Visualization in R using ggplot2

Angela Zoss

9/18/19

https://github.com/amzoss/ggplot2-F18

Set up environment

- R
- RStudio
- tidyverse package

Get workshop files

URL: https://github.com/amzoss/ggplot2-F18

With Git installed

In RStudio:

- Project → New project
- Version Control
- Git
  - Paste in GitHub URL
  - Project directory name: ggplot2-F18
  - 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

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!

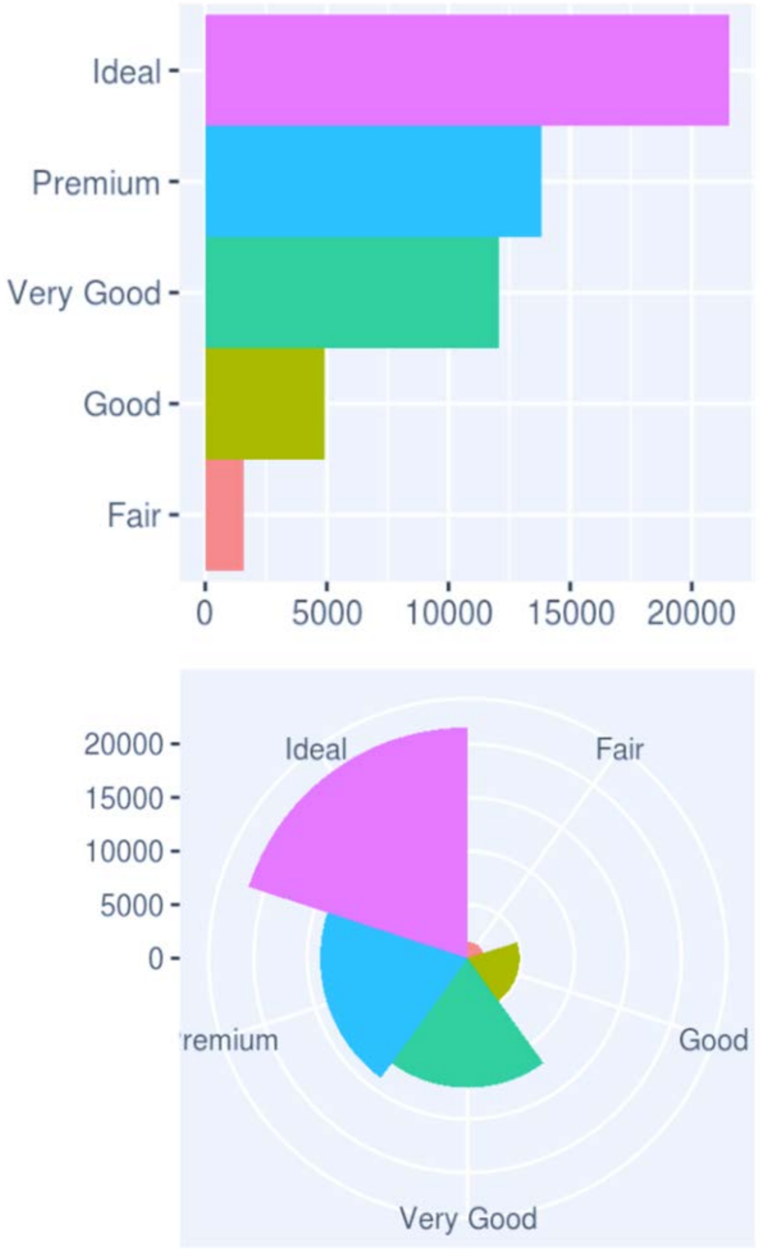
“Your closest collaborator is ***you*** *six months ago, but you don’t reply to emails.”*

- 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

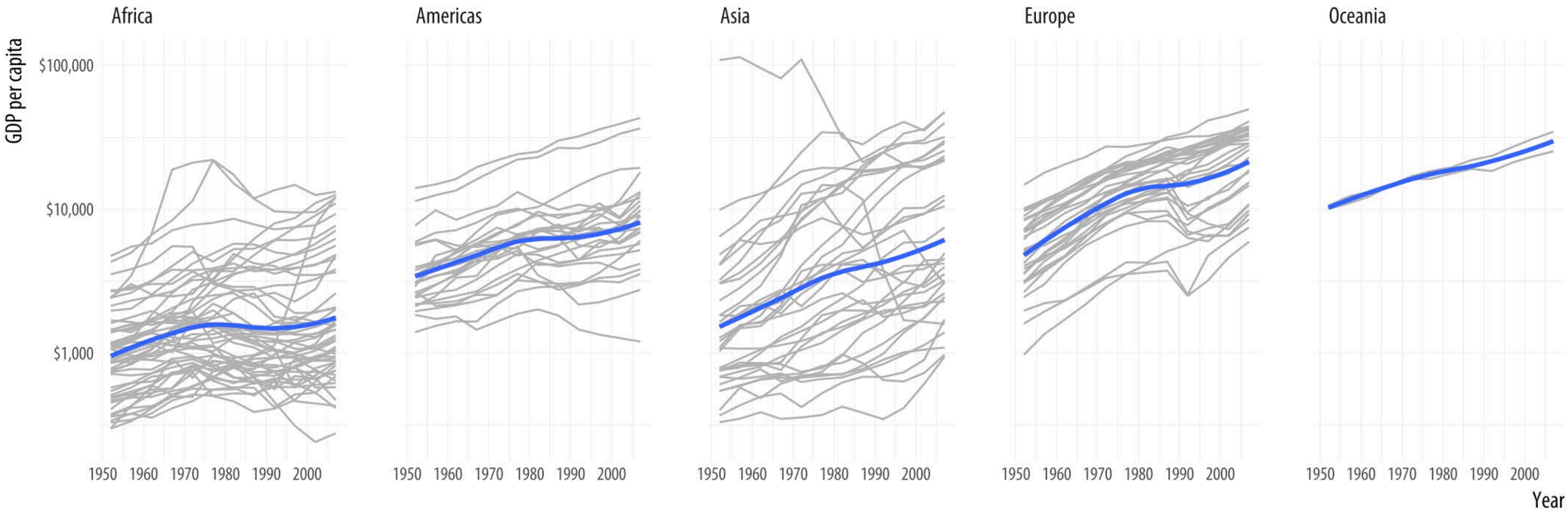
Grammar of graphics

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.).

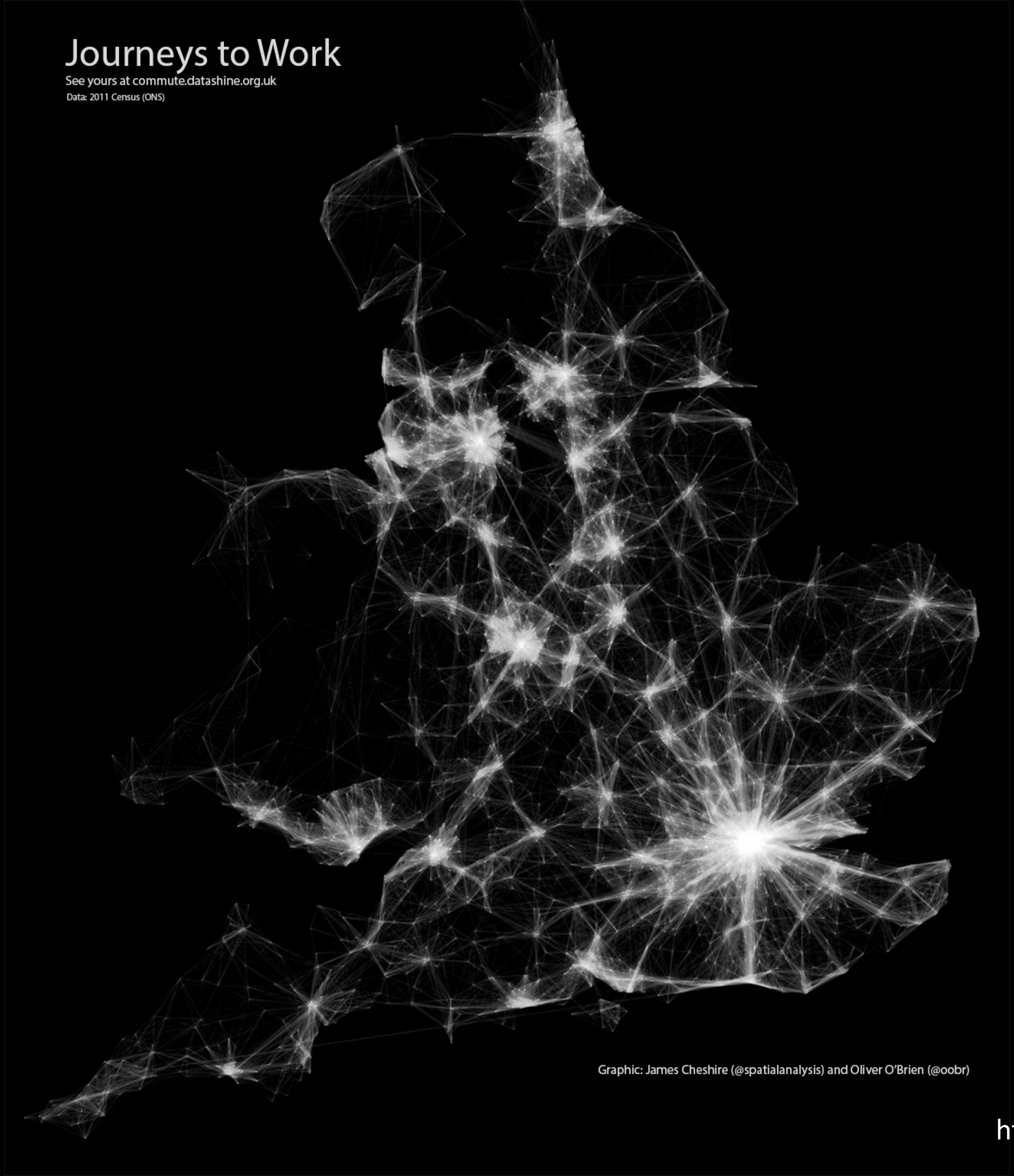
Wilkinson, Leland. (2005). *The grammar of graphics (2nd ed)*. New York: Springer.

ggplot2 examples

GDP per capita on Five Continents



http://socviz.co/groupfacettx.html



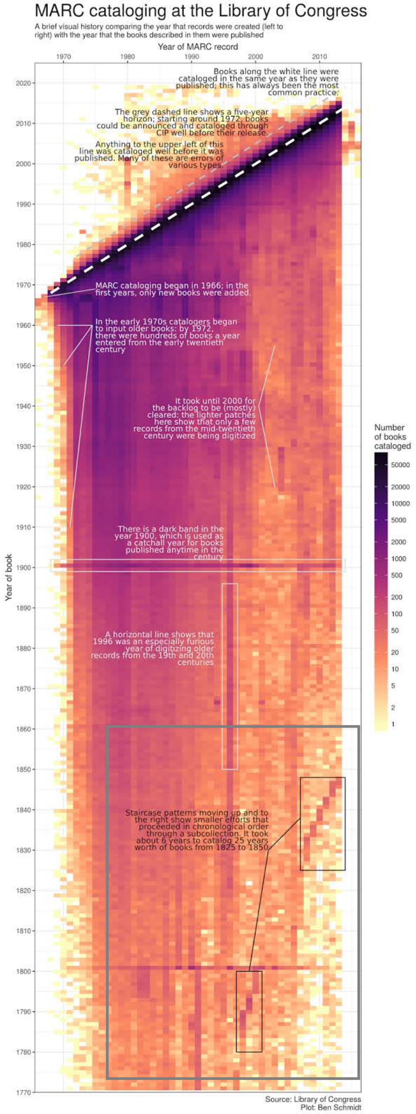
Journeys to Work

See yours at commute.datashine.org.uk

Data: 2011 Census (ONS)

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

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



MARC cataloging at the Library of Congress

A brief visual history comparing the year that records were created (left to right) with the year that the books described in them were published

Books along the white line were cataloged in the same year as they were published; this has always been the most common practice.

The grey dashed line shows a five-year horizon; starting around 1972, books could be announced and cataloged through CIP well before their release.

Anything to the upper left of this line was cataloged well before it was published. Many of these are errors of various types.

MARC cataloging began in 1966; in the first years, only new books were added.

In the early 1970s catalogers began to input older books: by 1972, there were hundreds of books a year entered from the early twentieth century

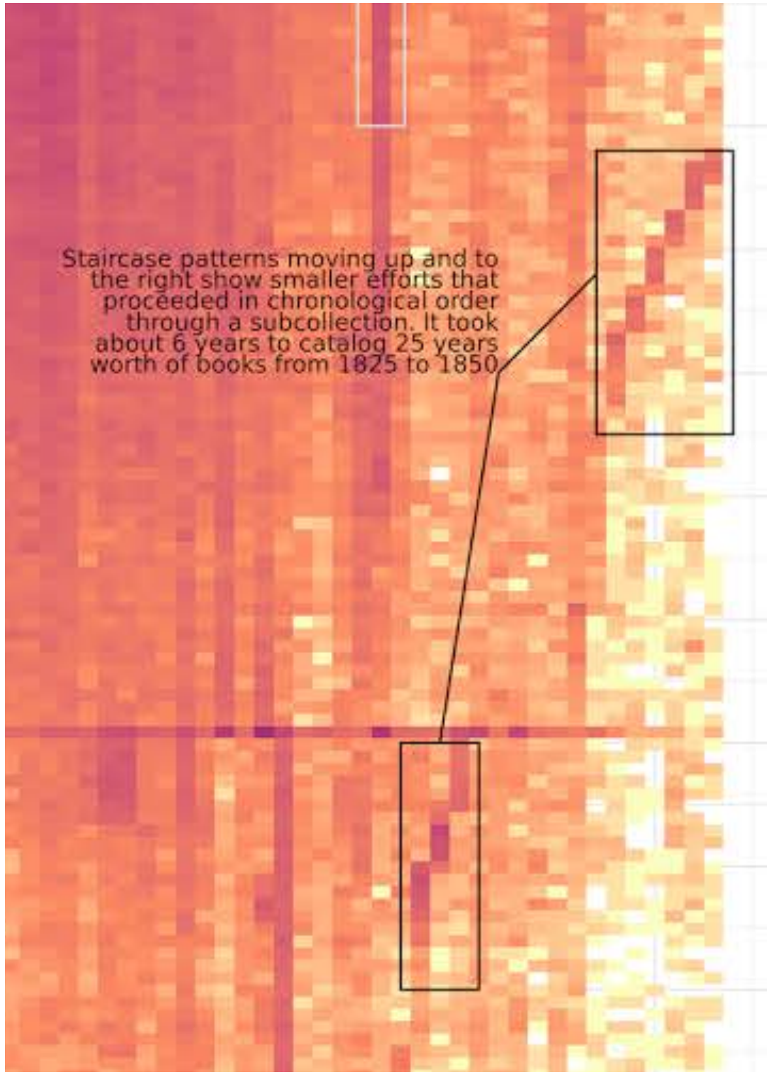
It took until 2000 for the backlog to be (mostly) cleared: the lighter patches here show that only a few records from the mid-twentieth century were being digitized

There is a dark band in the year 1900, which is used as a catchall year for books published anytime in the century

A horizontal line shows that 1996 was an especially furious year of digitizing older records from the 19th and 20th centuries

Staircase patterns moving up and to the right show smaller efforts that proceeded in chronological order through a subcollection. It took about 6 years to catalog 25 years worth of books from 1825 to 1850

Source: Library of Congress
Plot: Ben Schmidt



Staircase patterns moving up and to the right show smaller efforts that proceeded in chronological order through a subcollection. It took about 6 years to catalog 25 years worth of books from 1825 to 1850

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

Why ggplot2 instead of base R?

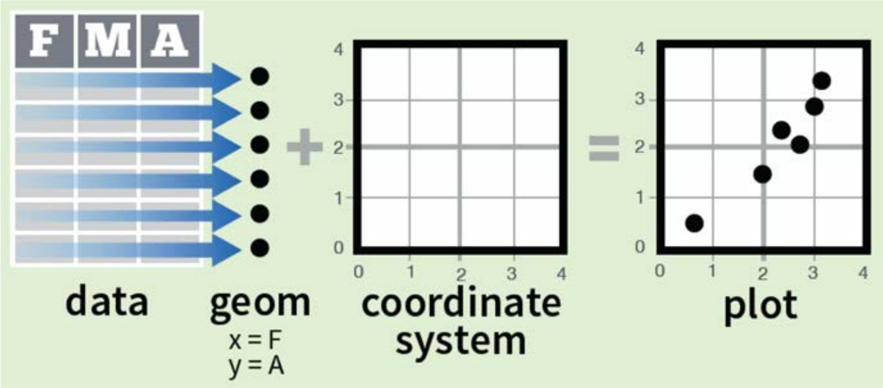
- nice defaults
- easy faceting
- (arguably) more natural syntax
- can switch chart types more easily

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

ggplot2: Elements

Basic elements in any ggplot2 visualization

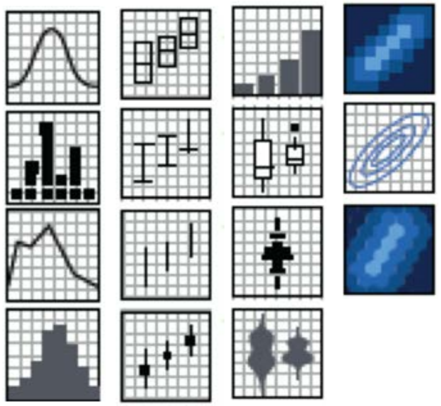
- **data**
- **aes**thetics
  (variable mappings)
- **geom**
  (chart type or shape)
- coordinate system
  (the arrangement of the marks;
  most geoms use default, cartesian)



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

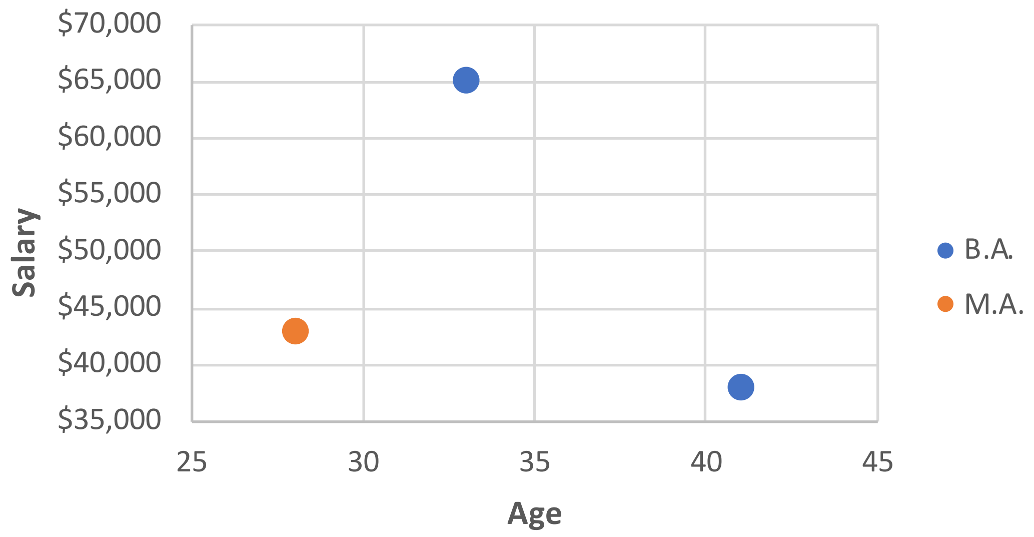
Template for a simple plot

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

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

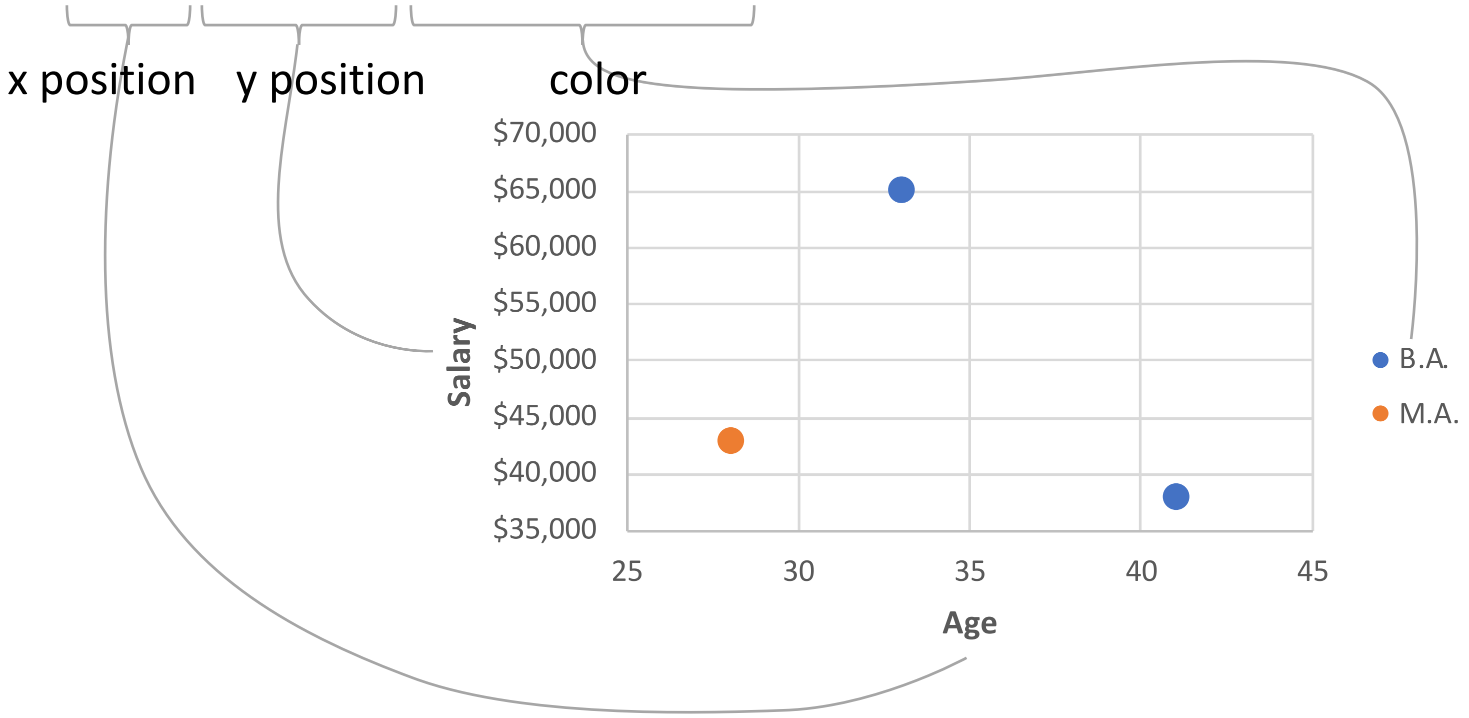
Aesthetic variable mappings

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.



x position

y position

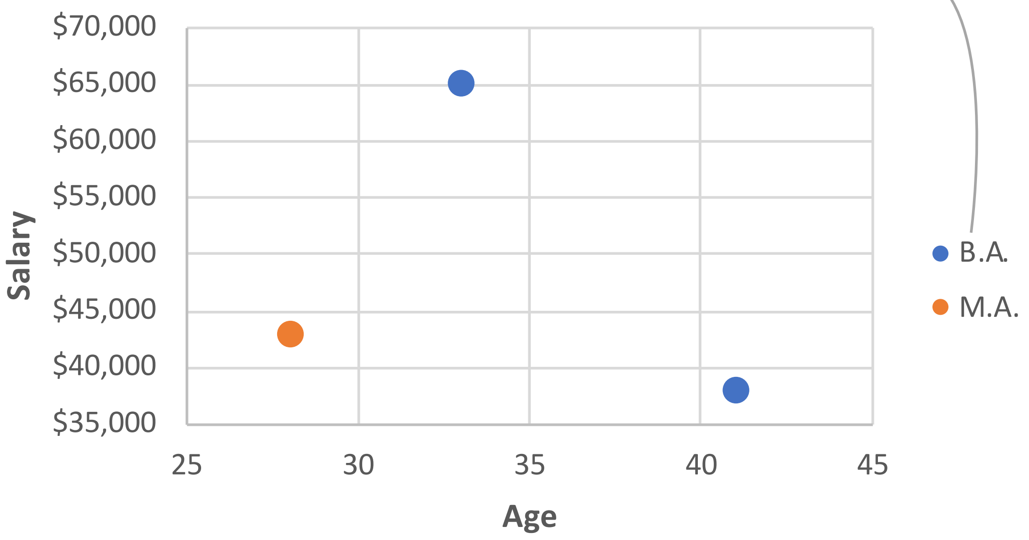
color

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))
```

x position   y position   color

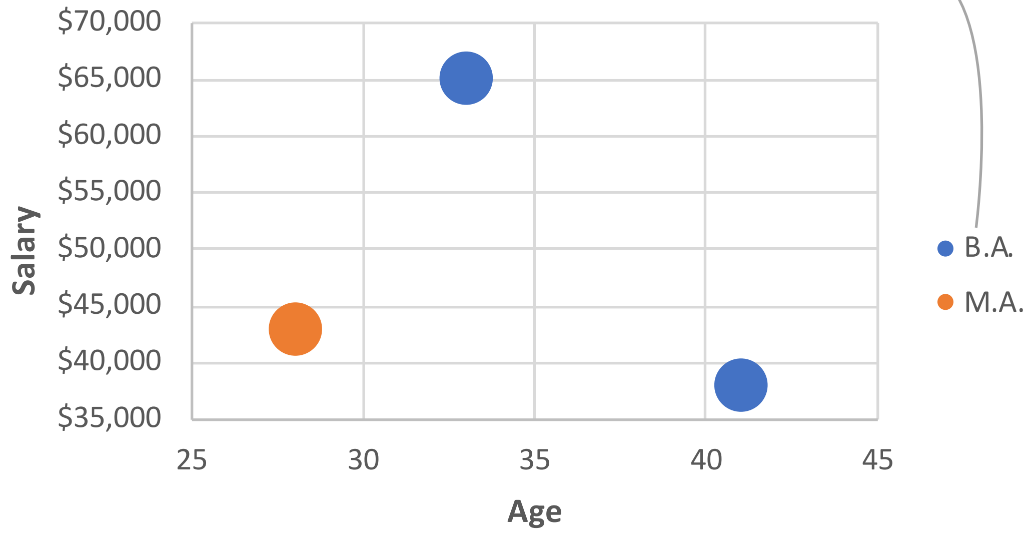


Non-variable adjustments

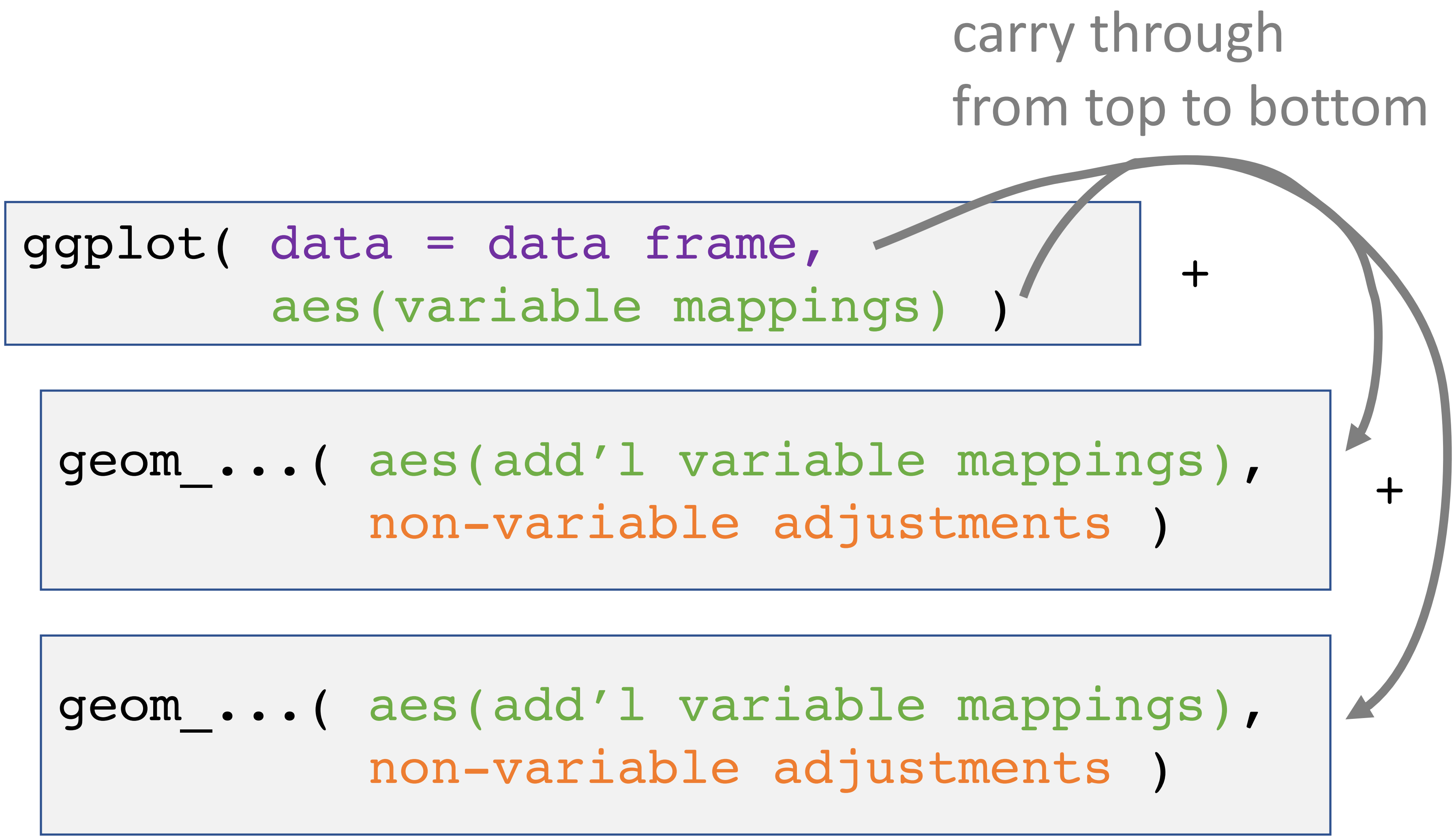
Name	Age	Salary	Highest Degree
Jane Smith	33	$65,000	B.A.
Abby Jones	28	$43,000	M.A.
Bridget Carden	41	$38,000	B.A.

x position y position color

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



Template for a more complex plot



carry through
from top to bottom

```
ggplot( data = data frame,
        aes(variable mappings) )
```

+

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

+

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

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.

Dataset 1:
Game of Thrones character ratings

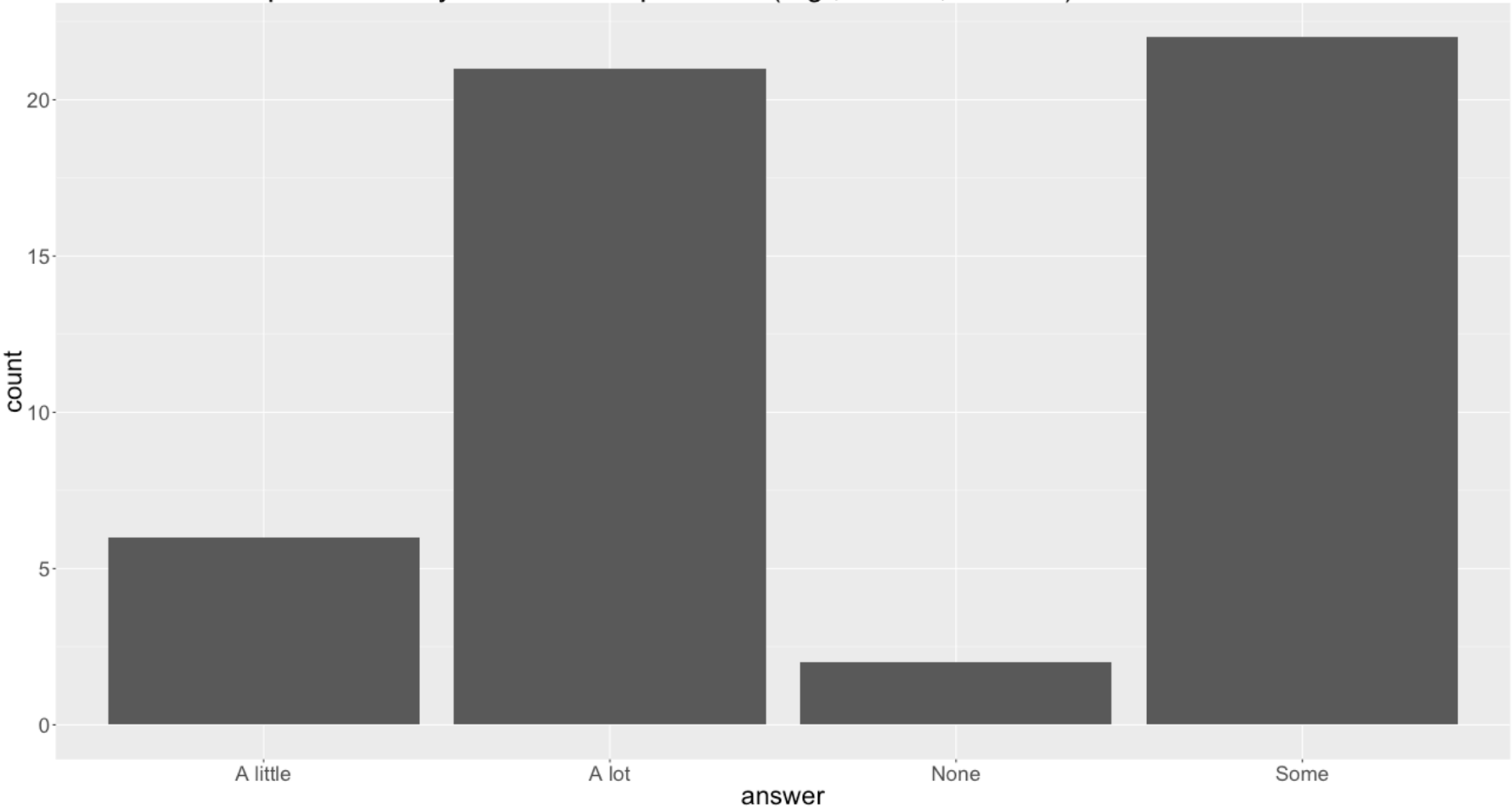
https://www.nytimes.com/interactive/2017/08/09/upshot/game-of-thrones-chart.html

Dataset 2:
Star Wars character data

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

Principles for Effective Visualizations

Principle 1: Order matters

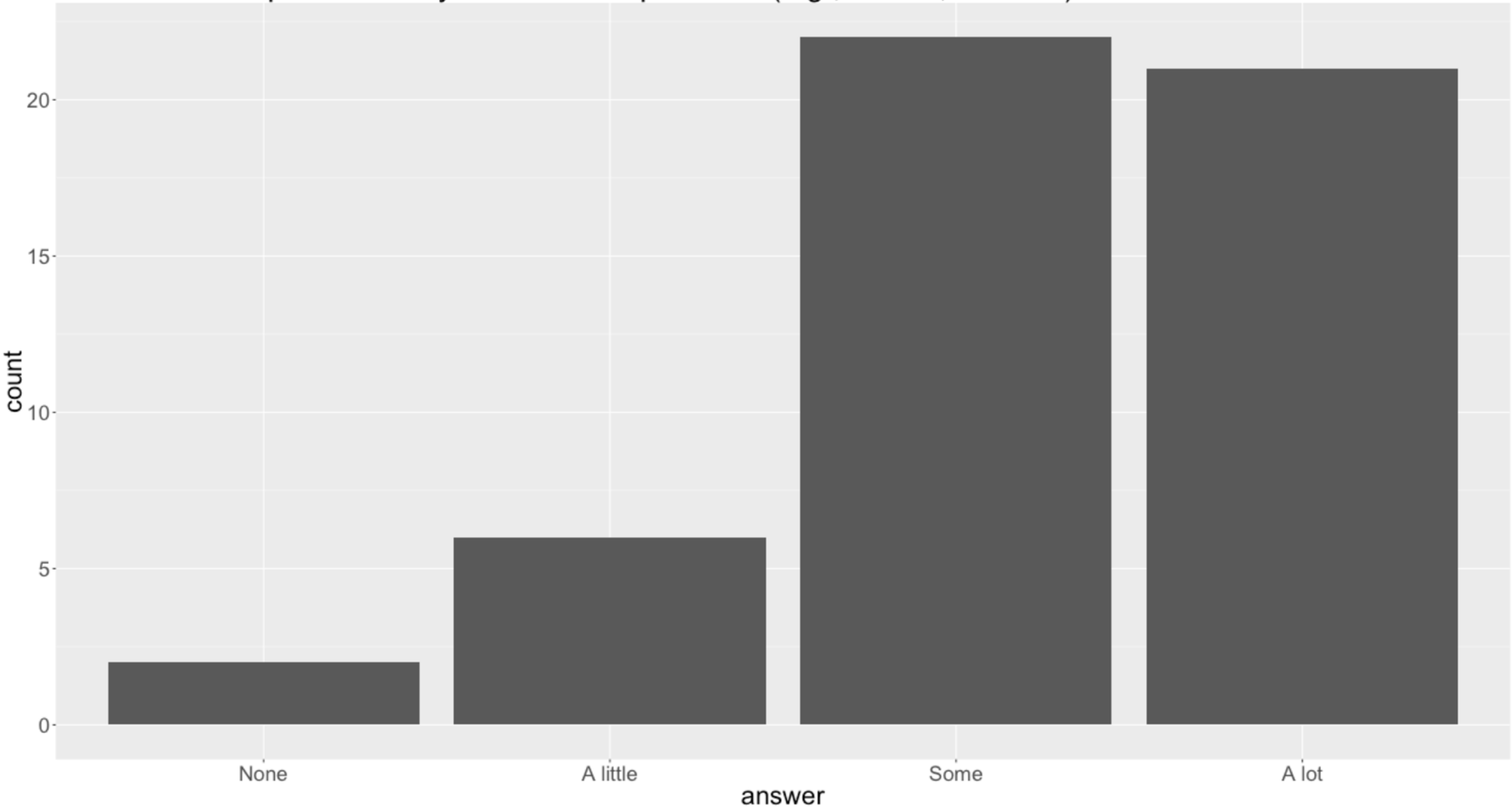


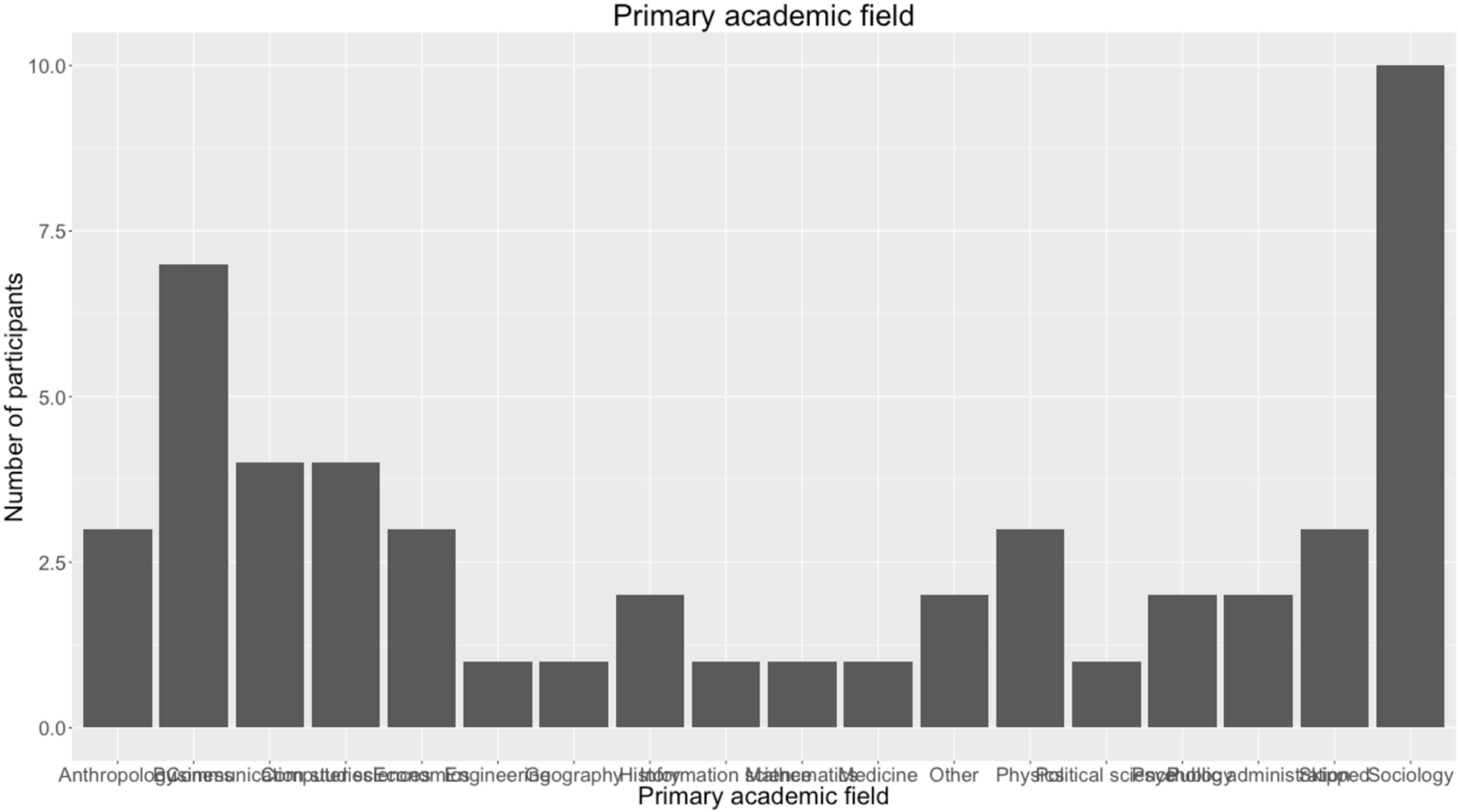
How much experience do you have as a producer (e.g., reader, follower) of network science research?

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?

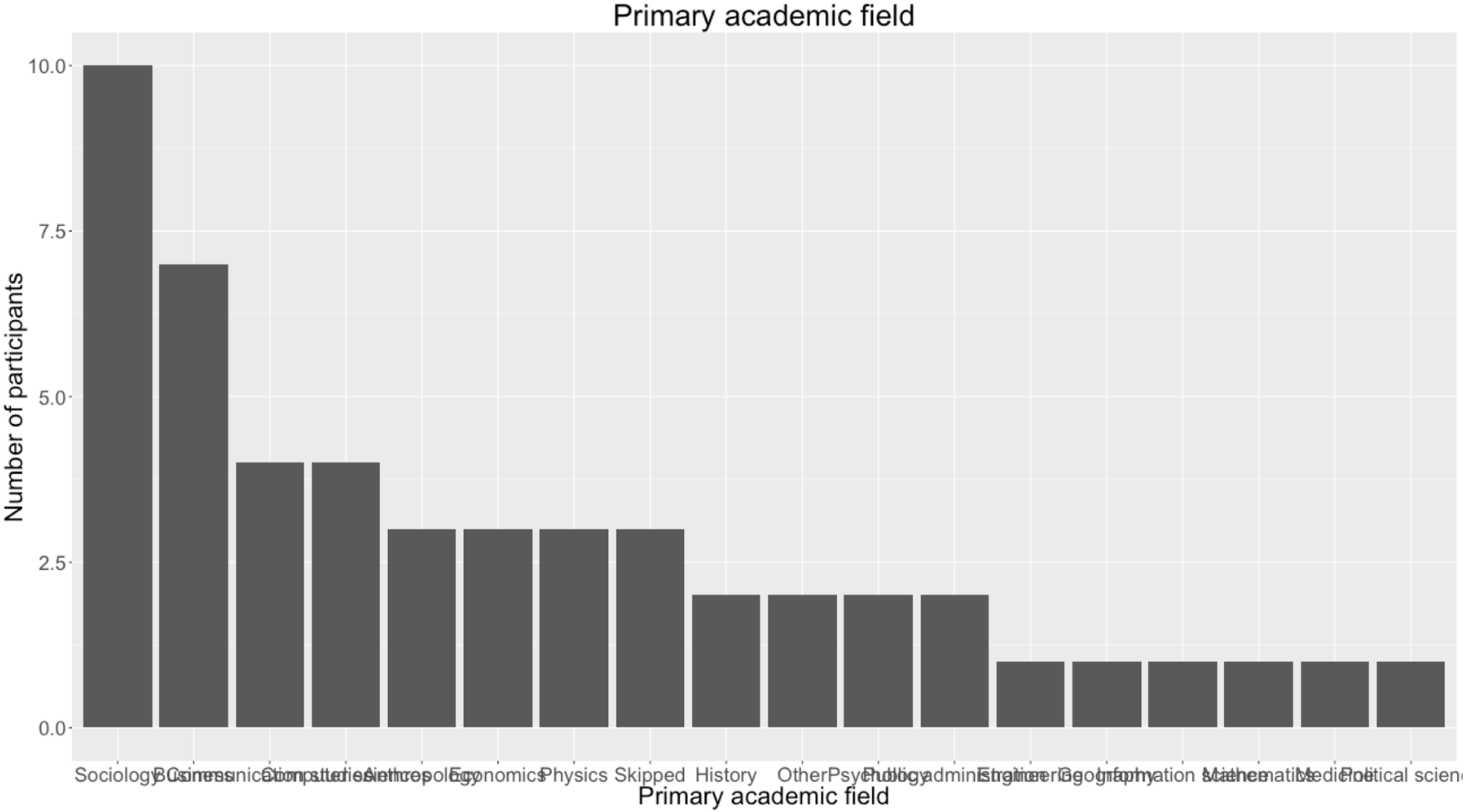



Primary academic field
Number of participants
10.0
7.5
5.0
2.5
0.0
Anthropology
Business
Communication studies
Computer science
Economics
Engineering
Geography
History
Information science
Mathematics
Medicine
Other
Physics
Political science
Psychology
Public administration
Skipped
Sociology
Primary academic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

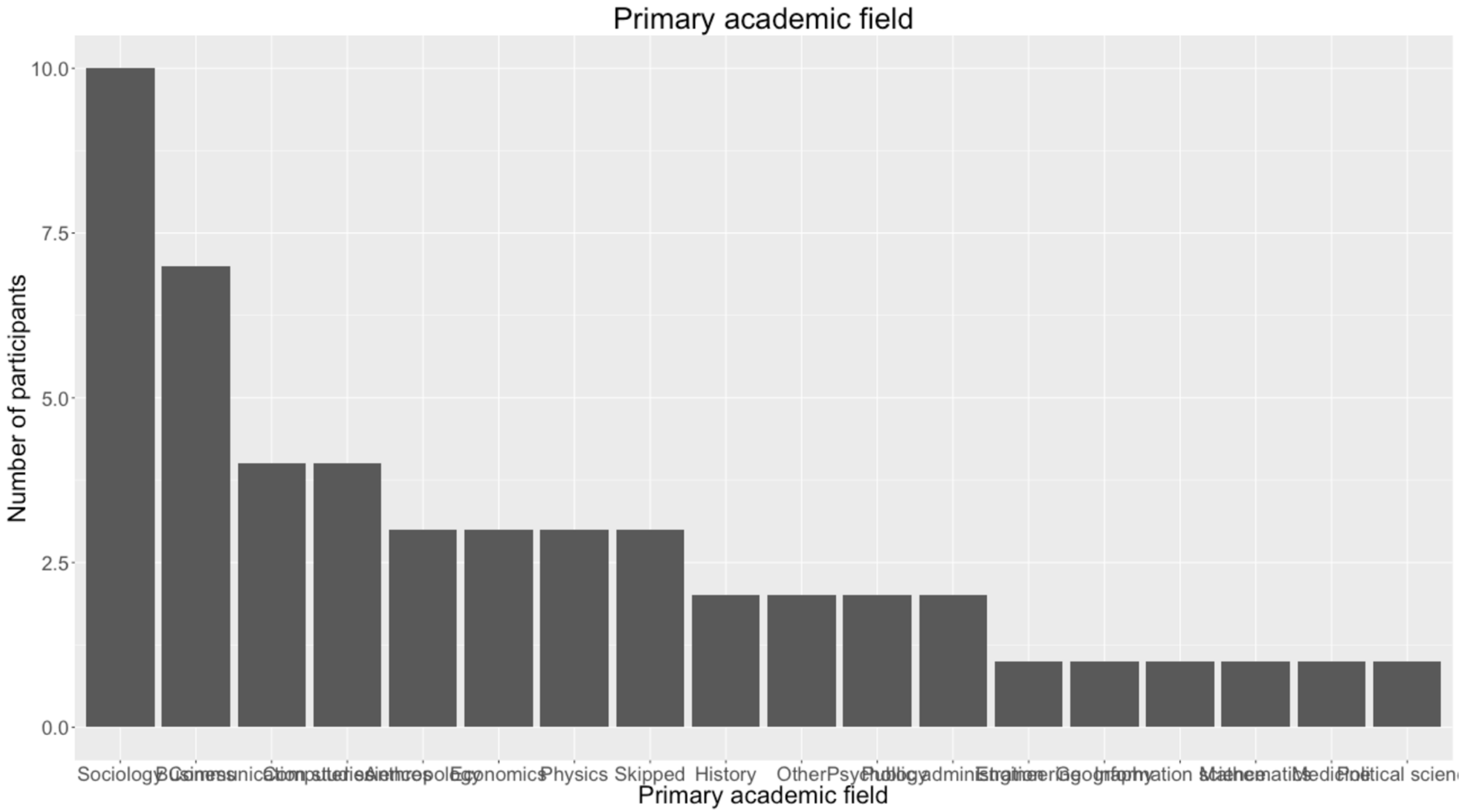
Order by value

```
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))
```

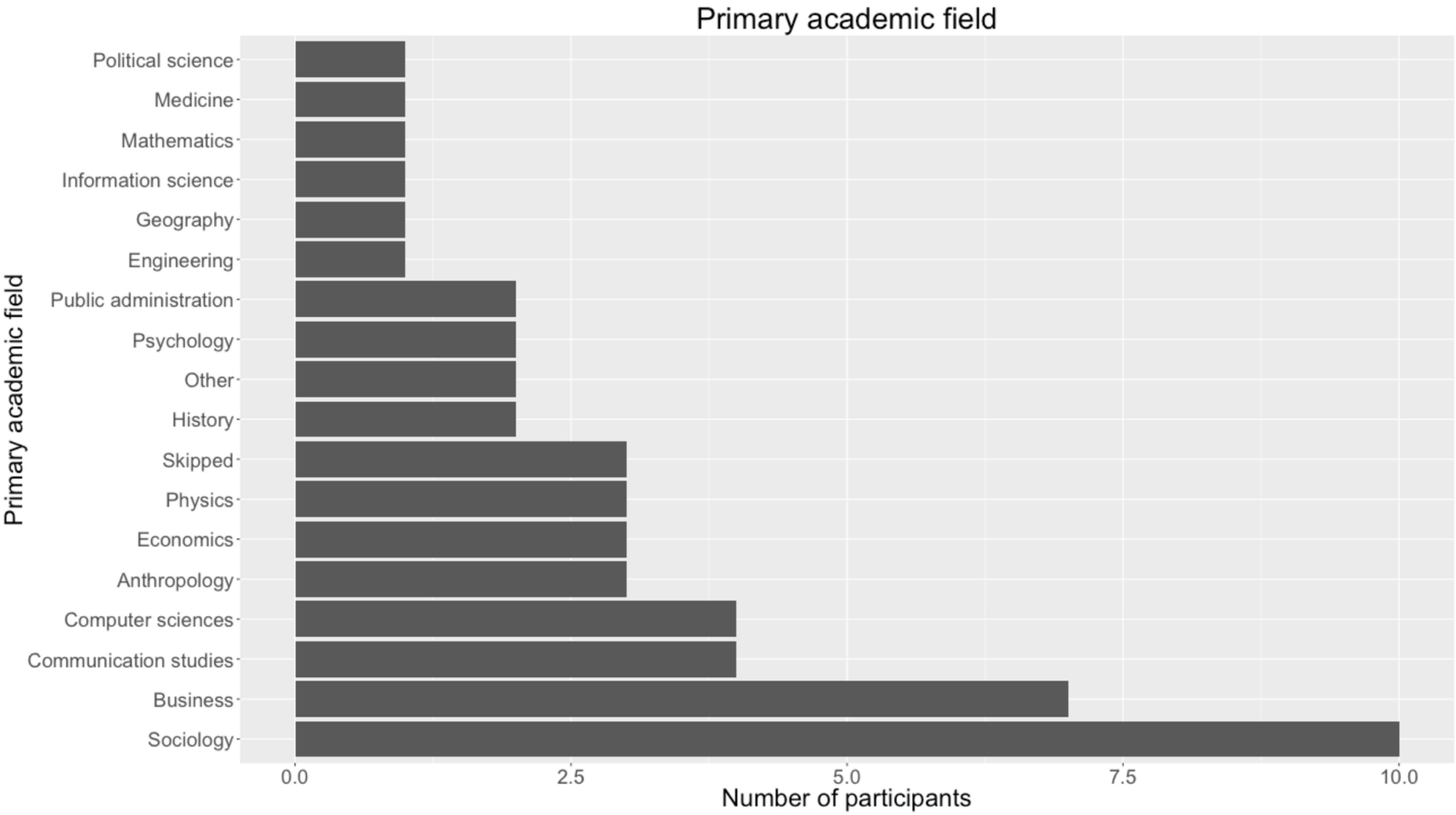

Primary academic field
Number of participants
10.0
7.5
5.0
2.5
0.0
Sociology
Business
Communication studies
Computer science
Anthropology
Economics
Physics
Skipped
History
Other
Psychology
Public administration
Engineering
Geography
Information science
Mathematics
Medicine
Political scien
Primary academic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

Principle 2:
Put long categories on y-axis


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

Flip the axes

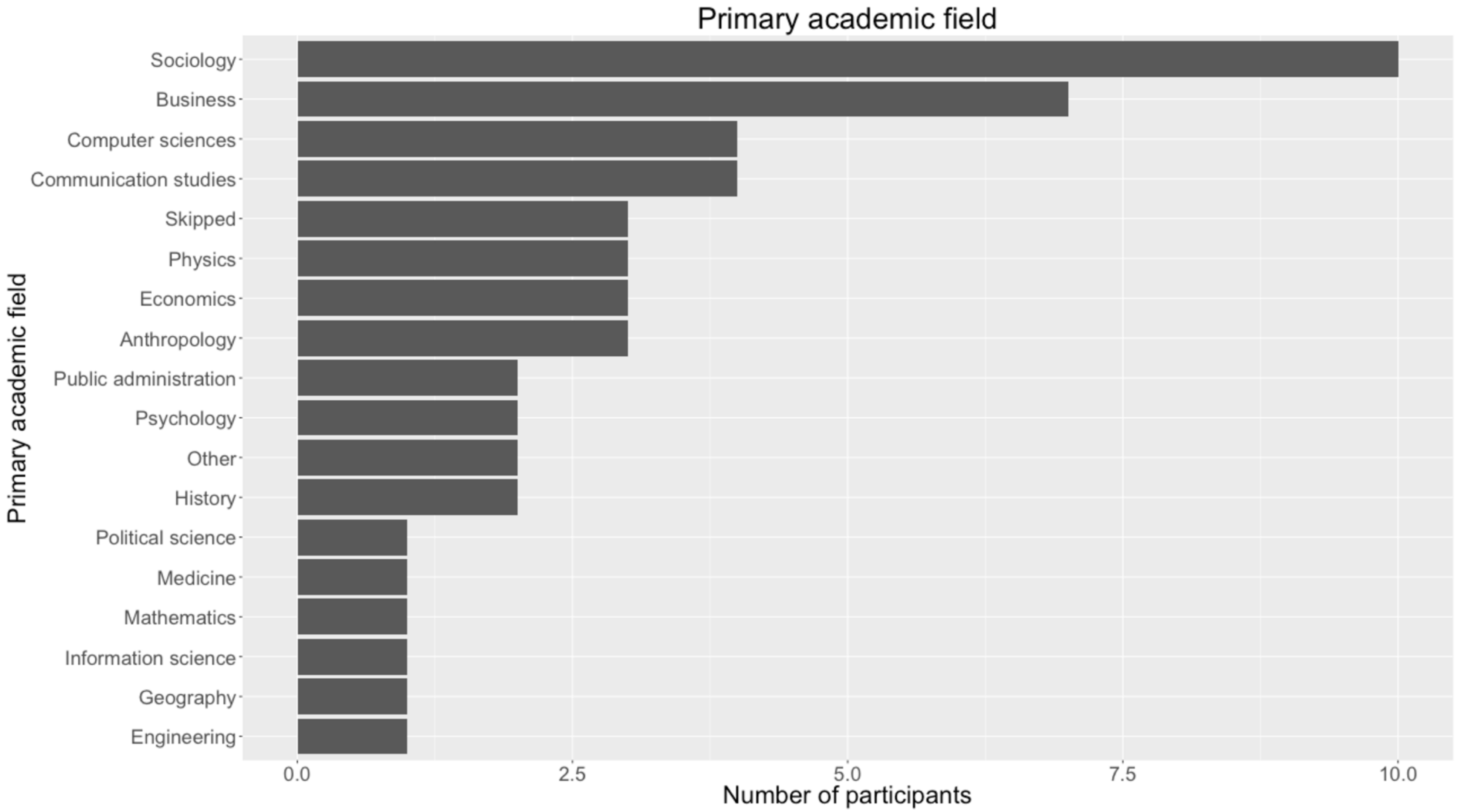
```
coord_flip()
```


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

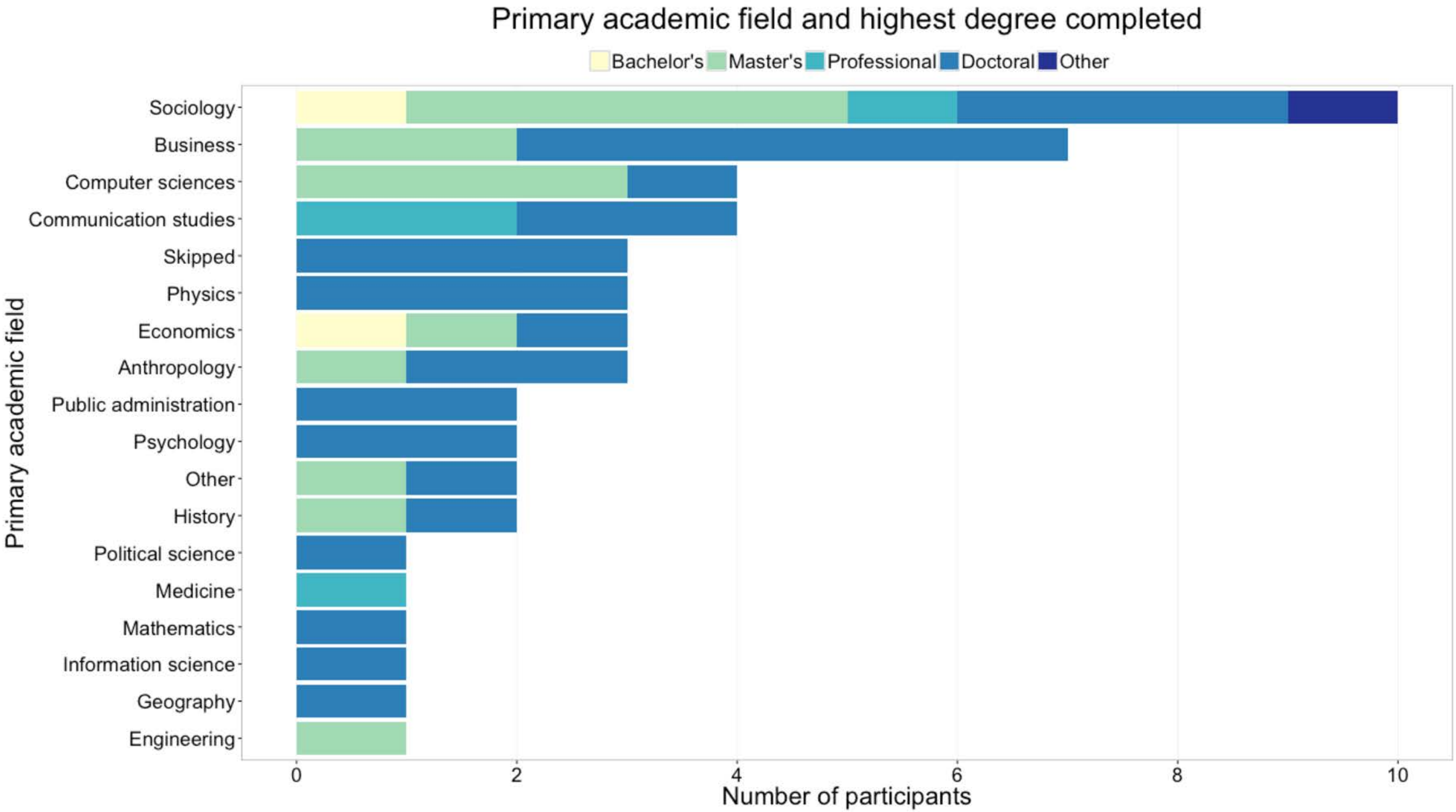
Oops!

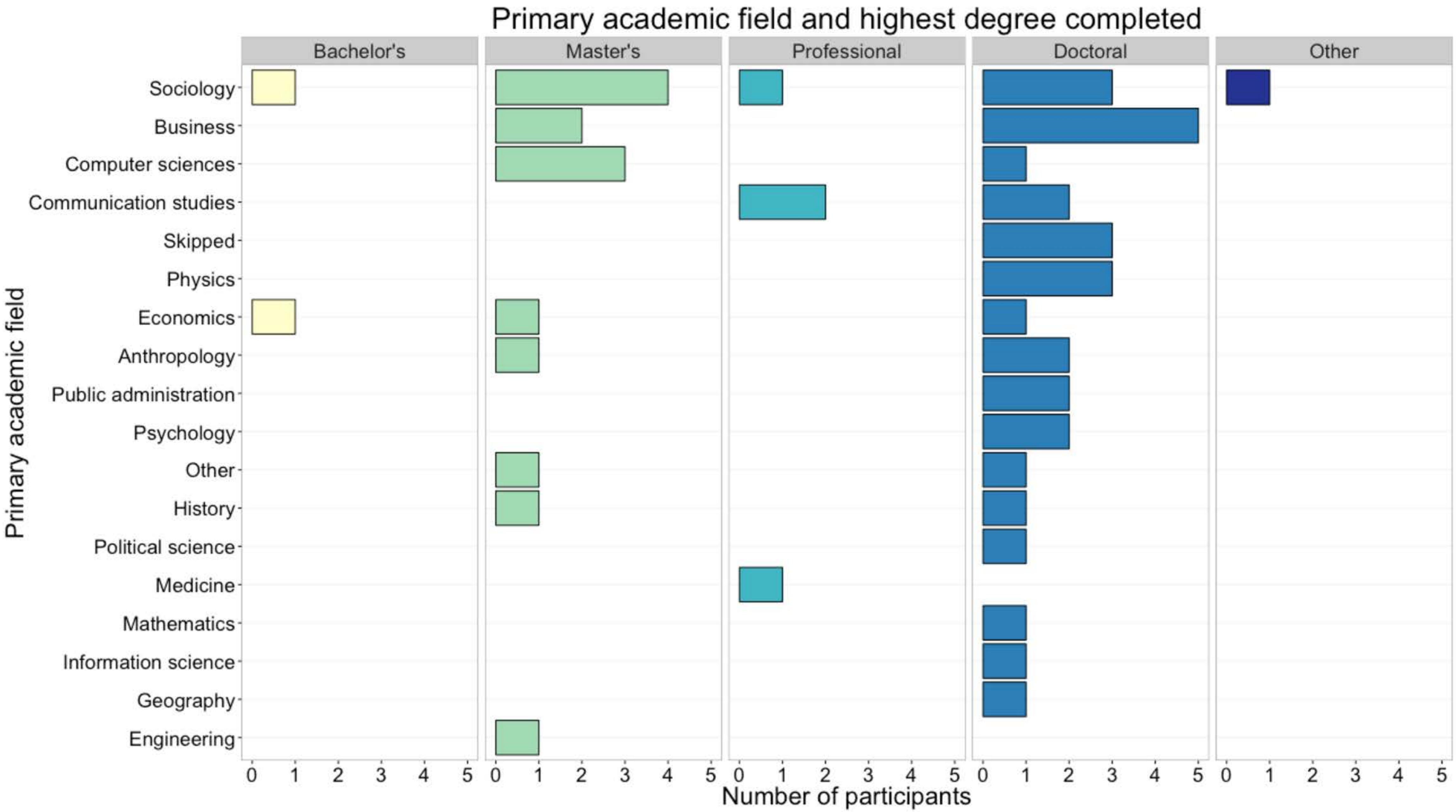
```
data$academic_field <-
     factor(data$academic_field,
           levels=names(
               sort(
                   table(data$academic_field),
                   ~~decreasing=TRUE~~)))
```

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


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

Principle 3: Pick a purpose


Primary academic field and highest degree completed
Bachelor's
Master's
Professional
Doctoral
Other
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
0
2
4
6
8
10
Number of participants

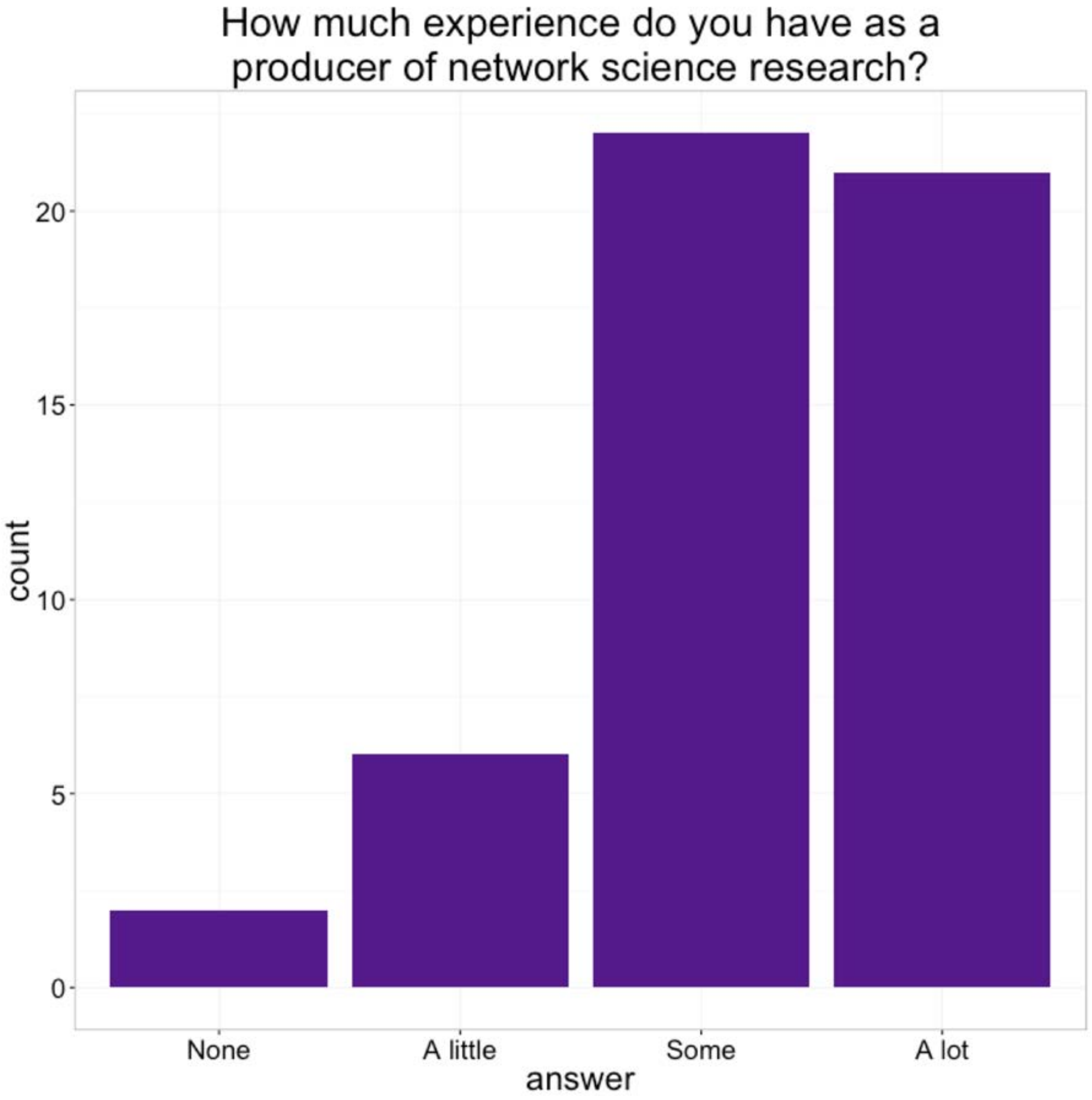

Primary academic field and highest degree completed
Bachelor's
Master's
Professional
Doctoral
Other
Primary academic field
Sociology
Business
Computer sciences
Communication studies
Skipped
Physics
Economics
Anthropology
Public administration
Psychology
Other
History
Political science
Medicine
Mathematics
Information science
Geography
Engineering
0
1
2
3
4
5
Number of participants

Different placement helps with different comparisons

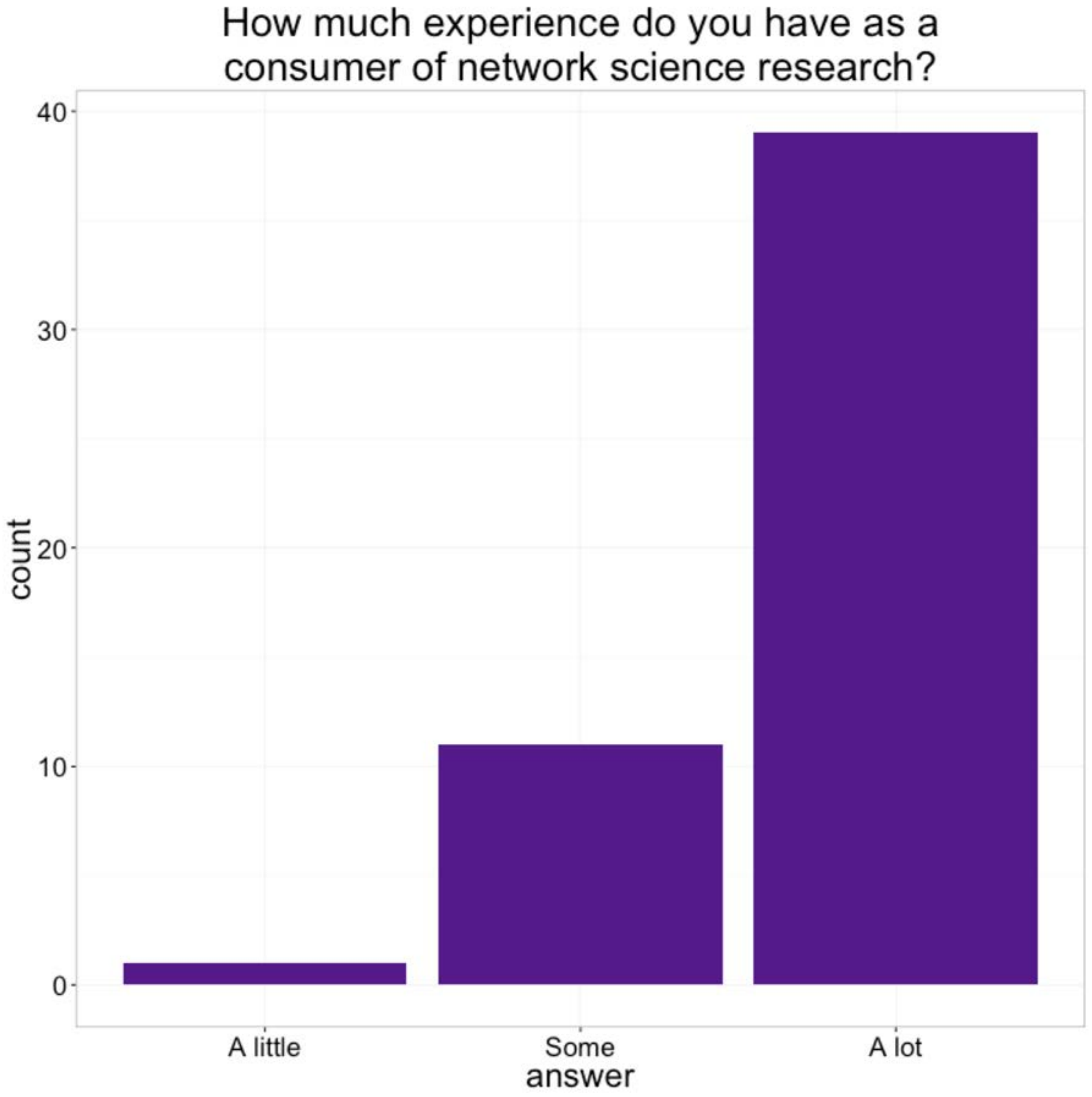
```
fill=highest_degree
```

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

Principle 4:
Keep scales consistent



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

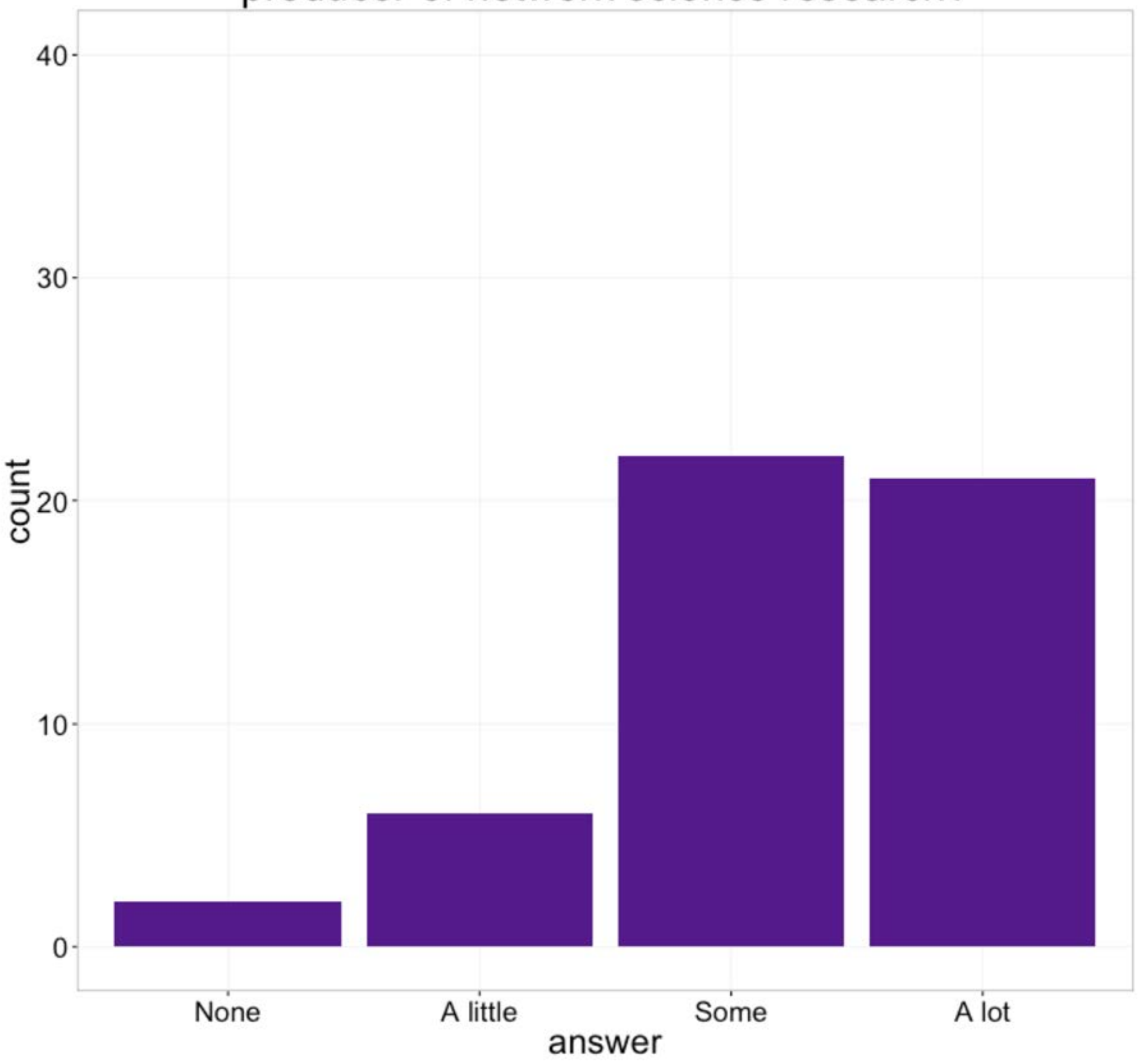


How much experience do you have as a consumer of network science research?

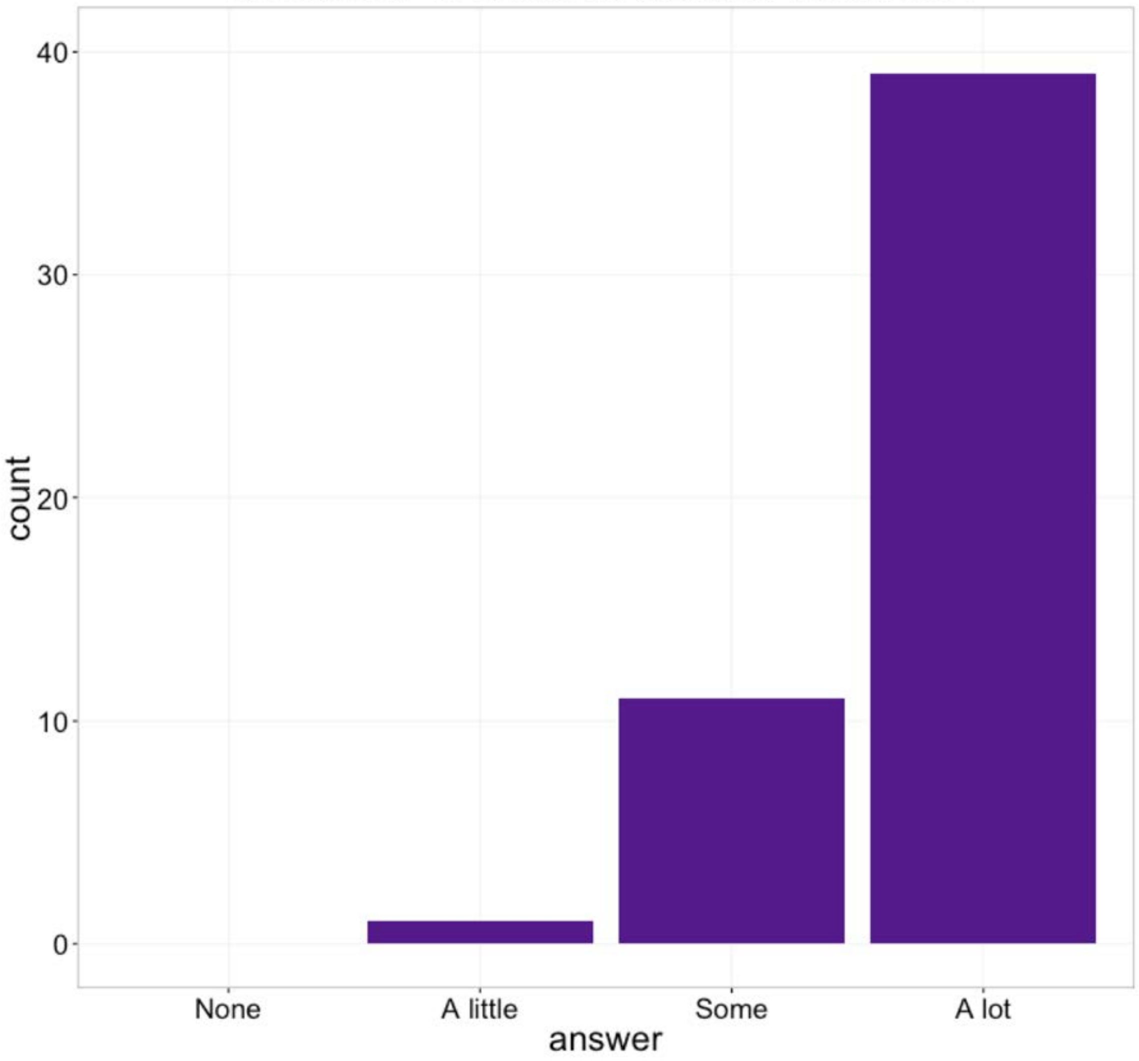
Keep all categories, manually set axes

```
scale_x_discrete(drop=FALSE)

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

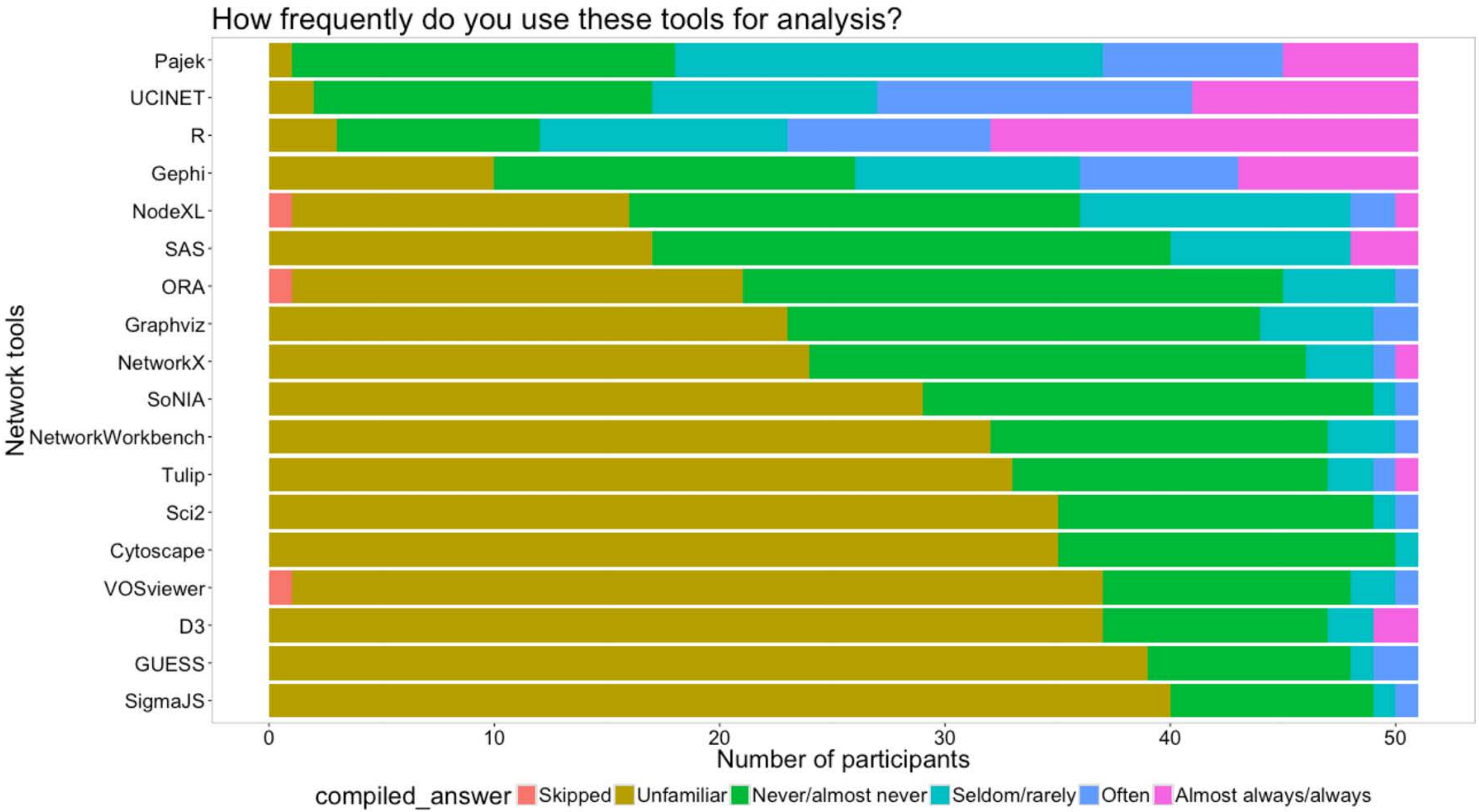


How much experience do you have as a producer of network science research?



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

Principle 5:
Select meaningful colors

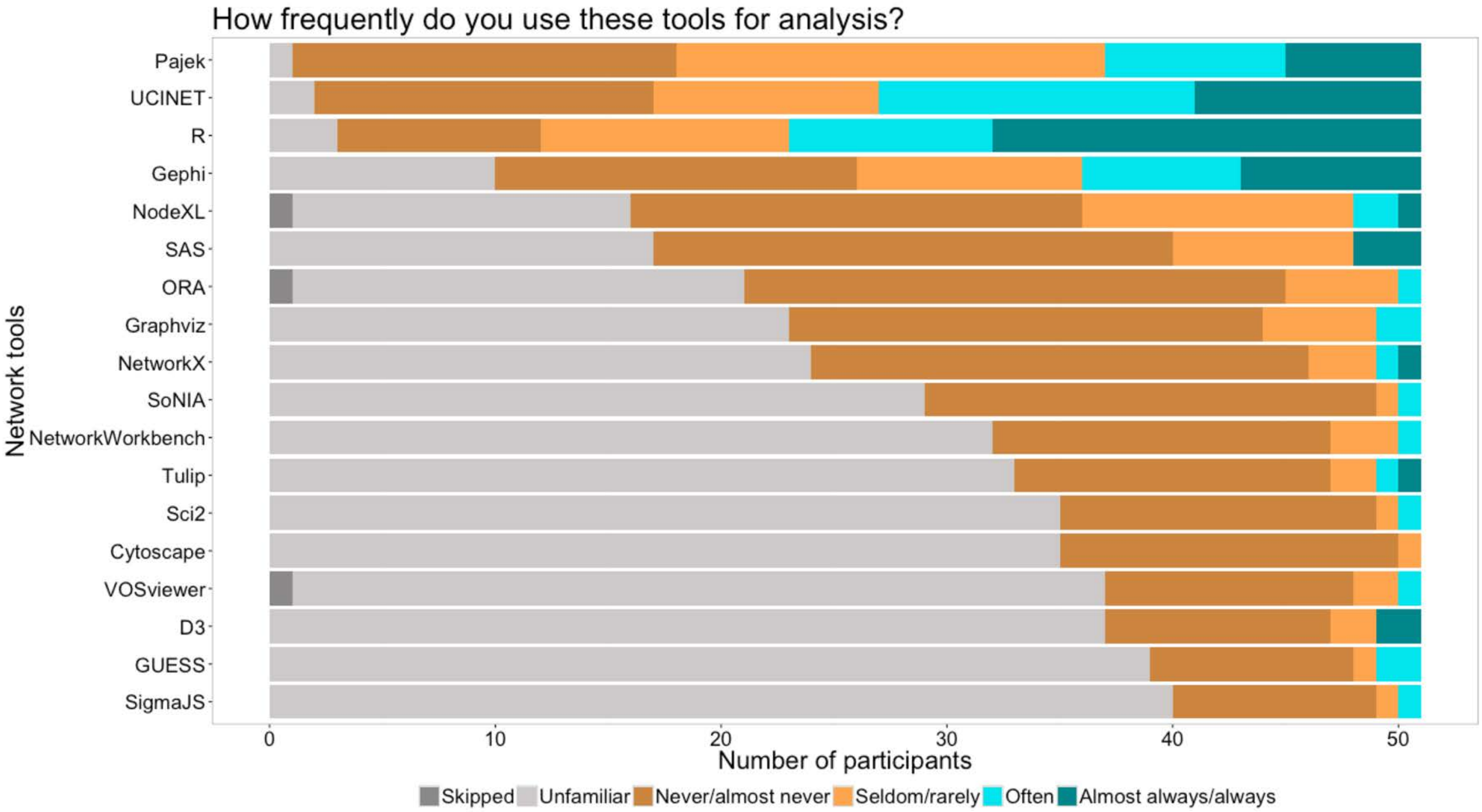

How frequently do you use these tools for analysis?
Network tools
Pajek
UCINET
R
Gephi
NodeXL
SAS
ORA
Graphviz
NetworkX
SoNIA
NetworkWorkbench
Tulip
Sci2
Cytoscape
VOSviewer
D3
GUESS
SigmaJS
0
10
20
30
40
50
Number of participants
compiled_answer
Skipped
Unfamiliar
Never/almost never
Seldom/rarely
Often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

Select colors manually, or use alternate palette

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

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?
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

Dataset 3:
Star Wars opinion survey

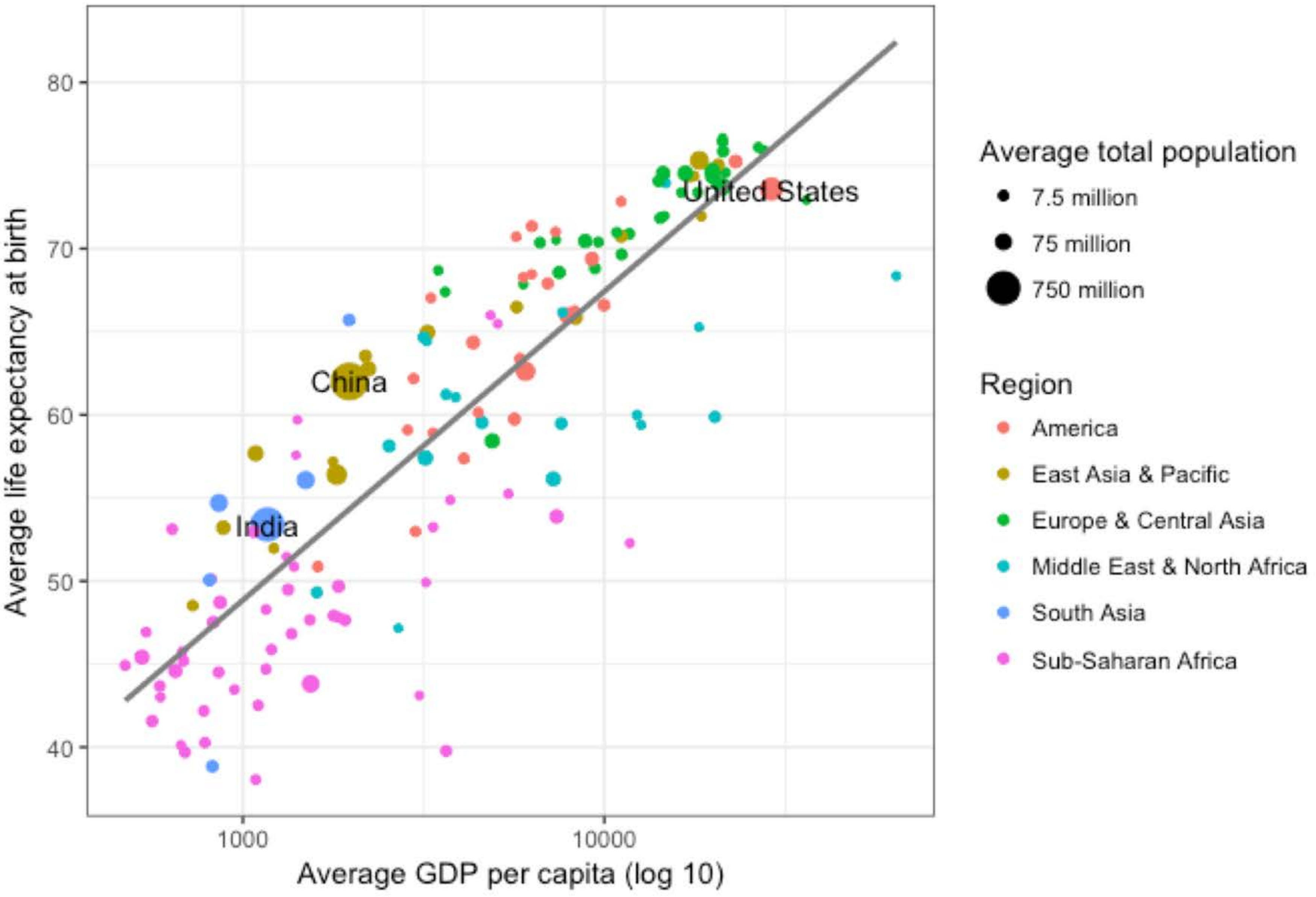
https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/

Saving charts out

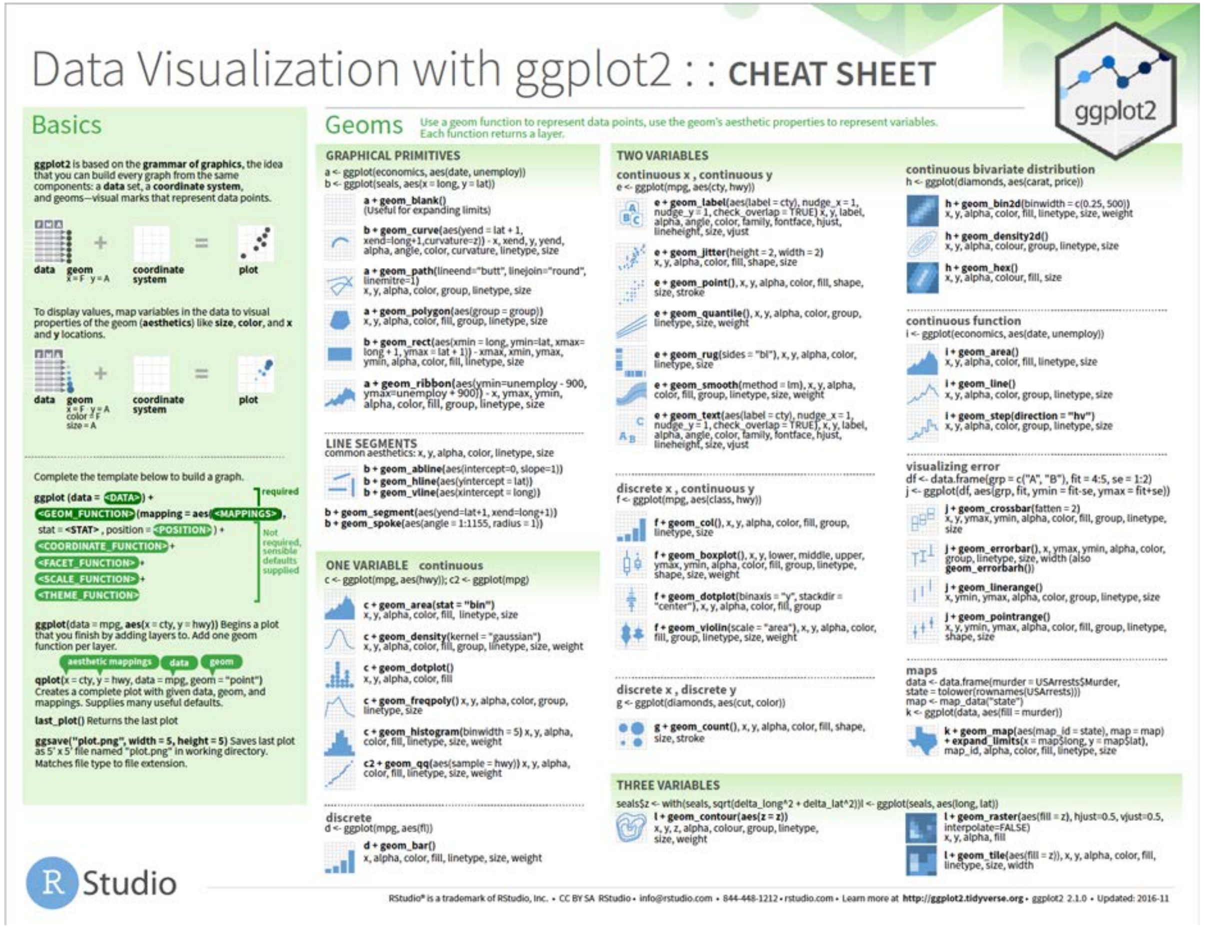
Dataset 4: Gapminder Data

http://www.gapminder.org/

Averages across all years of the traditional Gapminder dataset



ggplot2 Cheat Sheet



Data Visualization with ggplot2 : : CHEAT SHEET

ggplot2

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.

data
geom
x = F y = A
+
coordinate system
=
plot

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_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

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

aesthetic mappings · data · geom

qplot(x = cty, 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 <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
(Useful for expanding limits)

b + geom_curve(aes(yend = lat + 1, xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend="butt", linejoin="round", linemitre=1)
x, y, alpha, color, 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)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

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

LINE SEGMENTS

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

b + geom_abline(aes(intercept=0, slope=1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend=lat+1, xend=long+1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete
d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

continuous x , continuous y
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

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

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

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

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

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x , continuous y
f <- ggplot(mpg, aes(class, hwy))

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, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"), x, y, alpha, color, fill, group

f + geom_violin(scale = "area"), x, y, alpha, color, fill, group, linetype, size, weight

discrete x , discrete y
g <- ggplot(diamonds, aes(cut, color))

g + geom_count(), x, y, alpha, color, fill, shape, size, stroke

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
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

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

j + geom_errorbar(), x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

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

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

maps
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("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, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))l <- ggplot(seals, aes(long, lat))

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

l + geom_raster(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

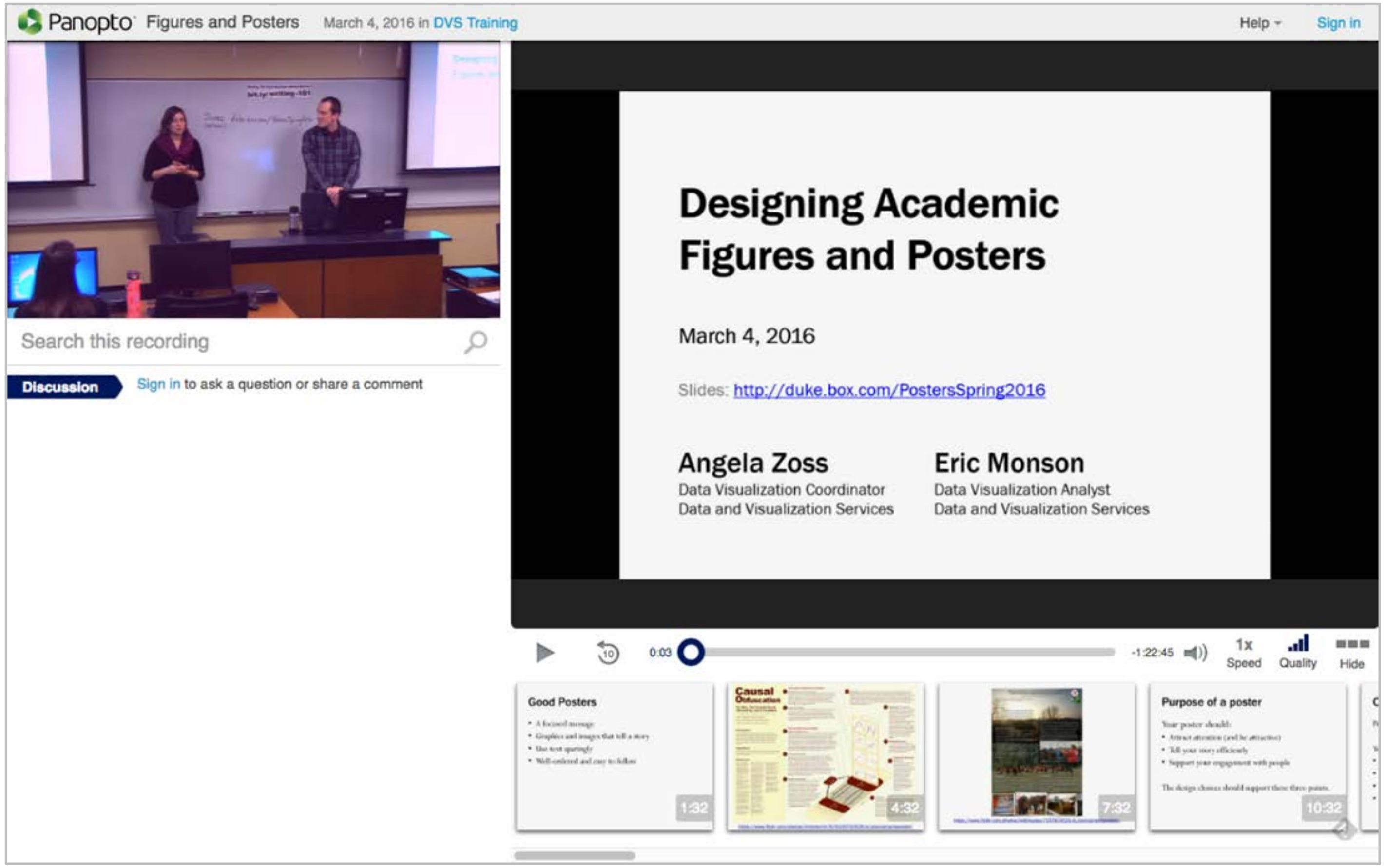
RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **http://ggplot2.tidyverse.org** • ggplot2 2.1.0 • Updated: 2016-11

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



Designing Academic Figures and Posters

March 4, 2016

Slides: http://duke.box.com/PostersSpring2016

Angela Zoss
Data Visualization Coordinator
Data and Visualization Services

Eric Monson
Data Visualization Analyst
Data and Visualization Services

Good Posters

- A focused message
- Graphics and images that tell a story
- Use text sparingly
- Well-ordered and easy to follow

Purpose of a poster

Your poster should:

- Attract attention (and be attractive)
- Tell your story efficiently
- Support your engagement with people

The design choices should support these three points.

http://bit.ly/DVSvideos

Questions?

angela.zoss@duke.edu