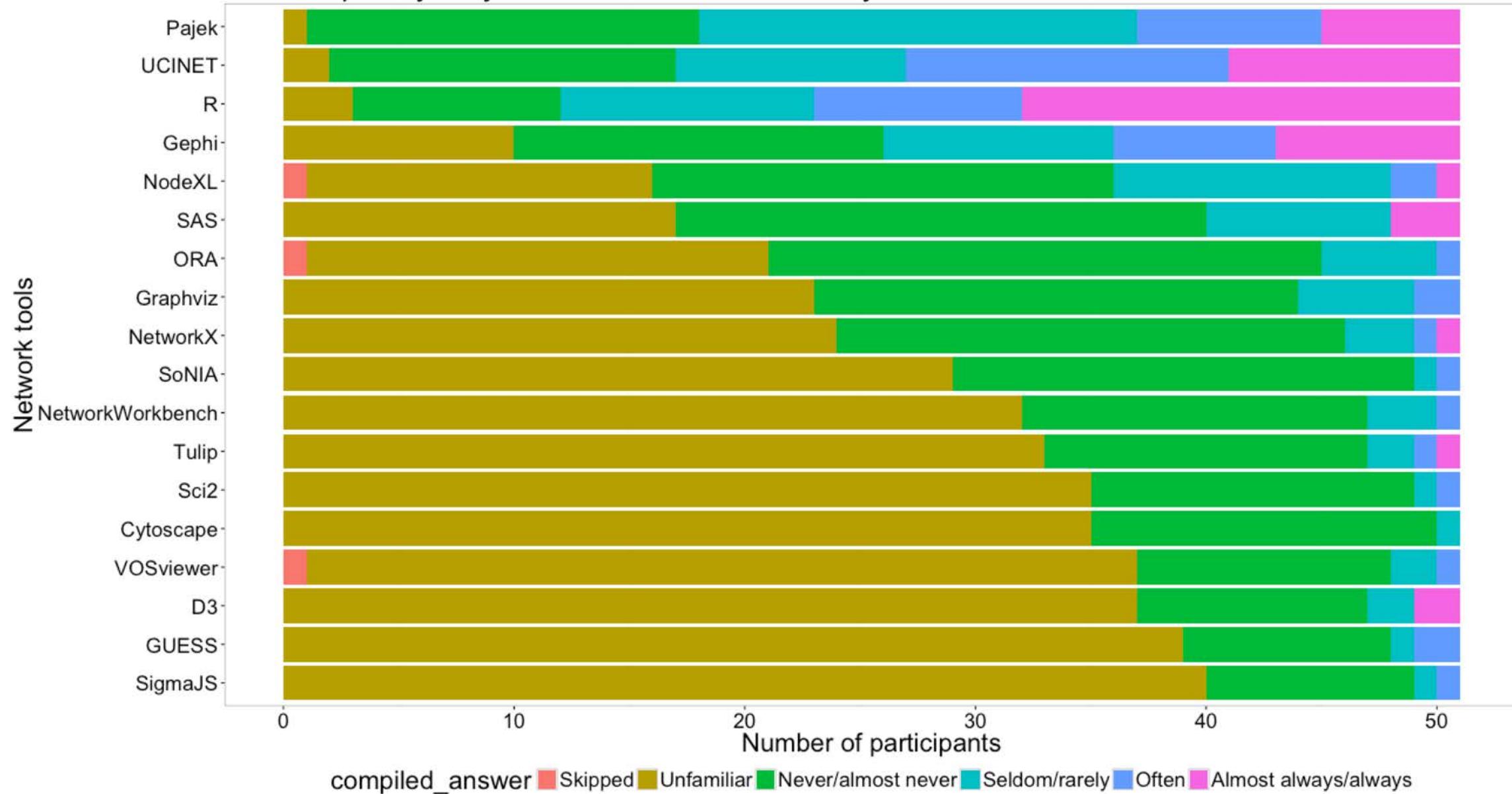


How much experience do you have as a consumer of network science research?



Principle 5:
Select meaningful colors

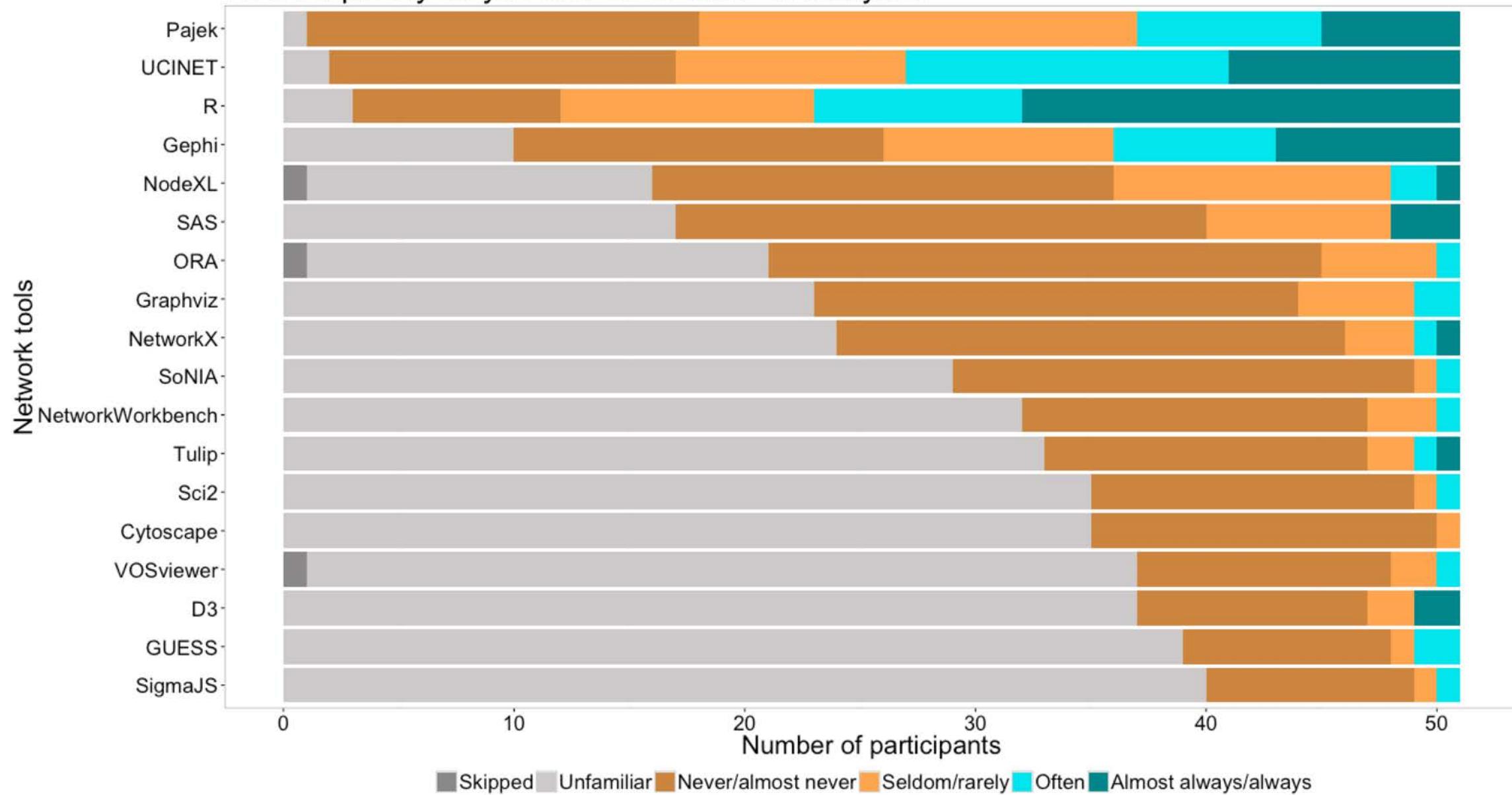
How frequently do you use these tools for analysis?



Select colors manually, or use alternate palette

```
scale_fill_manual(  
  values=c("snow4", "snow3",  
          "tan3", "tan1",  
          "turquoise2", "turquoise4"))  
  
scale_fill_manual(  
  values=c("#fee391", "#fe9929", "#cc4c02"))  
  
# Also see package RColorBrewer  
scale_fill_brewer(palette="BrBG")
```

How frequently do you use these tools for analysis?



Dataset 3: Star Wars opinion survey

<https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/>

ggplot2: Chart quirks

See “templates” file

<https://github.com/amzoss/RVis-DM2018/blob/master/Day%201/templates/templates.md>

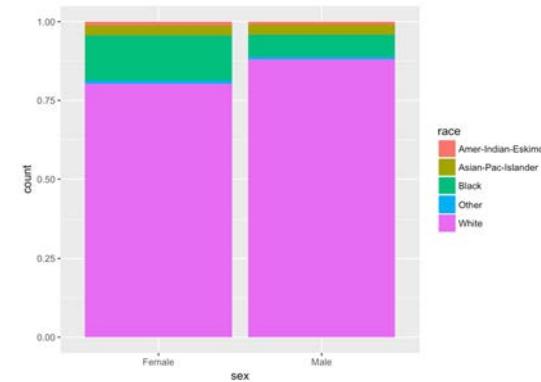
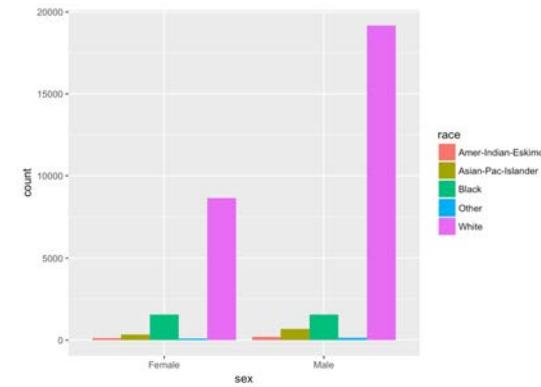
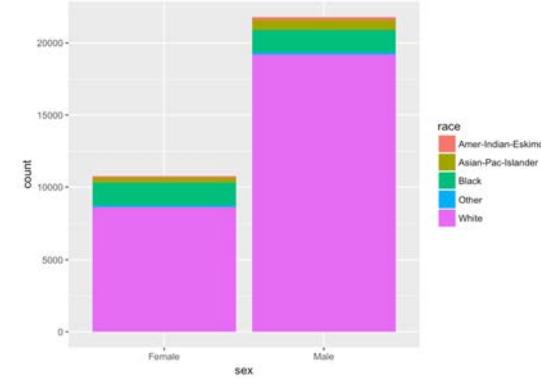
Chart components/slots

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)

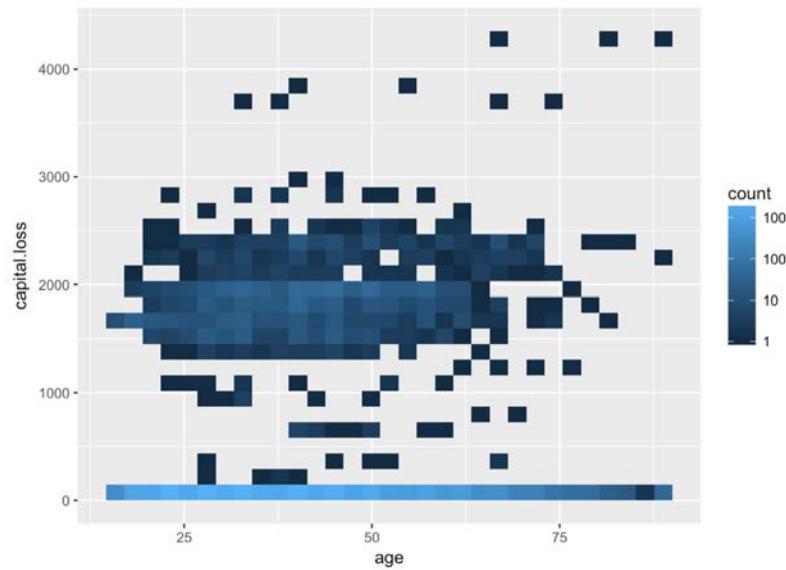
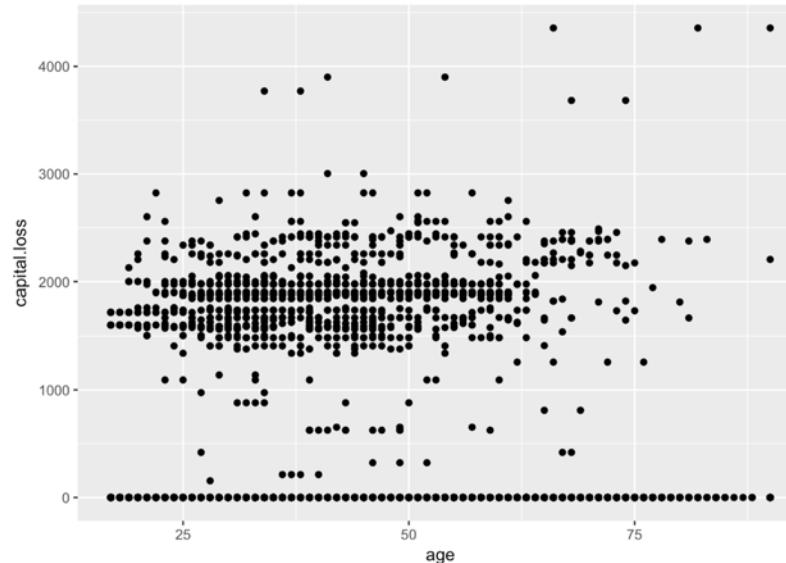
Bar chart

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



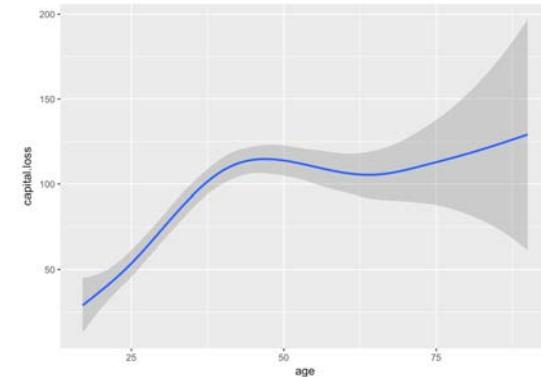
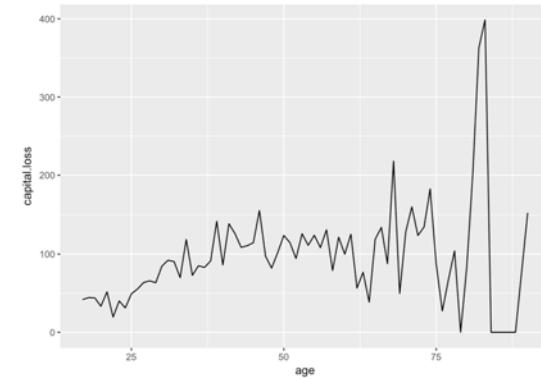
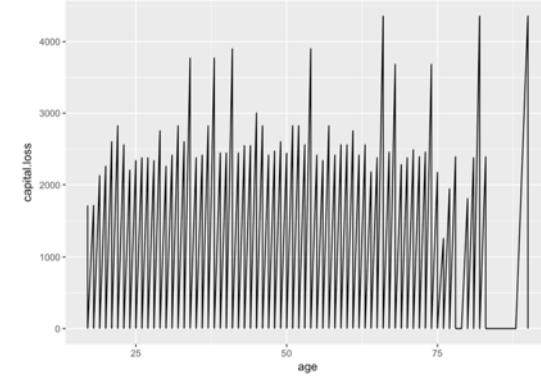
Scatter plot

- Overplotting
- point vs. bin2d



Line chart

- identity vs. summary
- line vs. smooth



ggplot2 Cheat Sheet

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' x 5' 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 + geom_blank()
b + geom_curve(aes(end = lat), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, alpha, angle, color, curvature, fontface, hjust, lineheight, size, vjust
c + geom_label(aes(label = cyl), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, alpha, color, fill, fontfamily, fontface, hjust, lineheight, size, vjust
a + geom_line(aes(group = group), lineend = "butt", linejoin = "round", linemetre = 1)
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_polygon(aes(group = group)) 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
b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) x, y, alpha, color, fill, group, linetype, size
```



#### LINE SEGMENTS



common aesthetics: x, y, alpha, color, linetype, size



```
b + geom_abline(aes(intercept = 0, slope = 1))
l + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))
b + geom_segment(aes(end = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:115, radius = 1))
```



Not required, sensible defaults supplied



#### TWO VARIABLES



continuous x, continuous y



```
e + geom_label(aes(label = cyl), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, alpha, color, fill, fontfamily, fontface, hjust, lineheight, size, vjust
a + geom_point() x, y, alpha, color, fill, shape, size, stroke
e + geom_quartile() x, y, alpha, color, group, linetype, size, weight
e + geom_rug(sides = "bl") x, y, alpha, color, group, linetype, size
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, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
a + geom_area() x, y, alpha, color, fill, group, linetype, size
b + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, size, weight
f + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group
f + geom_hex() x, y, alpha, color, fill, group, linetype, size
f + geom_violin(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight
```



discrete x, continuous y



```
f + geom_col() x, y, alpha, color, fill, group, linetype, size
f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, size, weight
f + geom_errorbar() x, y, max, ymin, alpha, color, fill, group, linetype, size, width(also geom_errorbarh())
j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size
j + geom_pointrange() x, y, min, max, alpha, color, fill, group, linetype, size
```



discrete x, discrete y



```
g + geom_count() x, y, alpha, color, fill, shape, size, stroke
c + geom_map(aes(map_id = state), map = map) + expand_limits(x = map$long, y = map$lat), map_id, alpha, color, fill, group, linetype, size
```



#### 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, unemploy))



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



#### Visualizing error



d < data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)



```
j + geom_crossbar(fatten = 2) x, y, ymax, ymin, alpha, color, fill, group, linetype, size
j + geom_errorbar() x, y, max, ymin, alpha, color, fill, group, linetype, size, width(also geom_errorbarh())
j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size
j + geom_pointrange() x, y, min, max, alpha, color, fill, group, linetype, size
```



maps



```
data = data.frame(murder = USAreests$Murder, state = followrownames(USAreests))
state$group = state %>% group_by(state)
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, group, linetype, size
```



#### THREE VARIABLES



```
sealsSz <- with(seals, sqrt(delta.long^2 + delta.lat^2)) l < ggplot(seals, aes(long, lat))
l + geom_raster(aes(fill = z)) x, y, z, alpha, colour, group, linetype, size, weight
l + geom_contour(aes(fill = z)) x, y, z, alpha, color, fill, group, linetype, size, weight
l + geom_bar() x, alpha, color, fill, linetype, size, weight
d + geom_bar() x, alpha, color, fill, linetype, size, weight
```



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


```

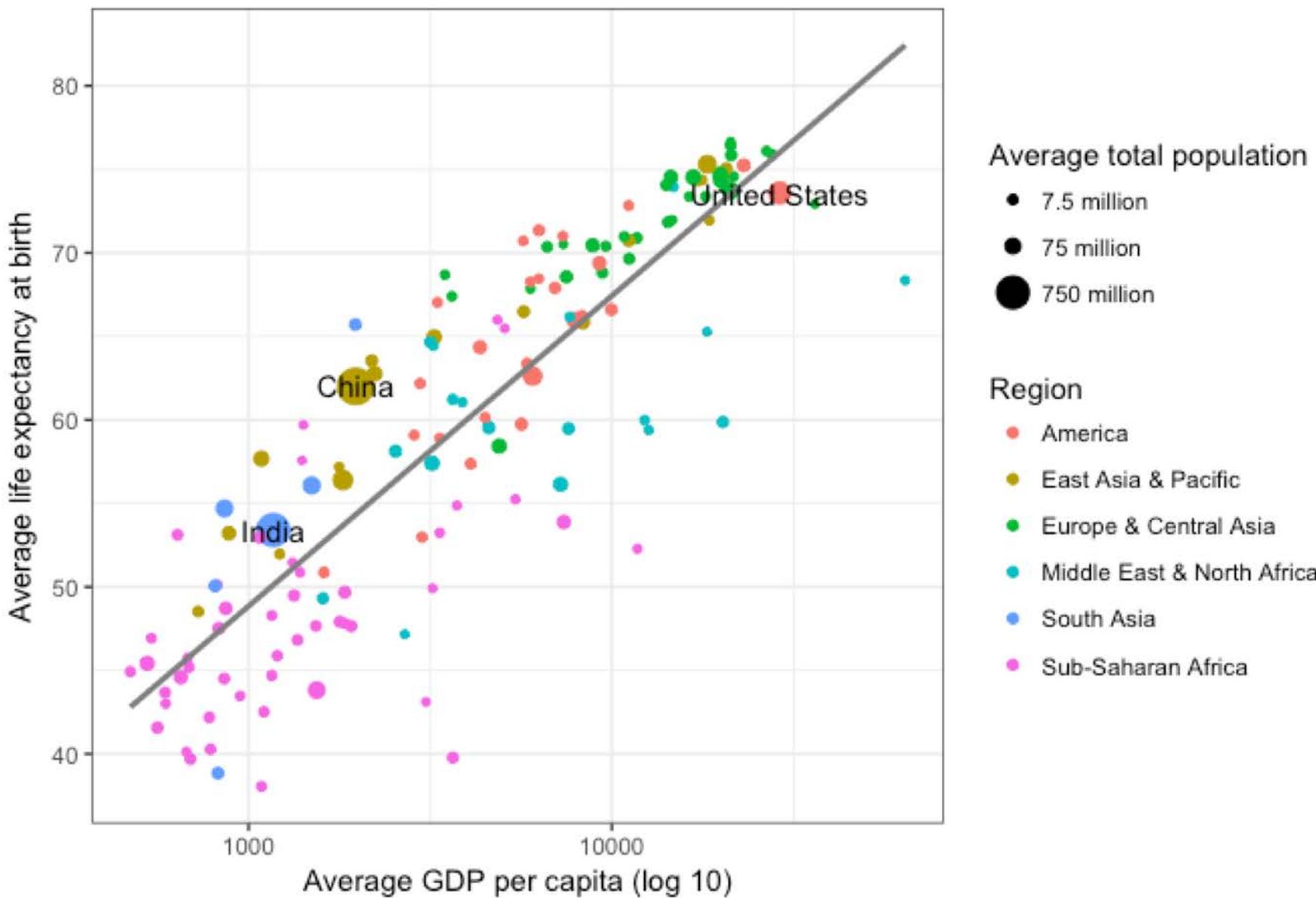


<https://www.rstudio.com/resources/cheatsheets/#ggplot2>

Dataset 4: Gapminder Data

<http://www.gapminder.org/>

Averages across all years of the traditional Gapminder dataset



Advanced topics:
Mapping, saving charts out

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>

Videos of past workshops

The image shows a screenshot of a Panopto video player interface. At the top left is the Panopto logo and the title "Figures and Posters". To the right are links for "March 4, 2016 in DVS Training", "Help", and "Sign in". On the left side of the main content area, there is a thumbnail image of two people standing in front of a whiteboard in a classroom setting. Below the thumbnail is a search bar with the placeholder "Search this recording" and a magnifying glass icon. Underneath the search bar are buttons for "Discussion" and "Sign in to ask a question or share a comment". The main content area features a large title "Designing Academic Figures and Posters" in bold black font, followed by the date "March 4, 2016" and a link "Slides: <http://duke.box.com/PostersSpring2016>". Below the title, there are two sections: "Angela Zoss" (Data Visualization Coordinator, Data and Visualization Services) and "Eric Monson" (Data Visualization Analyst, Data and Visualization Services). At the bottom of the video player are playback controls: a play button, a 10-second backward button, a 00:03 timestamp, a circular progress bar, a 1:22:45 timestamp, a 1X speed button, a quality button, and a hide button. Below the controls, there are four thumbnail images of slides from the presentation, each with a timestamp: 1:22, 4:32, 7:32, and 10:32. The slide thumbnails include titles like "Good Posters" and "Purpose of a poster".

<http://bit.ly/DVsvideos>

Thanks for your feedback!

angela.zoss@duke.edu