

# Visualization in R using ggplot2

Angela Zoss

1/25/19

<https://github.com/amzoss/ggplot2-S19>

# 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!

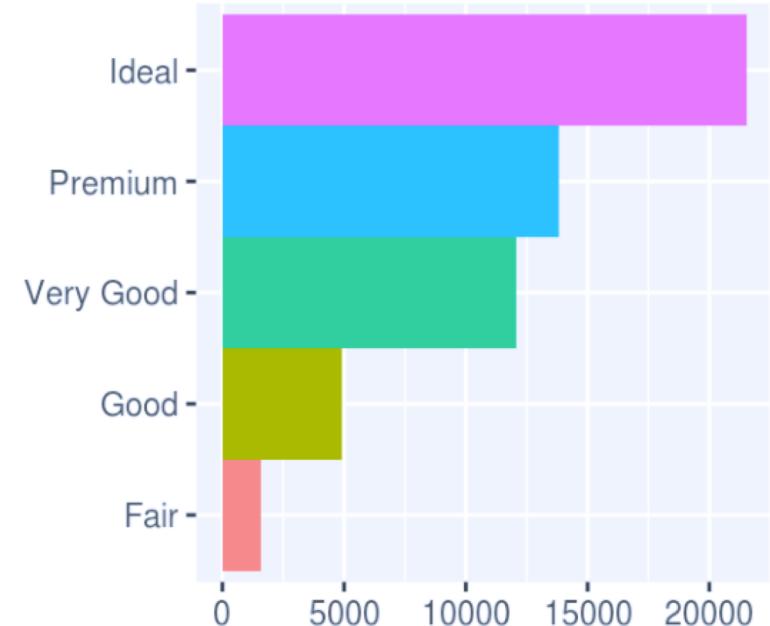
*“Your closest collaborator is **you** six months ago,  
but you don’t reply to emails.”*

- *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 (2<sup>nd</sup> ed)*. New York: Springer.

# Why ggplot2 instead of base R?

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

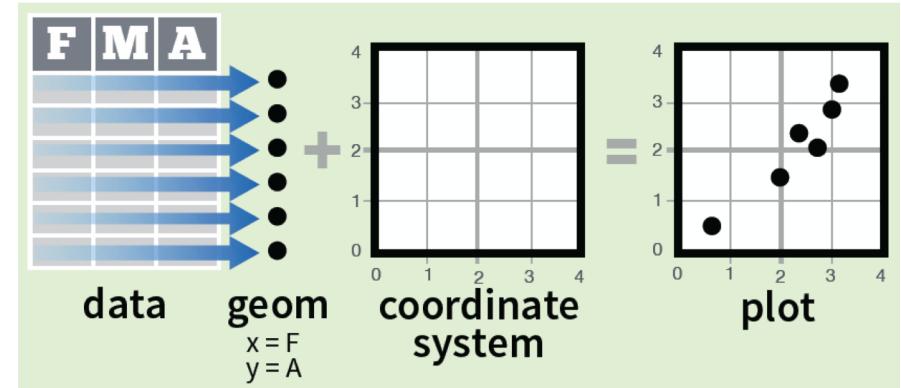
“Why I use ggplot2”, David Robinson

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

# ggplot2: Elements

# Basic elements in any ggplot2 visualization

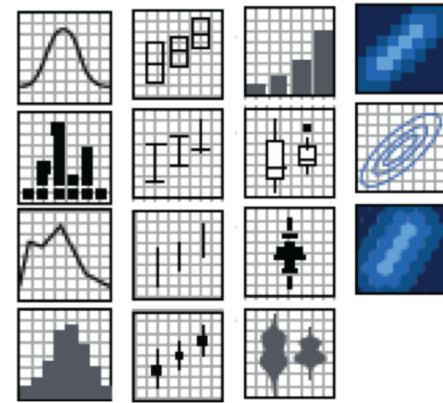
- **data**
- **aesthetics**  
(variable mappings)
- **geom**  
(chart type or shape)
- coordinate system  
(the arrangement of the marks;  
most geoms use default, cartesian)



<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

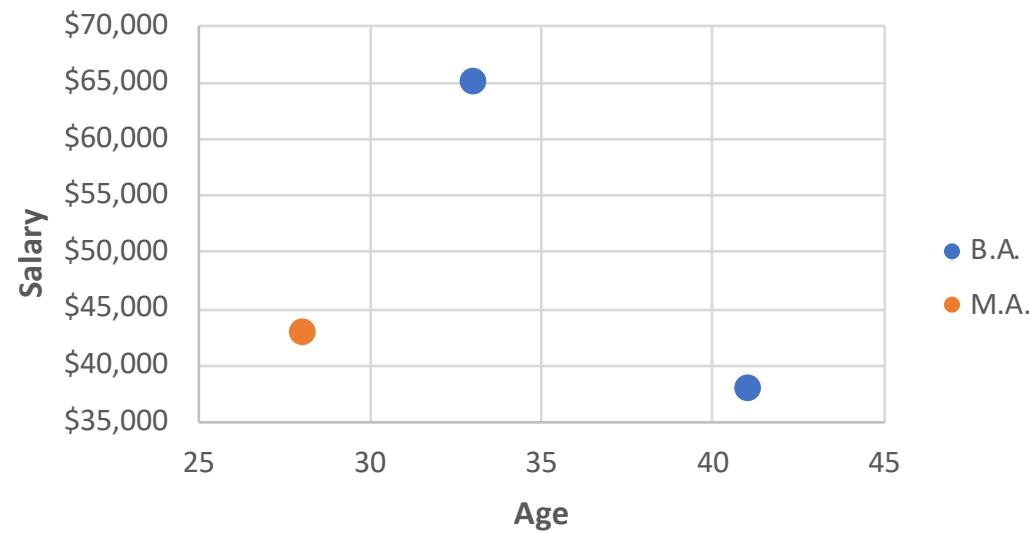
```
ggplot( data = data frame )
```

+

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

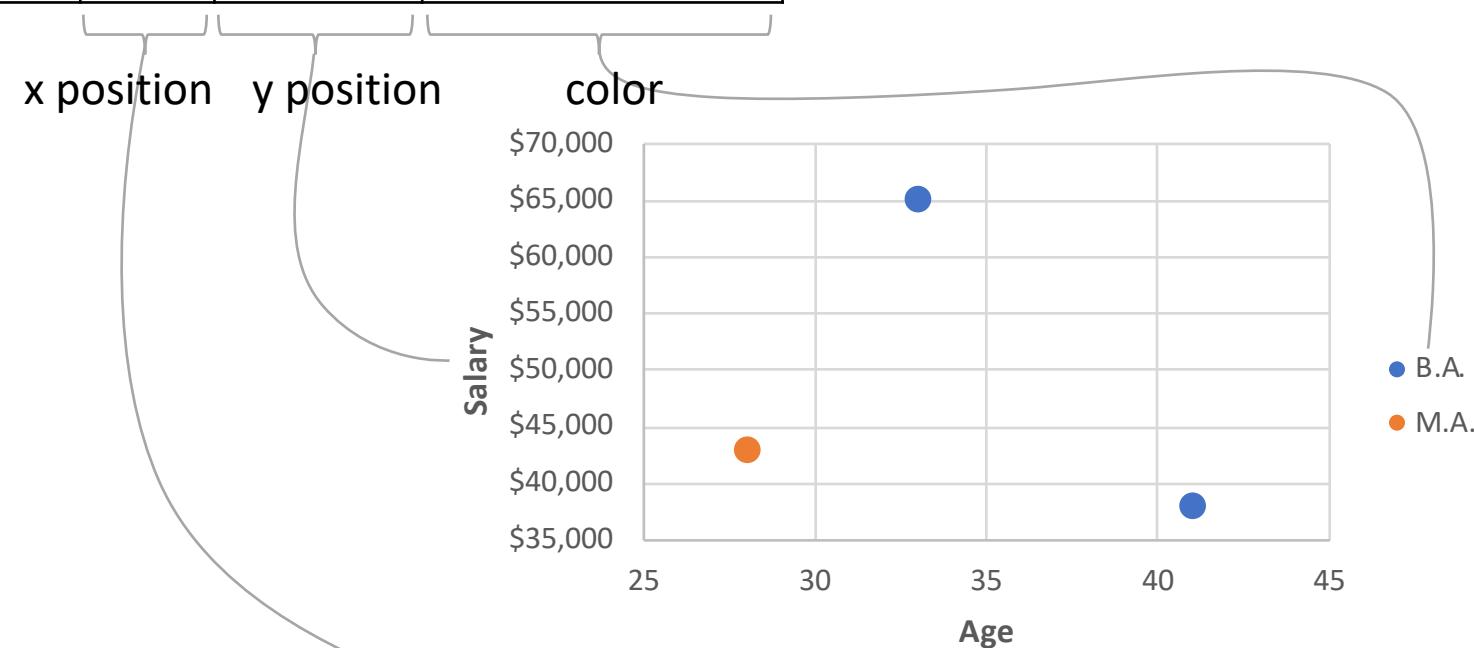
# Aesthetic variable mappings

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

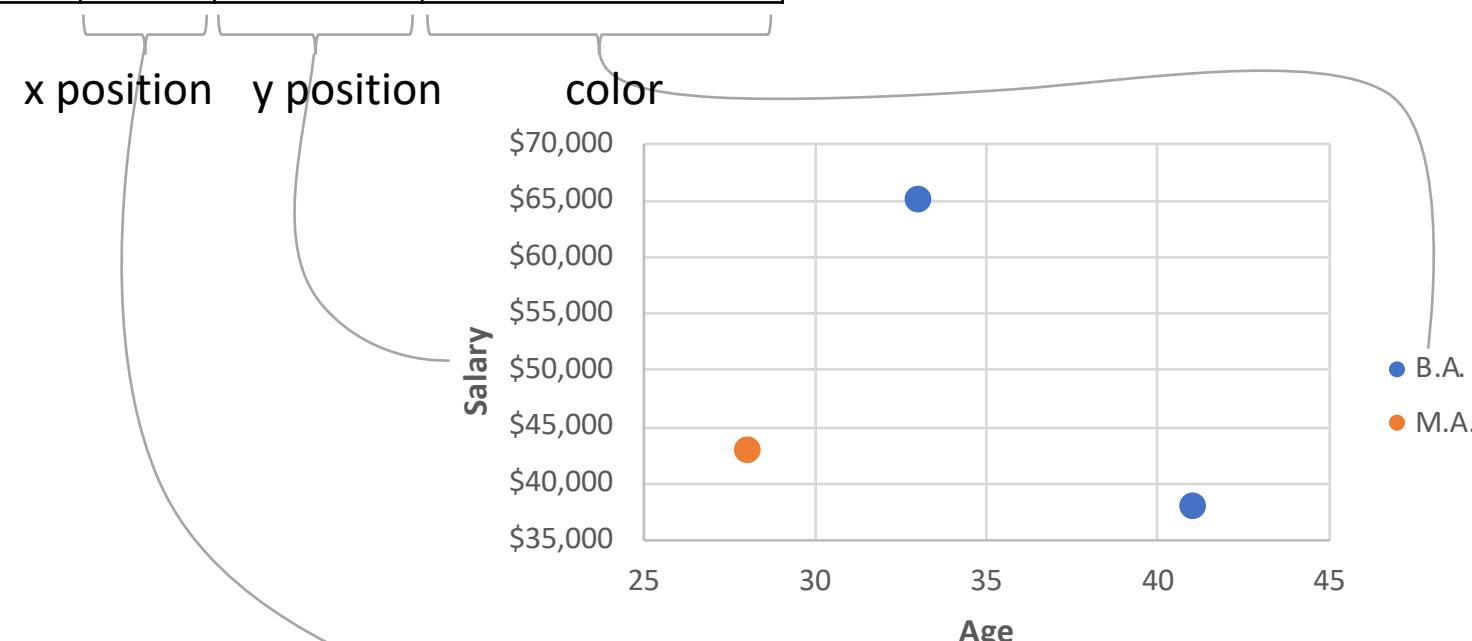
Name	Age	Salary	Highest Degree
Jane Smith	33	\$65,000	B.A.
Abby Jones	28	\$43,000	M.A.
Bridget Carden	41	\$38,000	B.A.



# 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.

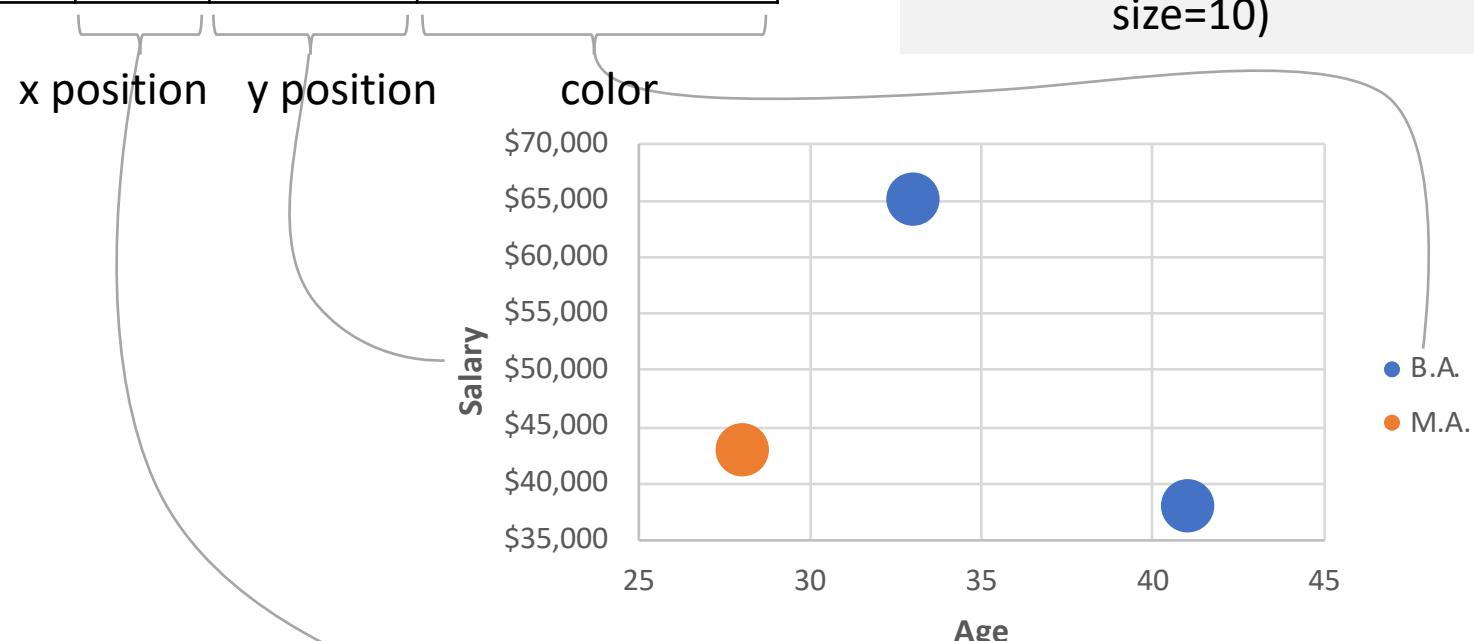
```
ggplot(data) +  
  geom_point(  
    aes(x=age,  
        y=salary,  
        color=degree))
```



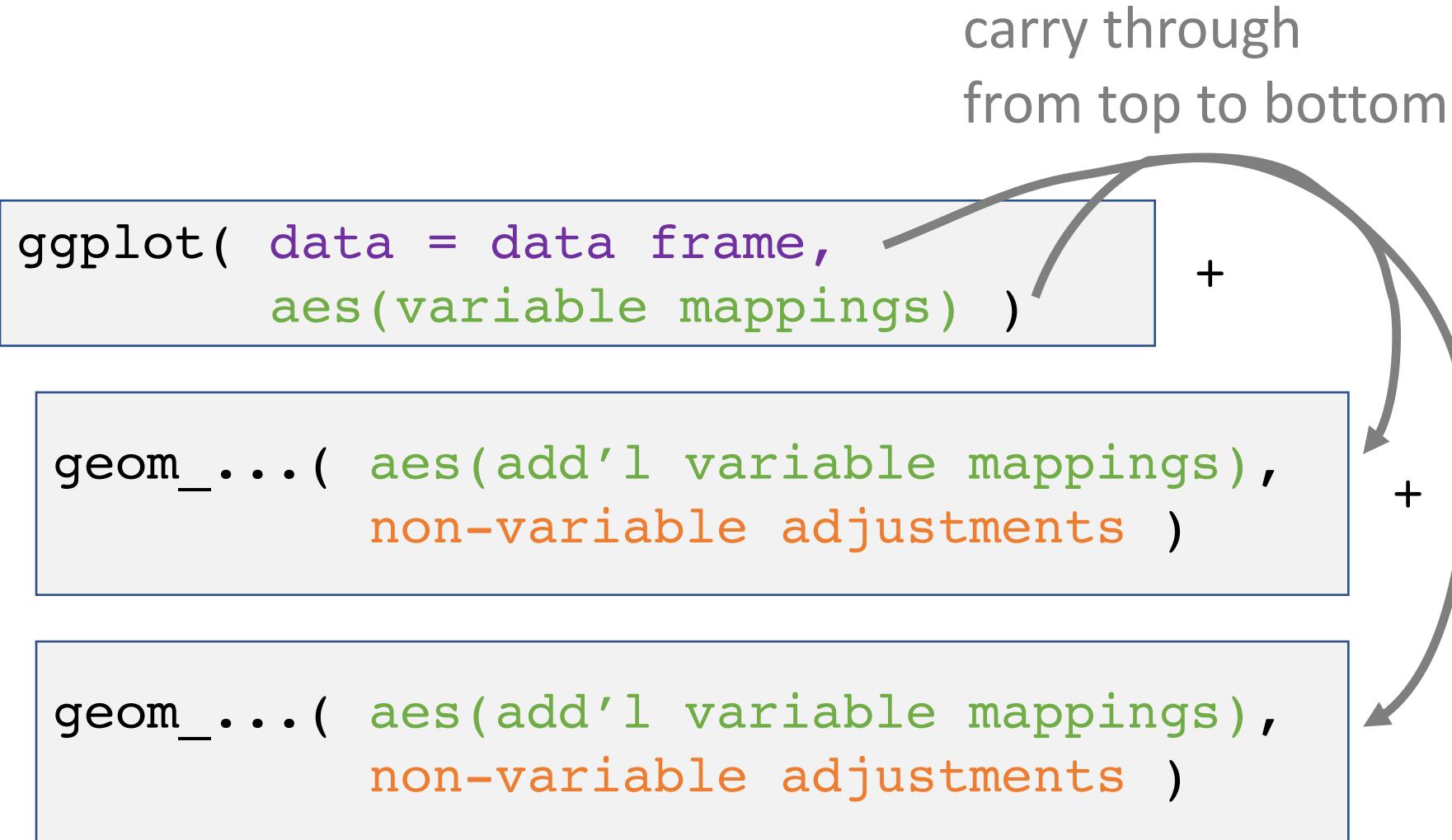
# Non-variable adjustments

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



# Set up environment

- R
- RStudio
- tidyverse package

Don't have it installed?

<https://vm-manage.oit.duke.edu/containers>

# Get workshop files

URL: <https://github.com/amzoss/ggplot2-S19>

## With Git installed

In RStudio:

- Project → New project
- Version Control
- Git
  - Paste in GitHub URL
  - Project directory name: **ggplot2-S19**
  - 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

# Using RStudio

- Projects
- Rmarkdown
- Cheat sheets

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

# Why Rmarkdown?

- 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.

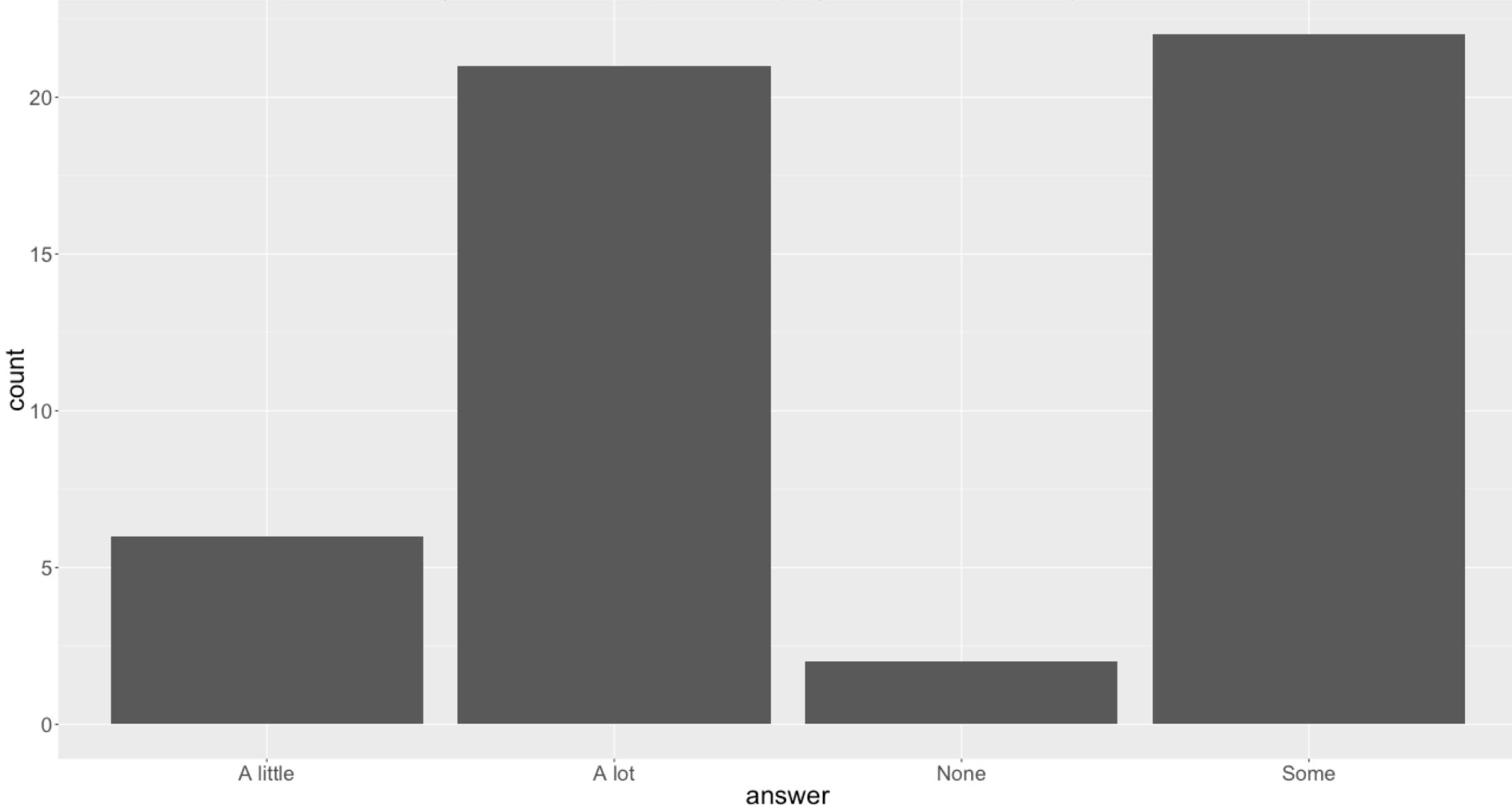
# Game of Thrones character ratings

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

# Principles for Effective Visualizations

Principle 1: Order matters

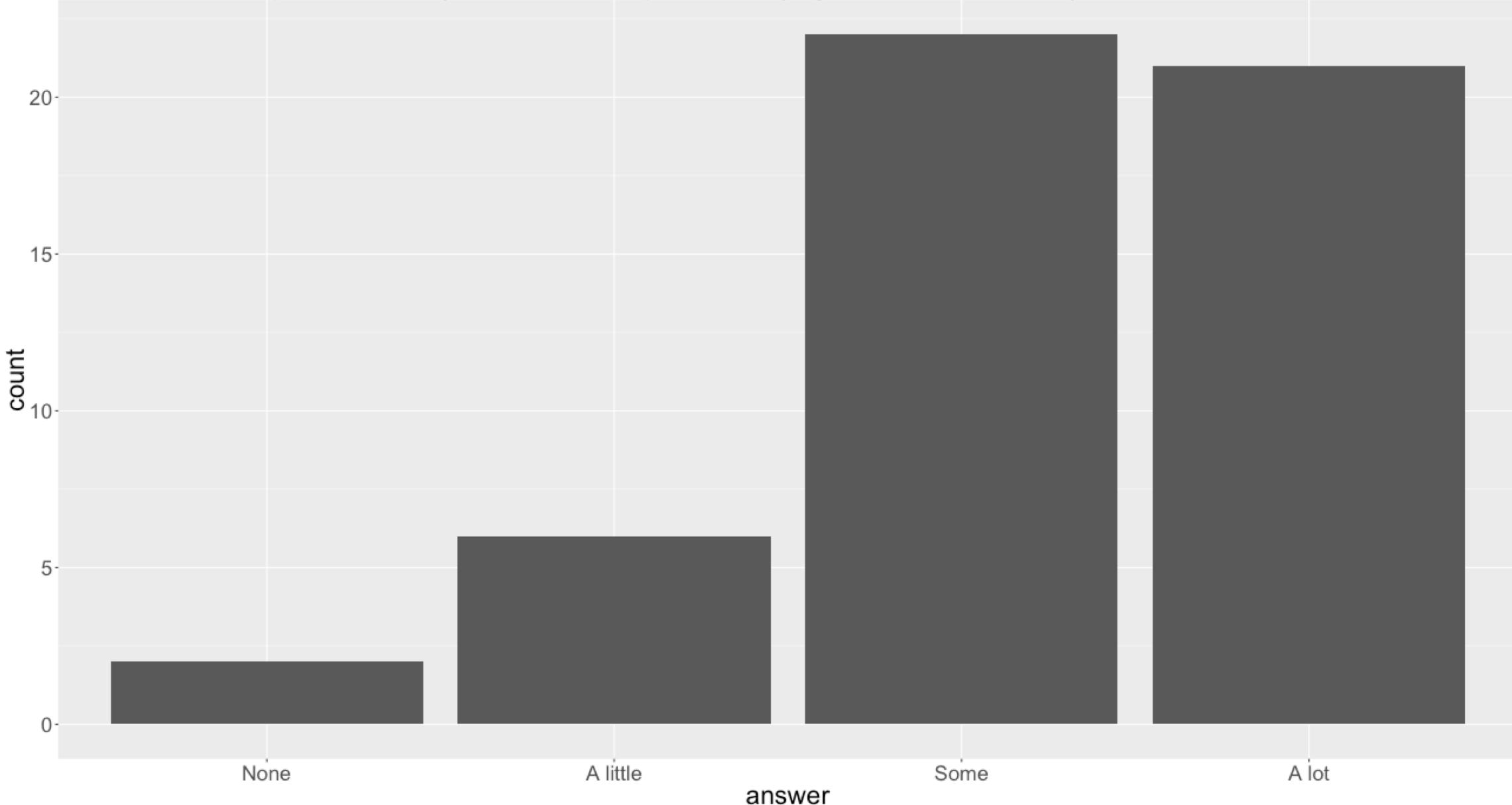
How much experience do you have as a producer (e.g., reader, follower) of network science research?



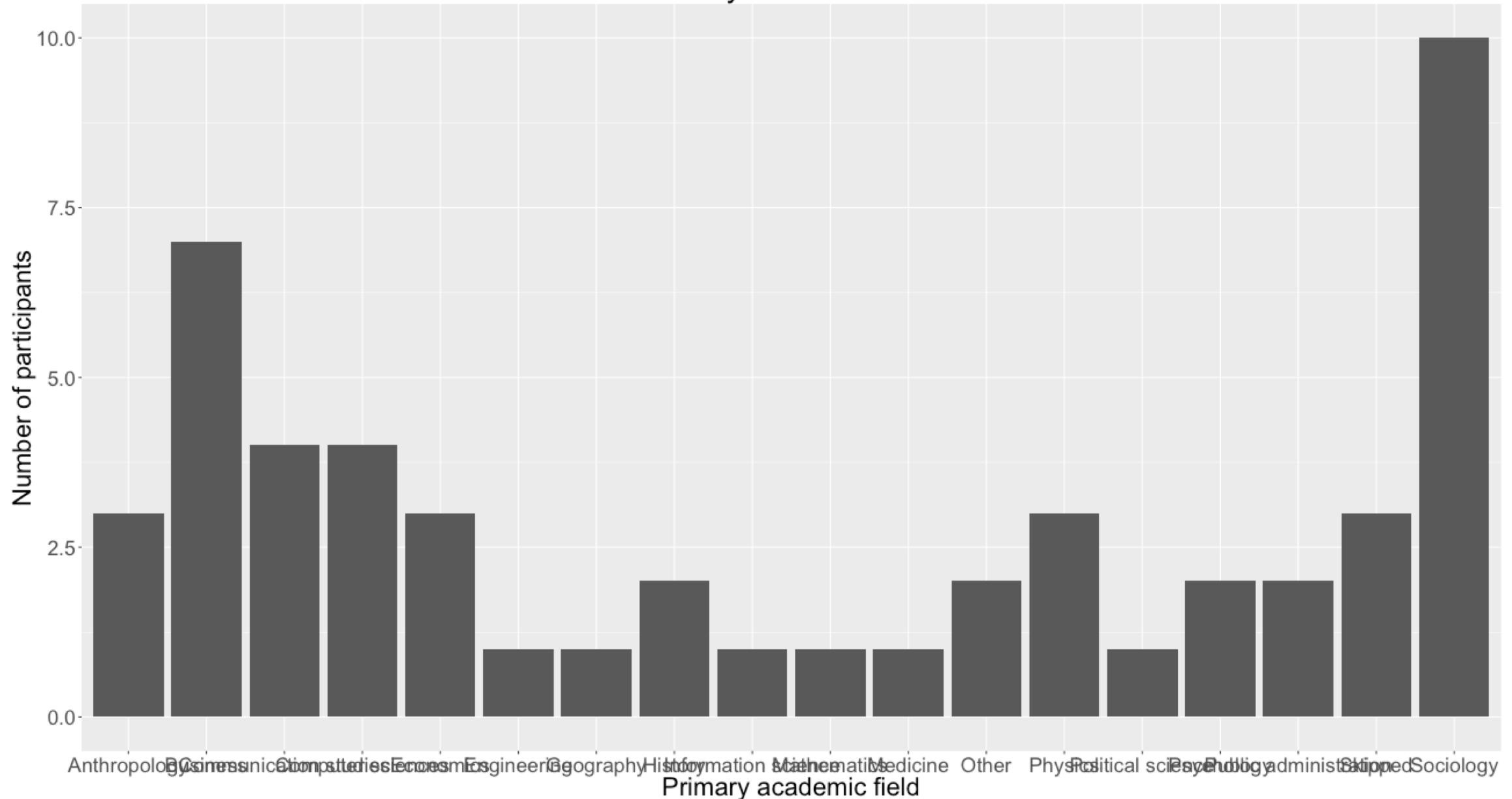
# Order by meaning

```
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

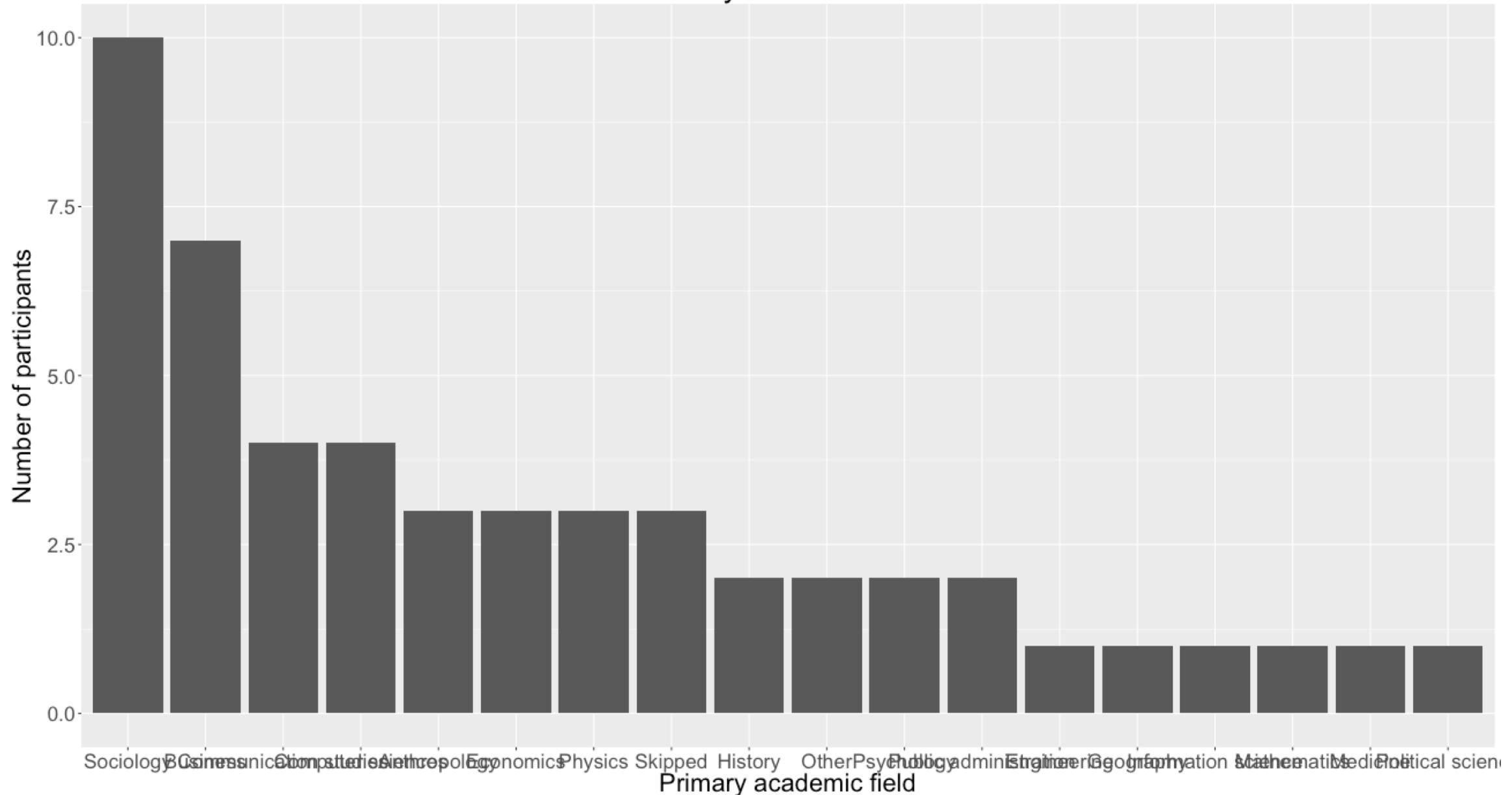


# Order by value

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

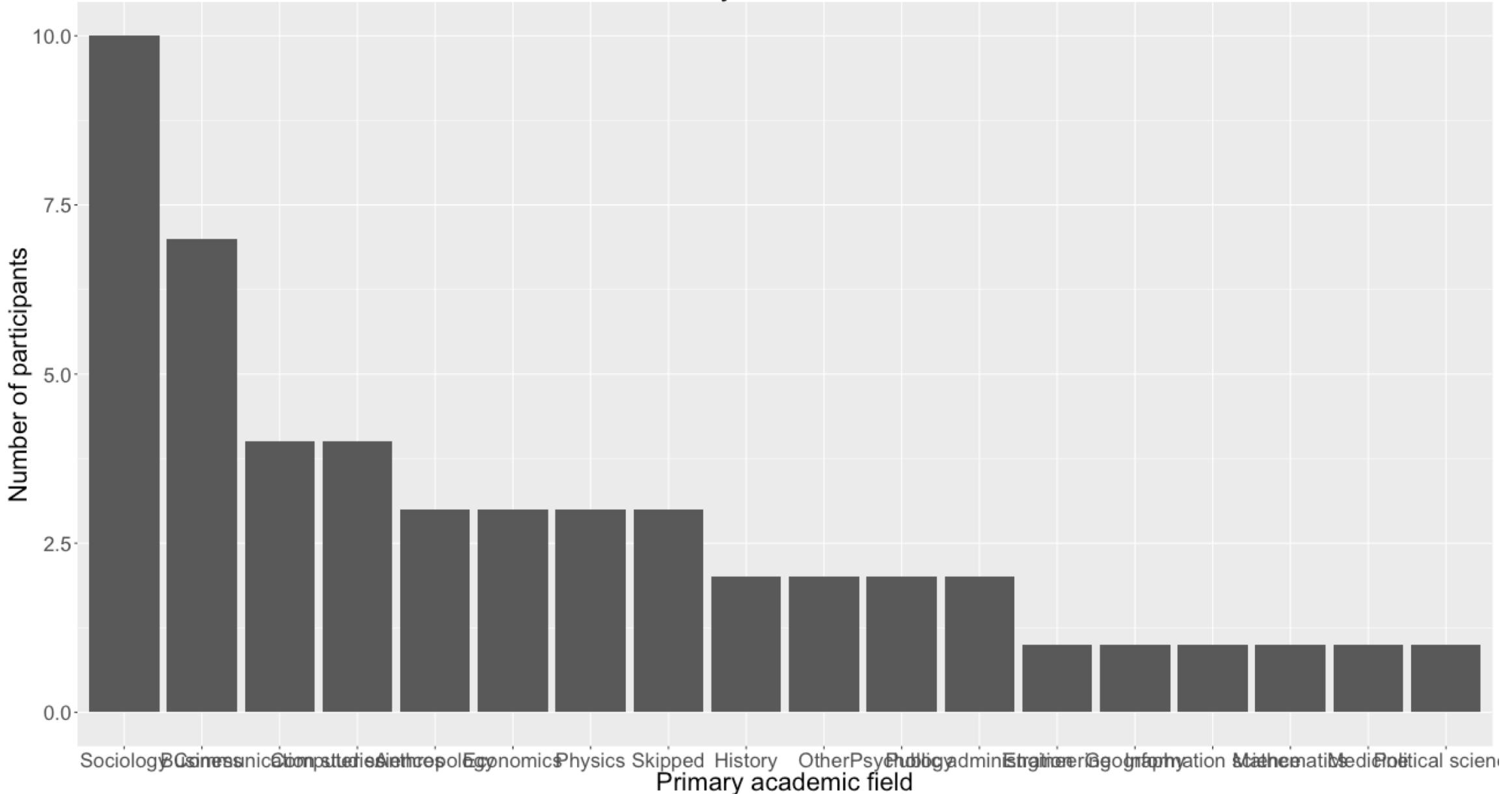
```
data$academic_field <-  
  fct_infreq(as_factor(data$academic_field))
```

## Primary academic field



Principle 2:  
Put long categories on y-axis

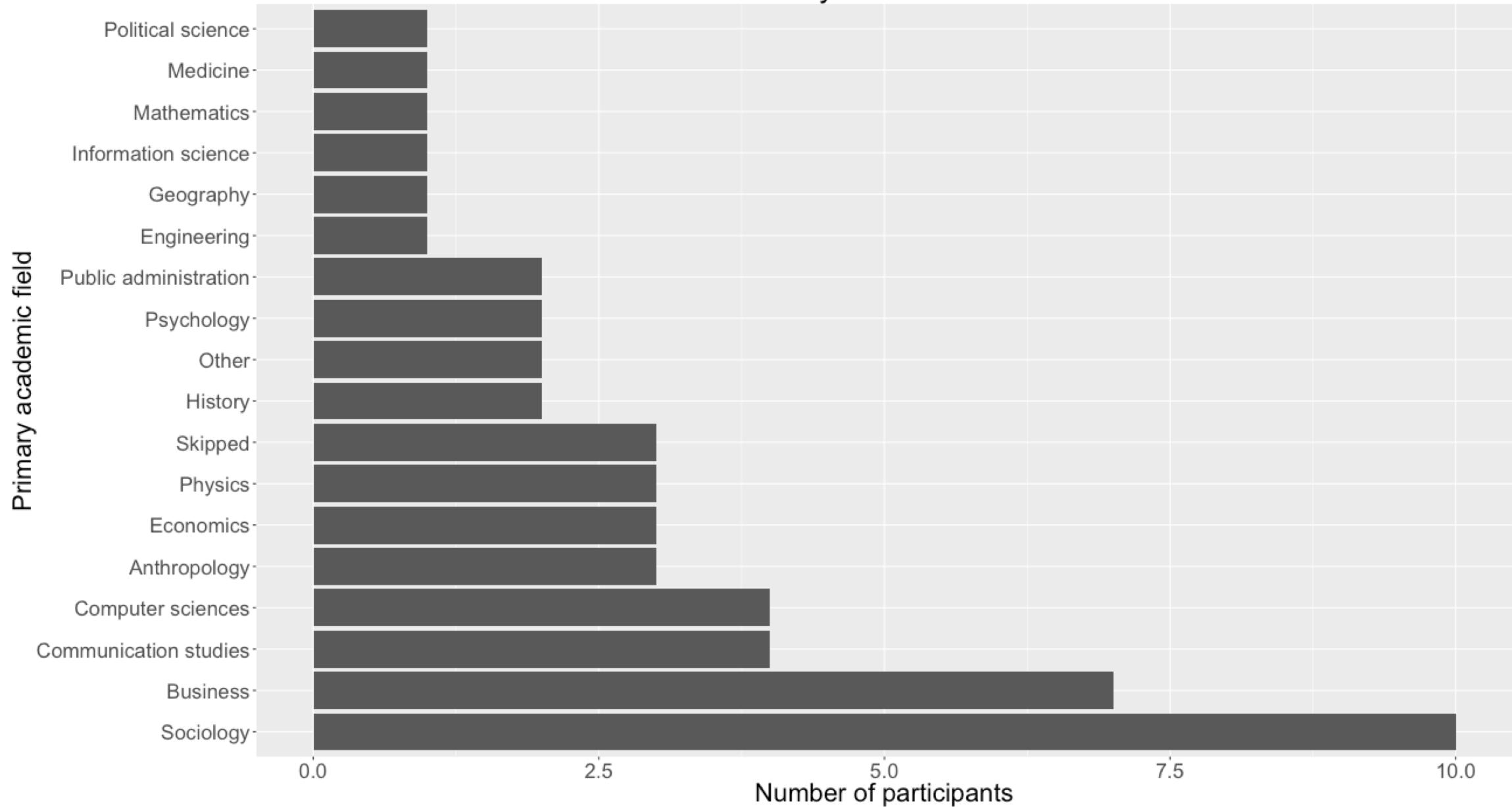
## Primary academic field



# Flip the axes

```
coord_flip()
```

## Primary academic field

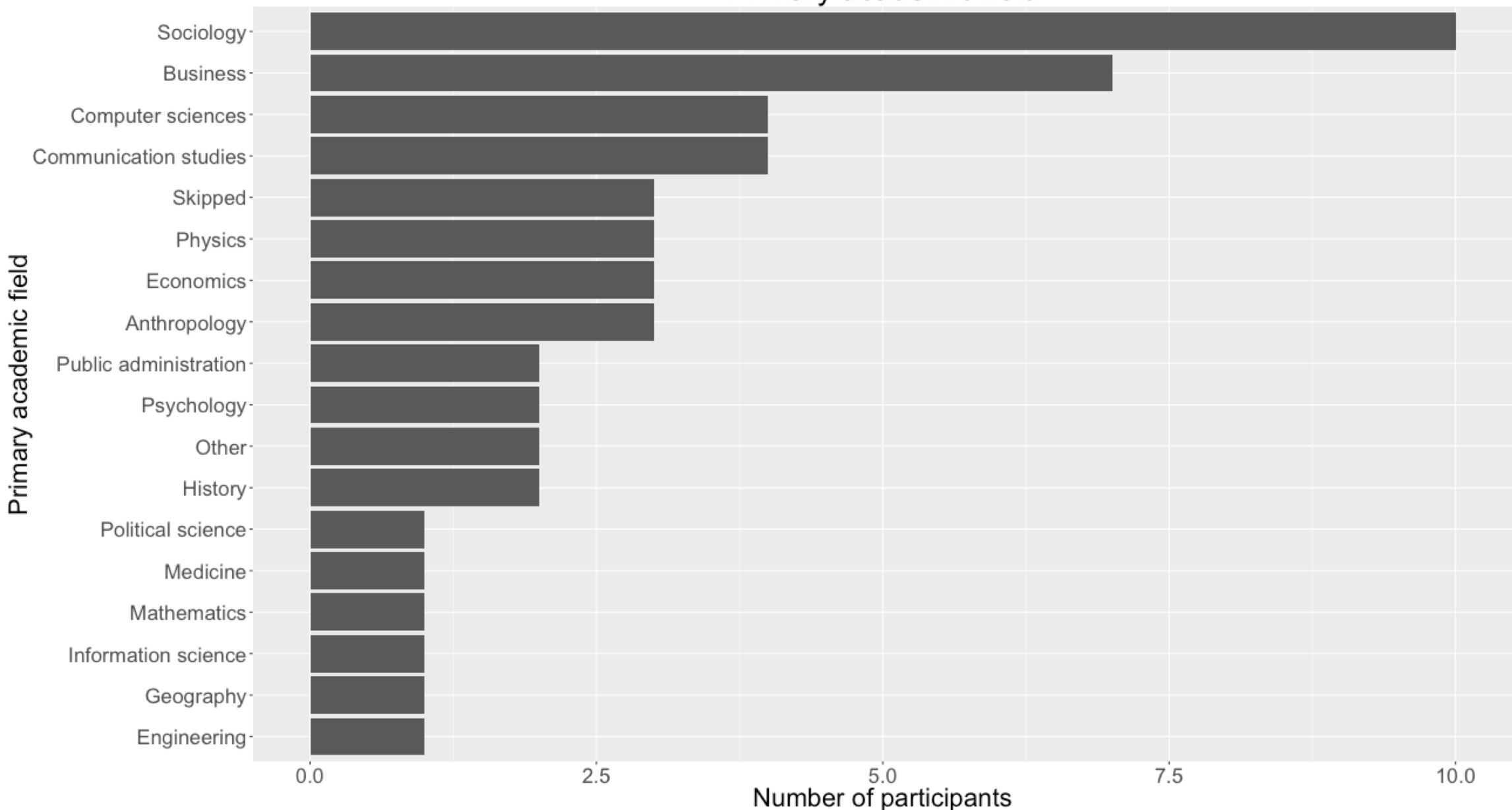


# Oops!

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

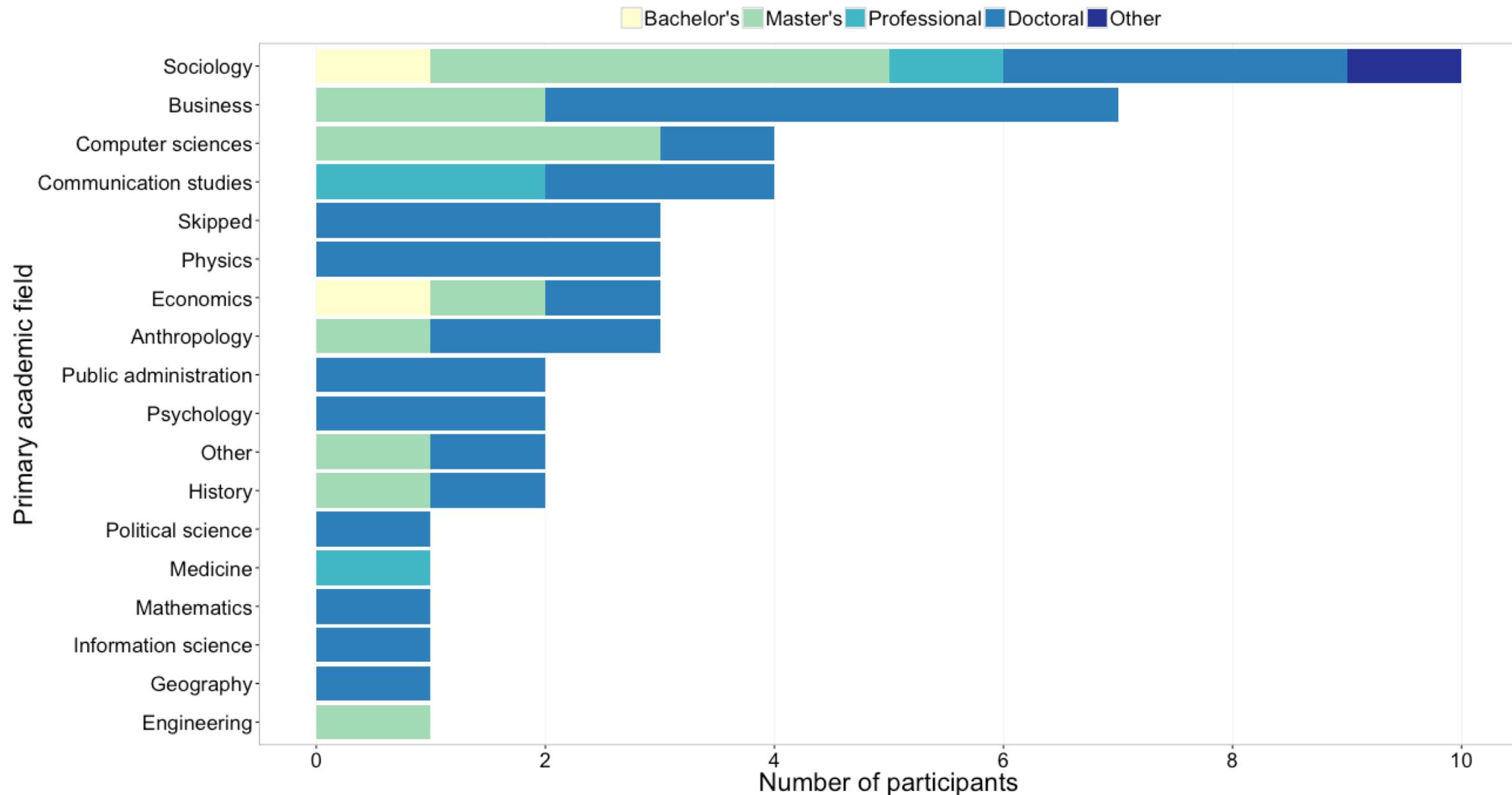
```
data$academic_field <-  
  fct_rev(fct_infreq(as_factor(data$academic_field))))
```

## Primary academic field

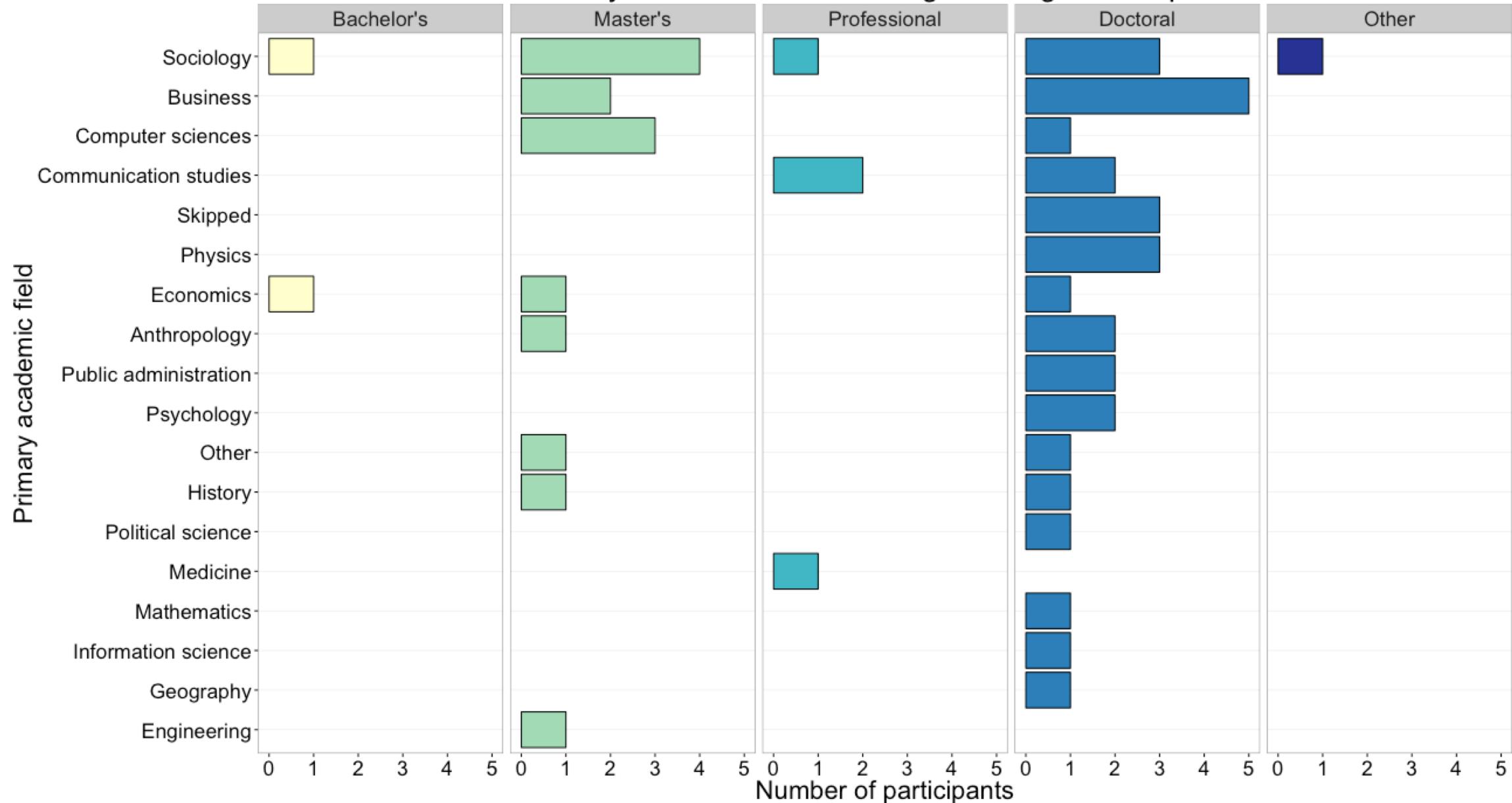


Principle 3: Pick a purpose

## Primary academic field and highest degree completed



## Primary academic field and highest degree completed



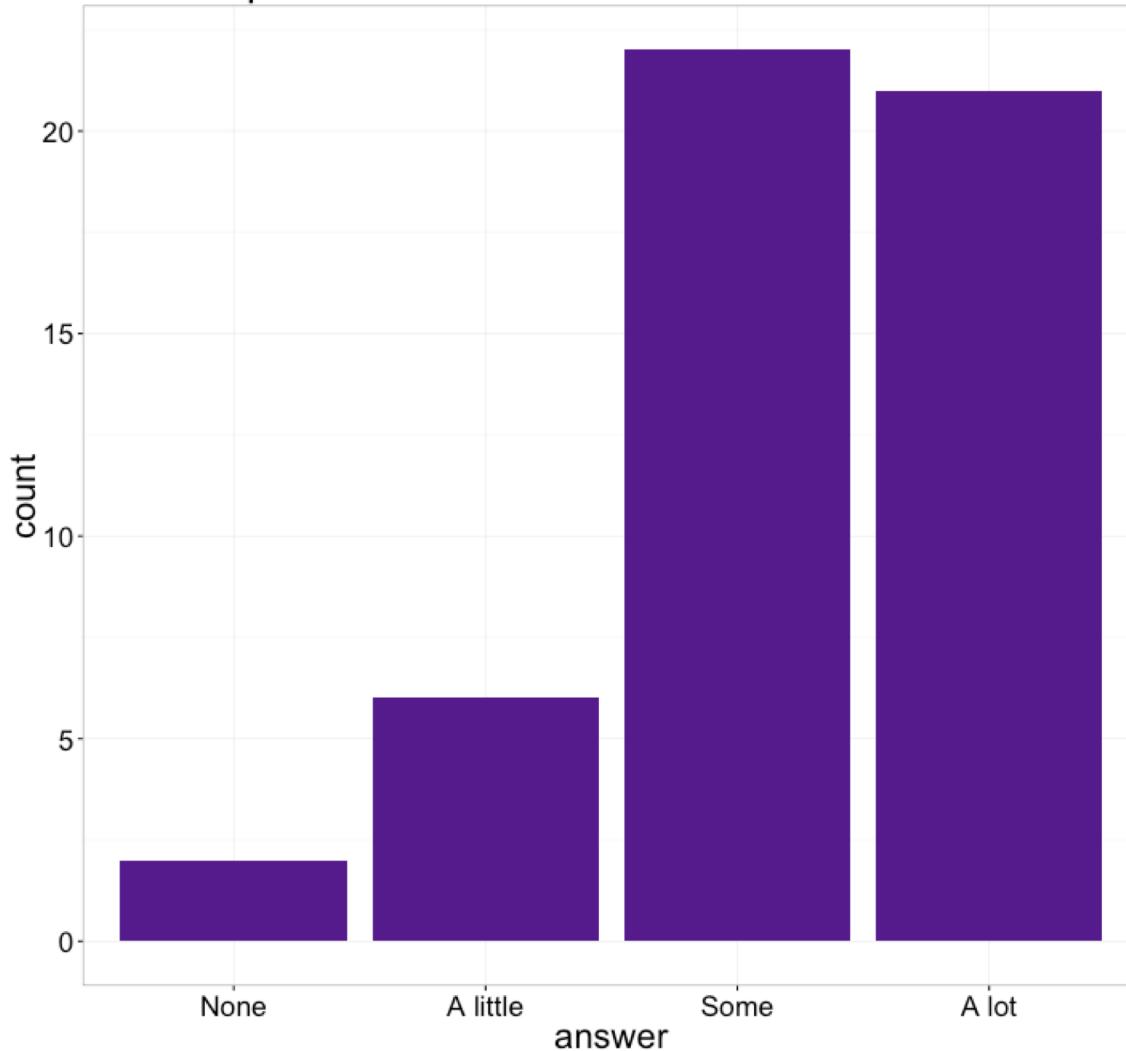
# Different placement helps with different comparisons

```
fill=highest_degree
```

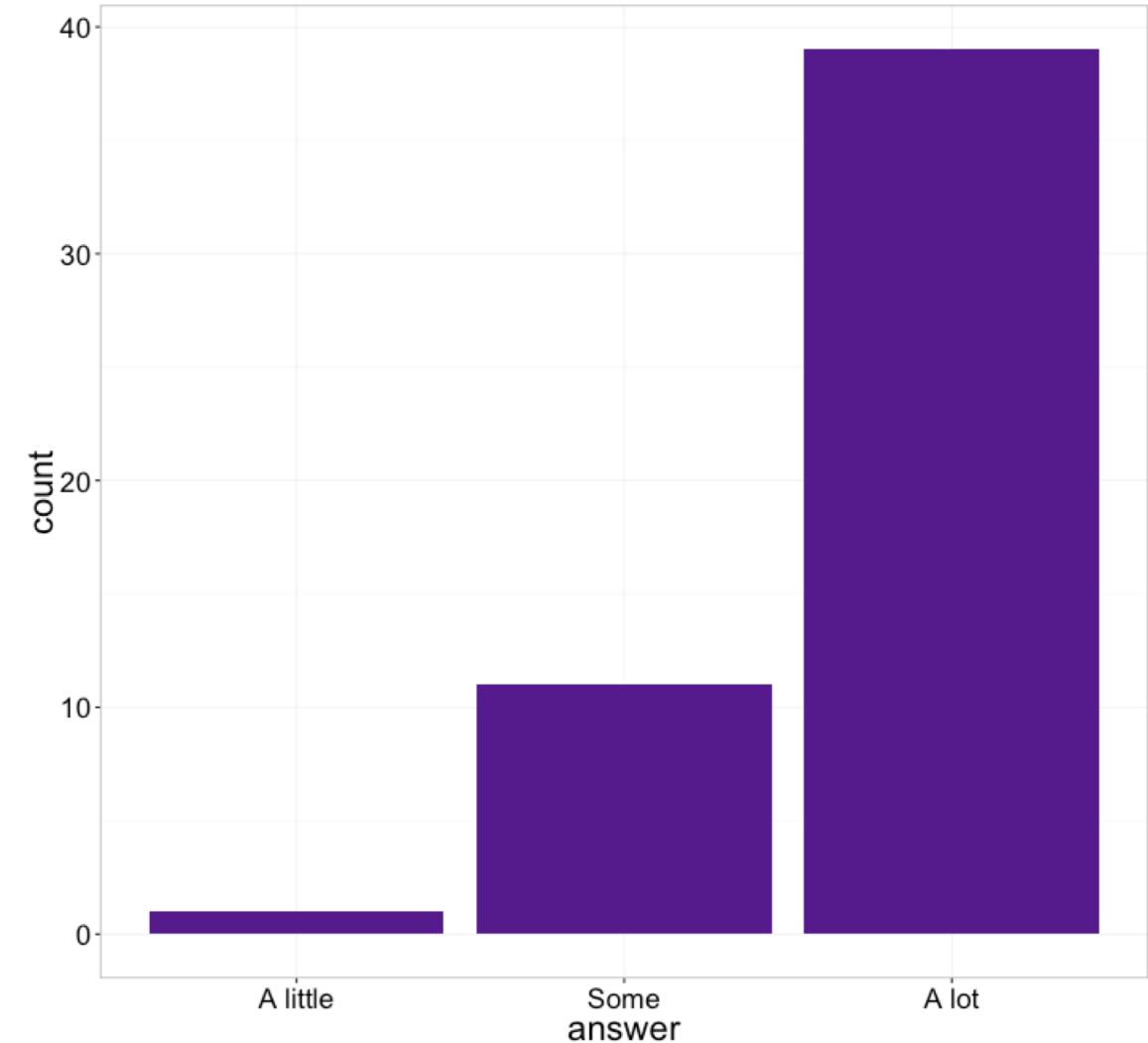
```
facet_grid(.~highest_degree)
```

Principle 4:  
Keep scales consistent

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



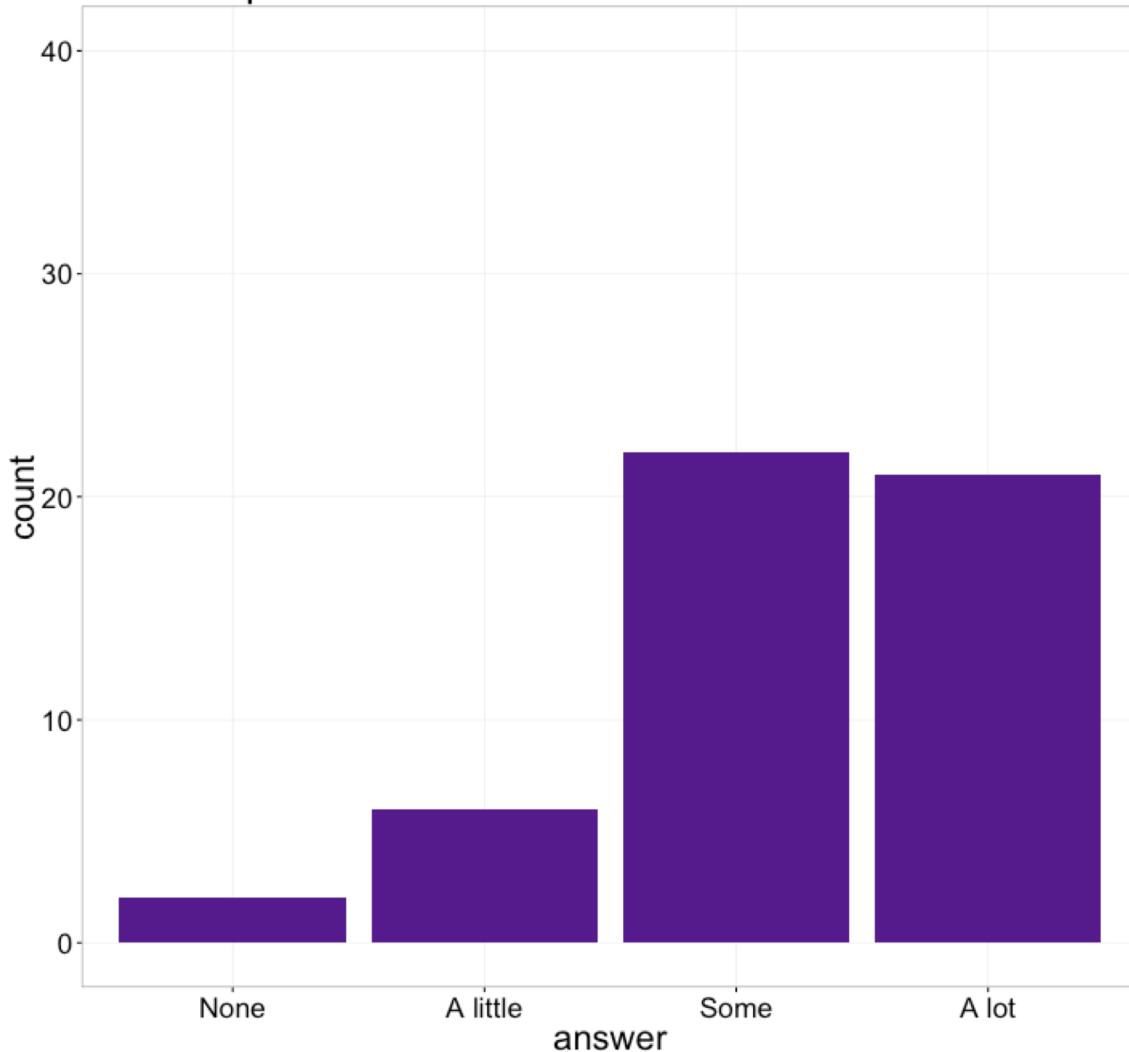
How much experience do you have as a consumer of network science research?



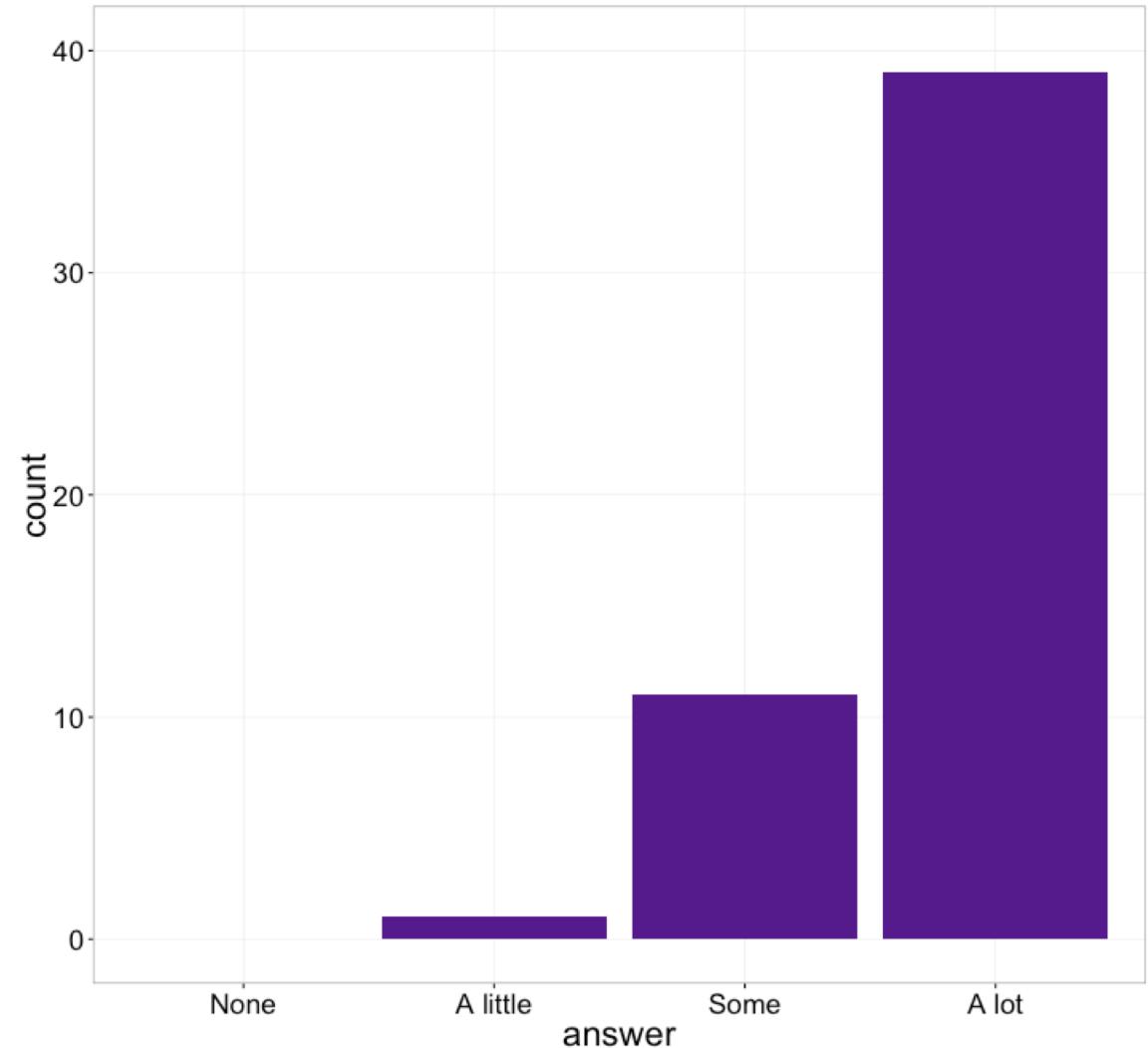
# Keep all categories, manually set axes

```
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?

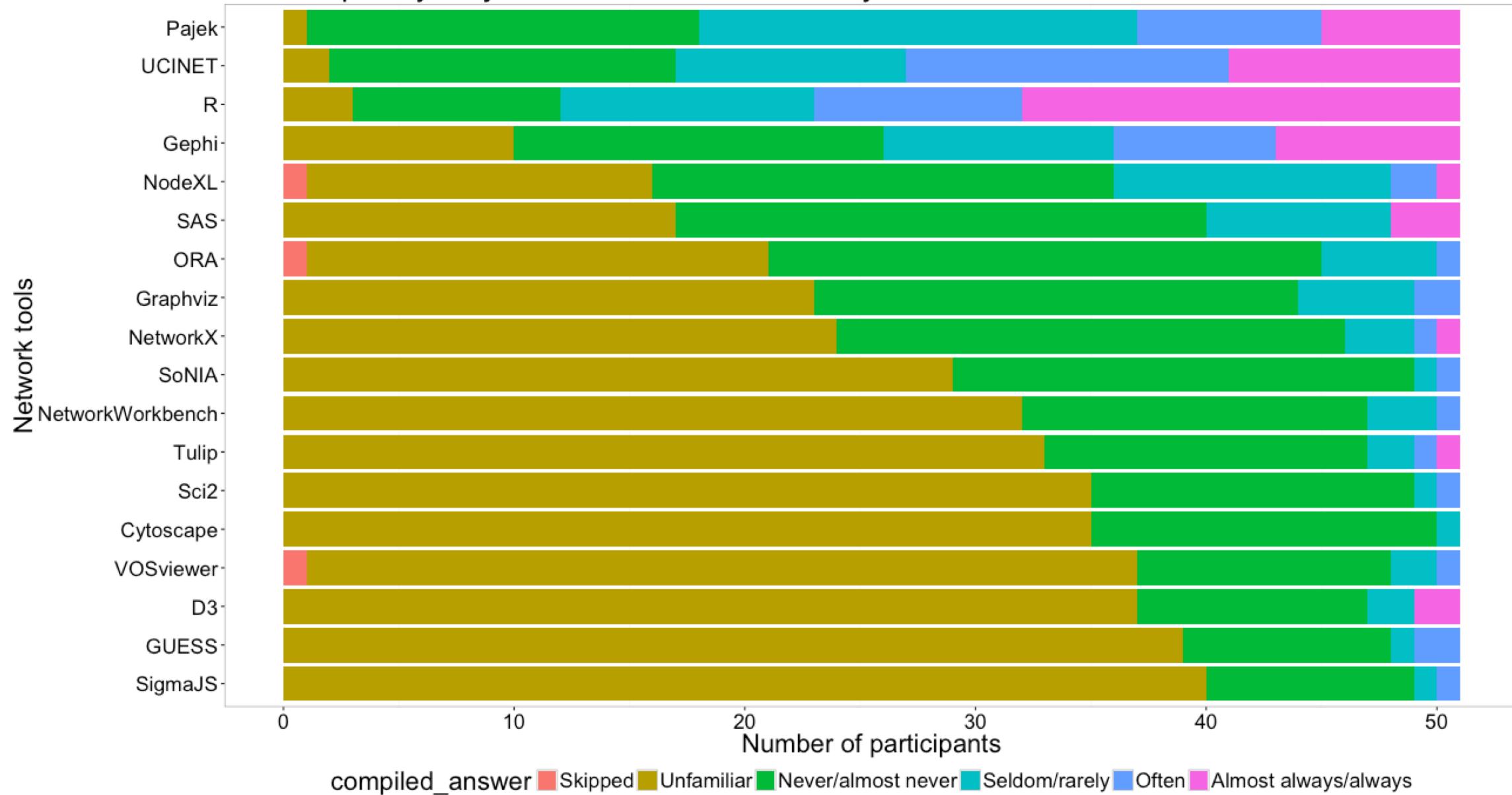


How much experience do you have as a consumer of network science research?



Principle 5:  
Select meaningful colors

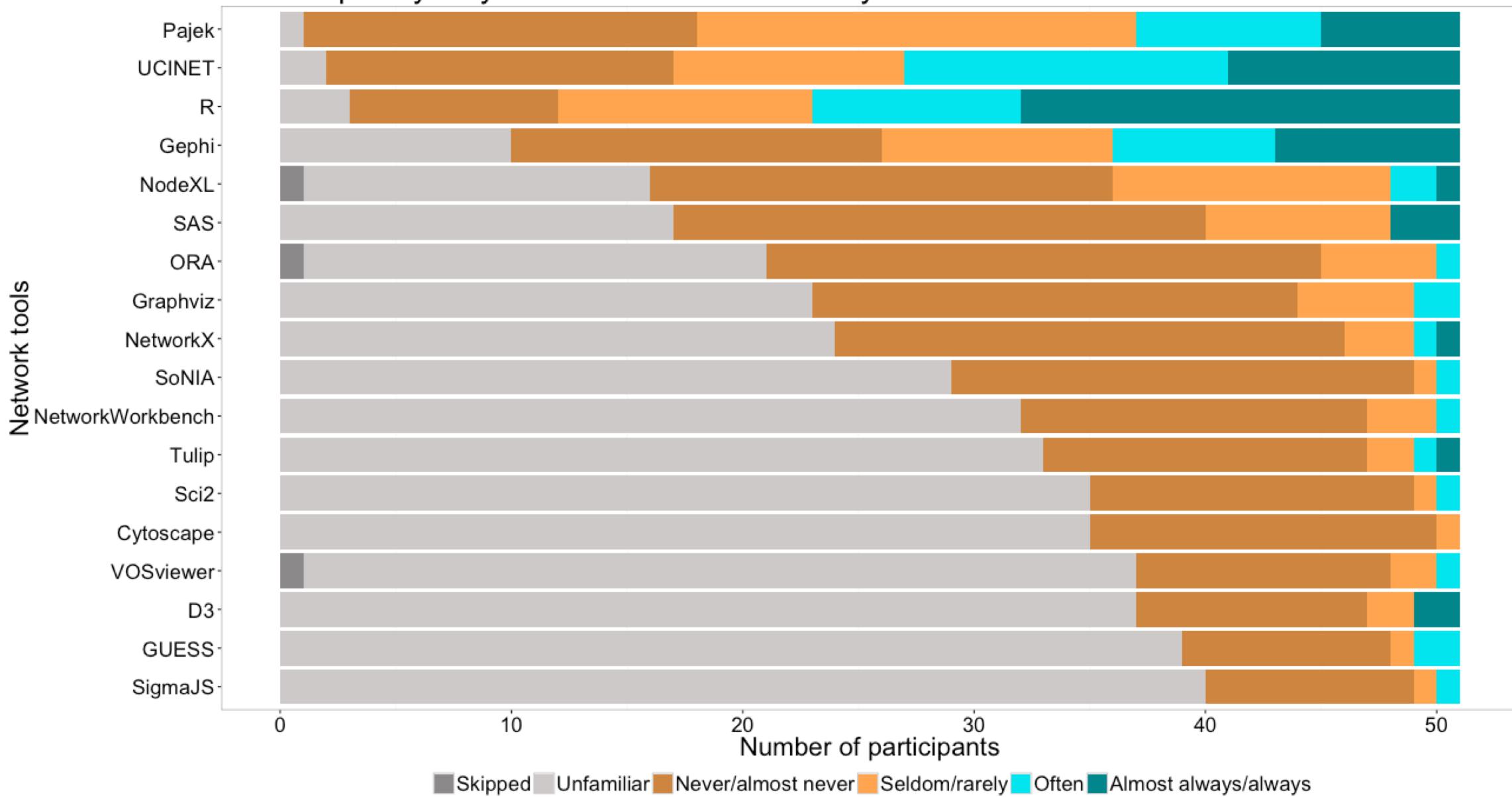
## How frequently do you use these tools for analysis?



# 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?



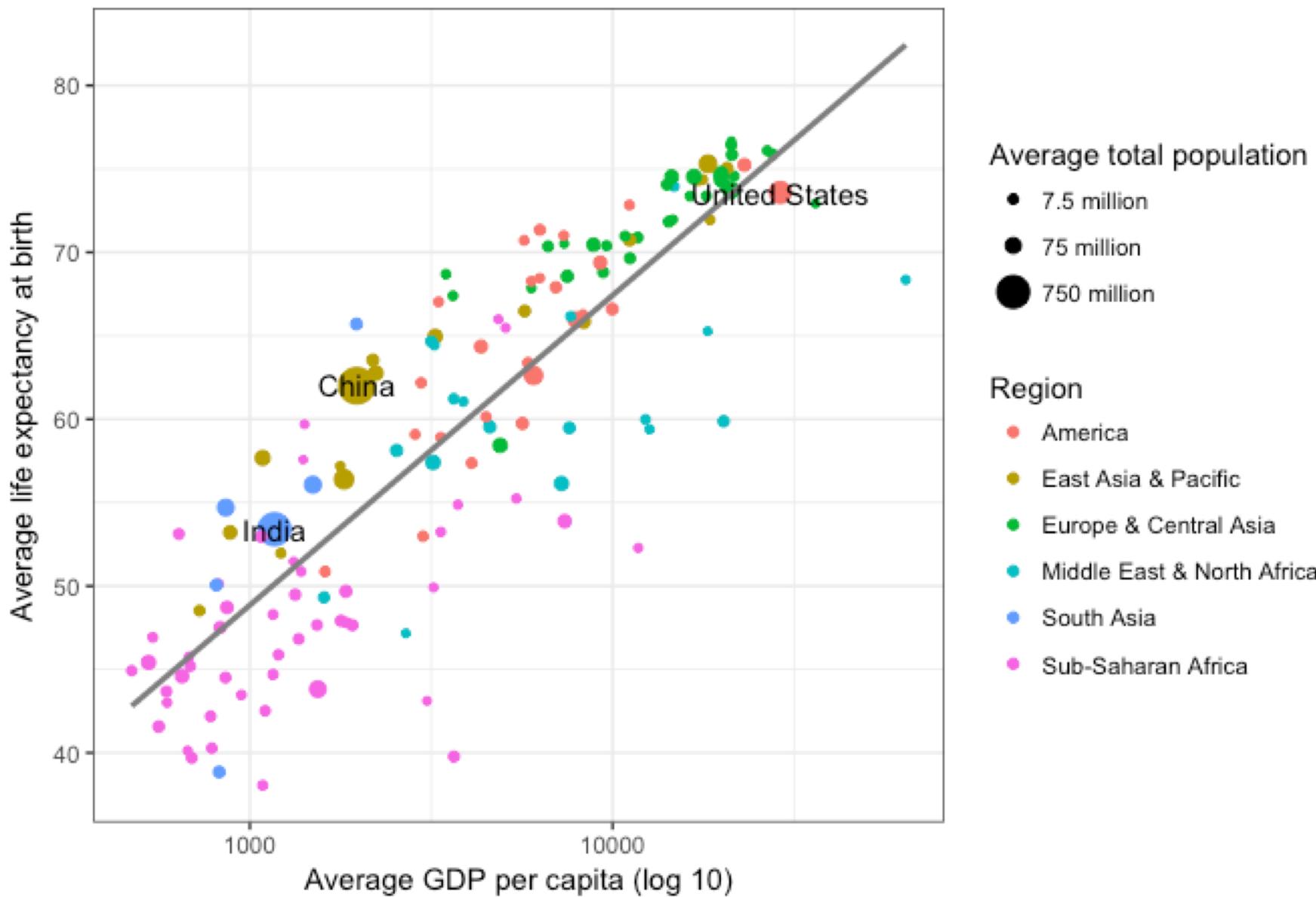
# Star Wars opinion survey

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

# Gapminder Data

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

Averages across all years of the traditional Gapminder dataset



Saving charts out

# 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's **aesthetics** like **size**, **color**, and **x and y locations**.

Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(aes(x = <cty>, y = <hwy>))  
  stat = <STAT>, position = <POSITION> +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTIONS> +  
  <SCALE_FUNCTIONS> +  
  <THEME_FUNCTIONS>
```

ggplot(data = mpg, aes(x = cty, y = hwy))  
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 <- ggplot(economics, aes(date, unemployed))  
b <- ggplot(seals, aes(x = long, y = lat))  
  
a + geom_blank()  
a + geom_label(aes(label = cty), nudge_x = 1, nudge_y = -1, check_overlap = TRUE)  
b + geom_curve(aes(end = lat + 1, xend = long + 1, curvature = z)) -> x, yend, y, yend, alpha, angle, color, curvature, linetype, size, linemetre=1)  
a + geom_path(linend = "butt", linejoin = "round", x, y, alpha, color, group, linetype, size, stroke)  
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, xmin, ymin, alpha, color, fill, group, linetype, size  
a + geom_ribbon(aes(ymin = unemployed - 900, ymax = unemployed + 900)) -> x, ymin, ymax, 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 = cty), nudge_x = 1, nudge_y = -1, check_overlap = TRUE)
```

#### TWO VARIABLES

```
continuous x, continuous y  
e <- ggplot(mpg, aes(cty, hwy))  
b + geom_abline(aes(intercept = 0, slope = 1))  
b + geom_hline(aes(yintercept = lat))  
b + geom_vline(aes(xintercept = long))  
b + geom_segment(aes(xend = lat + 1, yend = long + 1))  
b + geom_spline(aes(angle = 1:115, radius = 1))
```

#### continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))  
h + geom_bin2d(binwidth = c(0.25, 500))  
h + geom_density2d()  
h + geom_hex()
```

#### continuous function

```
I <- ggplot(economics, aes(date, unemployed))  
I + geom_area()  
I + geom_line()  
I + geom_step(direction = "hv")
```

#### visualizing error

```
d <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)  
j <- ggplot(d, aes(grp, fit, ymin = fit - se, ymax = fit + se))  
j + geom_crossbar(fatten = 2)  
j + geom_errorbar()  
j + geom_errorbar(l, x, y, max, ymin, alpha, color, group, linetype, size, width)  
j + geom_linerange()  
j + geom_pointrange()  
j + geom_violin(scale = "area")
```

#### 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 = mapLong, y = mapLat), map_id, alpha, color, fill, linetype, size
```

### Three Variables

```
seals$z <- with(seals, sqrt(data.long^2 + data.lat^2))  
l <- ggplot(seals, aes(long, lat))  
l + geom_contour(aes(z = z))  
x, z, alpha, colour, group, linetype, size, weight  
d + geom_bar()  
d + geom_bar()
```

### R Studio

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>

# 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 video frame showing two people in a room, one standing at a whiteboard. Below this thumbnail is a search bar labeled "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 main content area is a playback control bar with a play button, a timestamp of "0:03", a circular progress bar, a timestamp of "1:22:45", a speed control set to "1X", a quality control, and a "Hide" button. Below the control bar are four small thumbnail images representing different parts of the presentation, with timestamps 1:32, 4:32, 7:32, and 10:32.

<http://bit.ly/DVsvideos>

# Questions?

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

# Star Wars character data

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