# Visualization in R using ggplot2

Angela Zoss

9/19/19

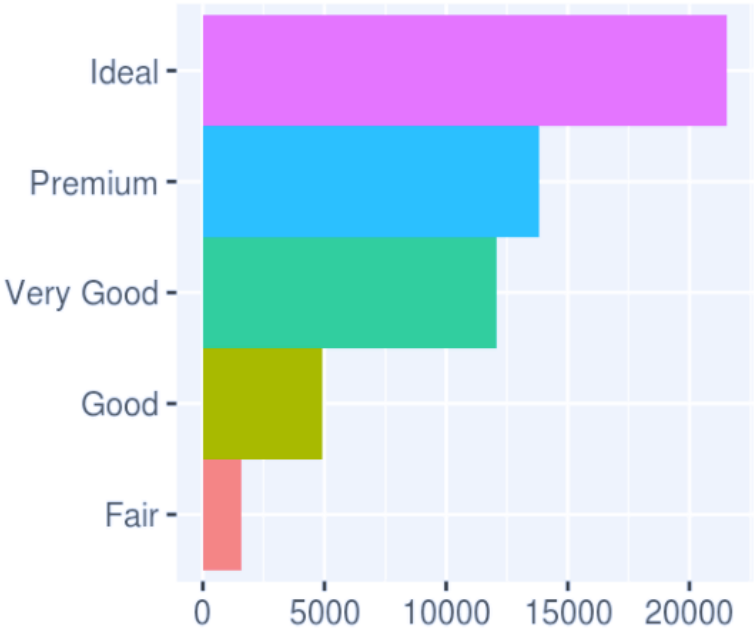
https://github.com/amzoss/ggplot2-F19

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

# 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.

# Working in RStudio

# Using RStudio

- Projects
- R Markdown
- Cheat sheets

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

# Don’t have it installed?

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

# Create a new project with workshop files

URL: https://github.com/amzoss/ggplot2-F19

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

**In RStudio:**

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

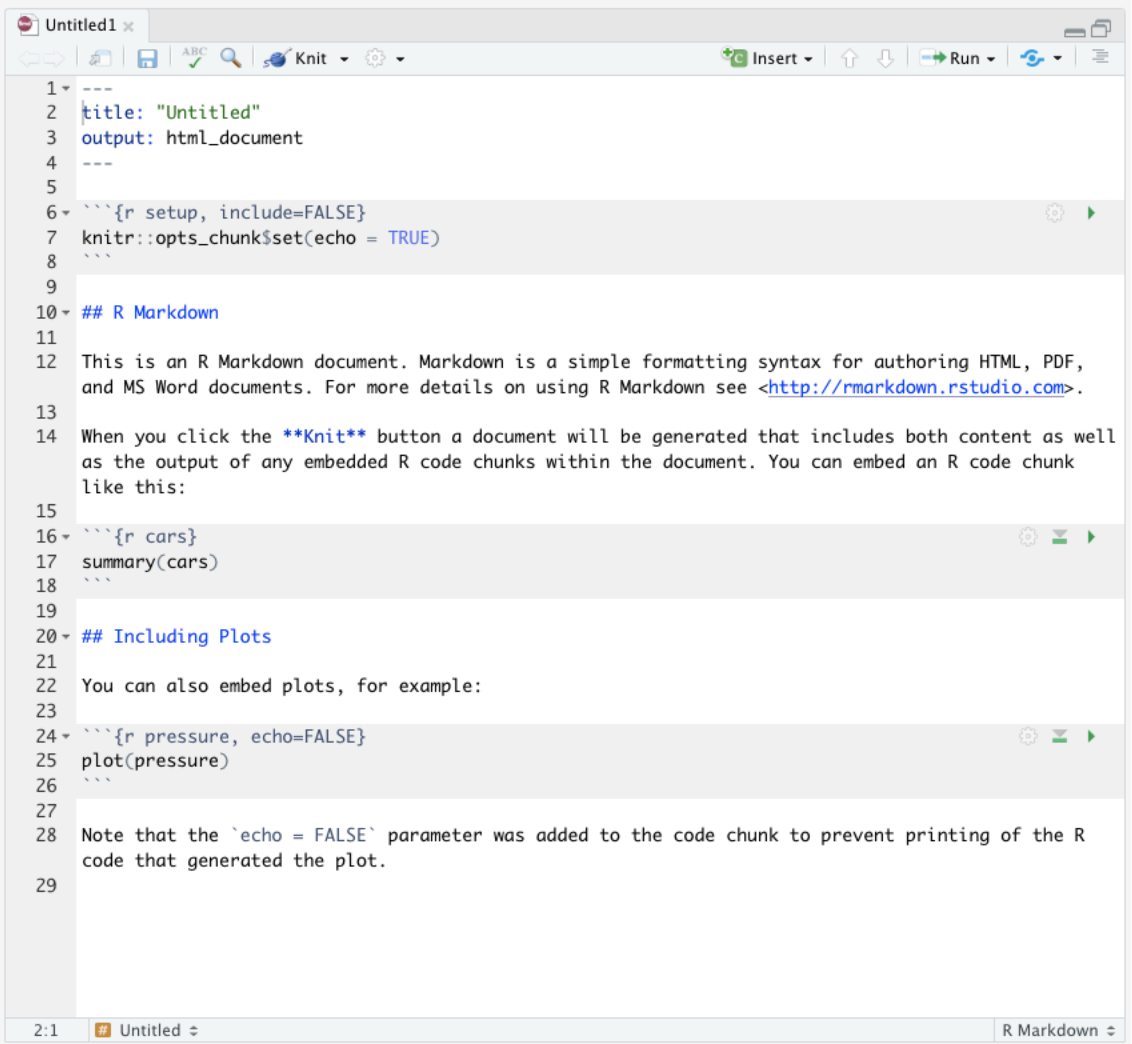
# Why R Markdown?

- 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.

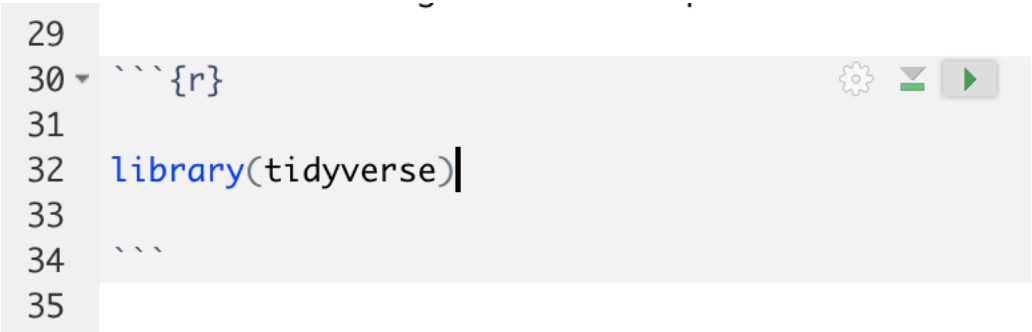
# R Markdown files

- 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

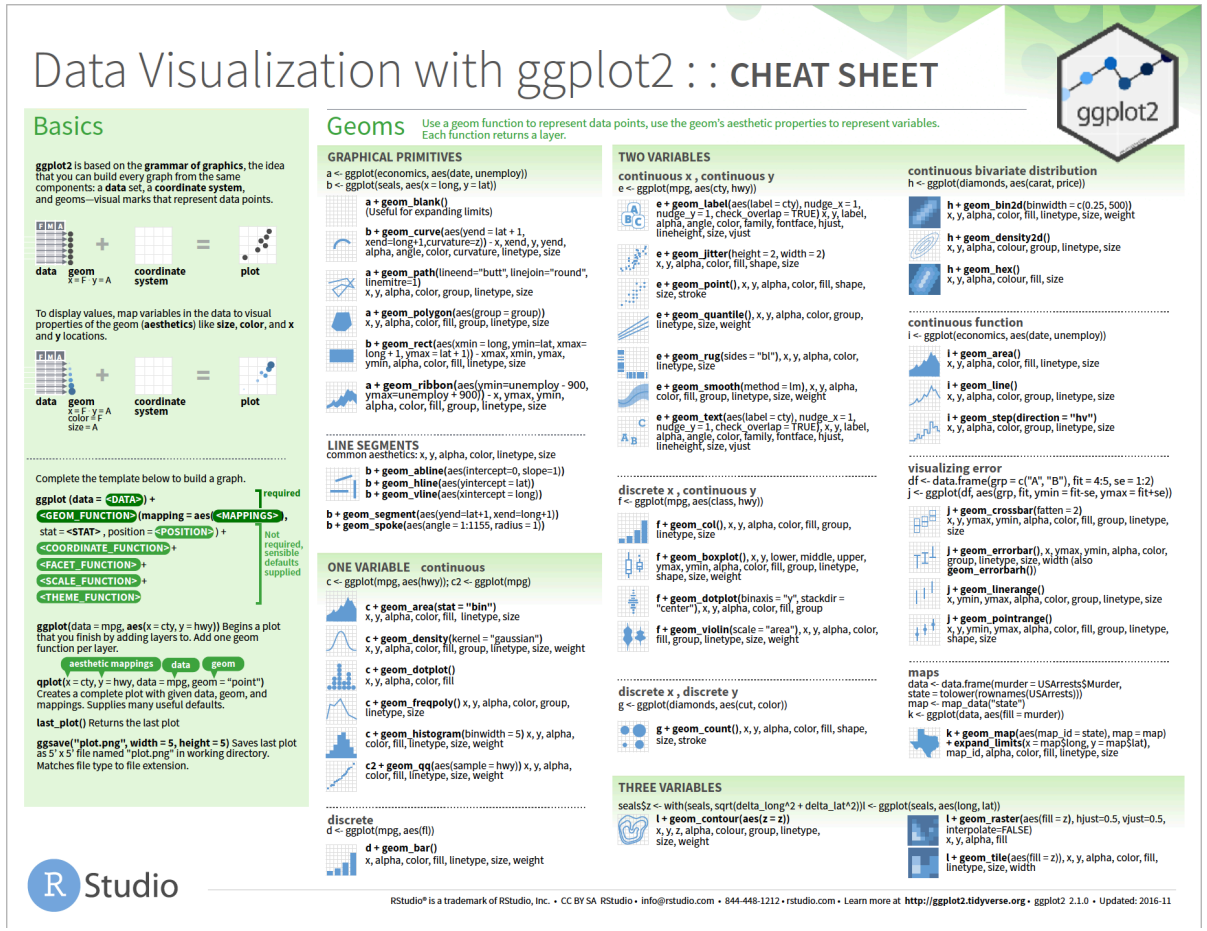


# ggplot2 Cheat Sheet

Help →

Cheatsheets →

Data Visualization with ggplot2

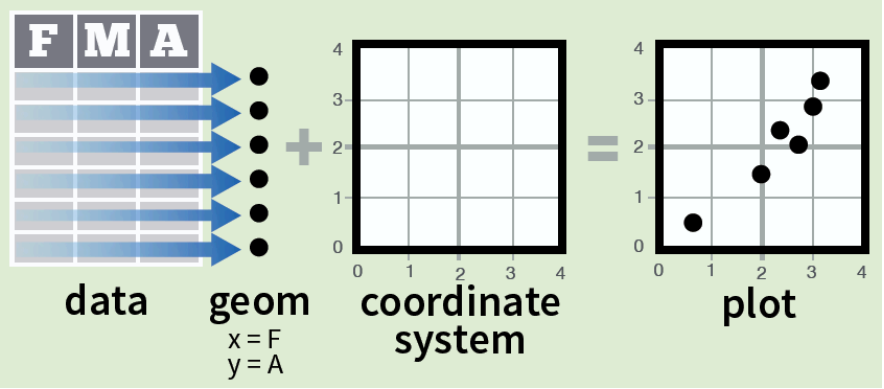



https://www.rstudio.com/resources/cheatsheets/#ggplot2

# ggplot2: making a basic plot

# Basic elements in any ggplot2 visualization

- **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

# Template for a simple plot

**Main function**

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

+

**Shape layer**

```
geom_...( non-variable adjustments )
```

# Step-by-step

1. Set the data

**Main function**

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

**Shape layer**

# 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)
```

# Step-by-step

1. Set the data
2. Map variables to aesthetics

**Main function**

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

**Shape layer**

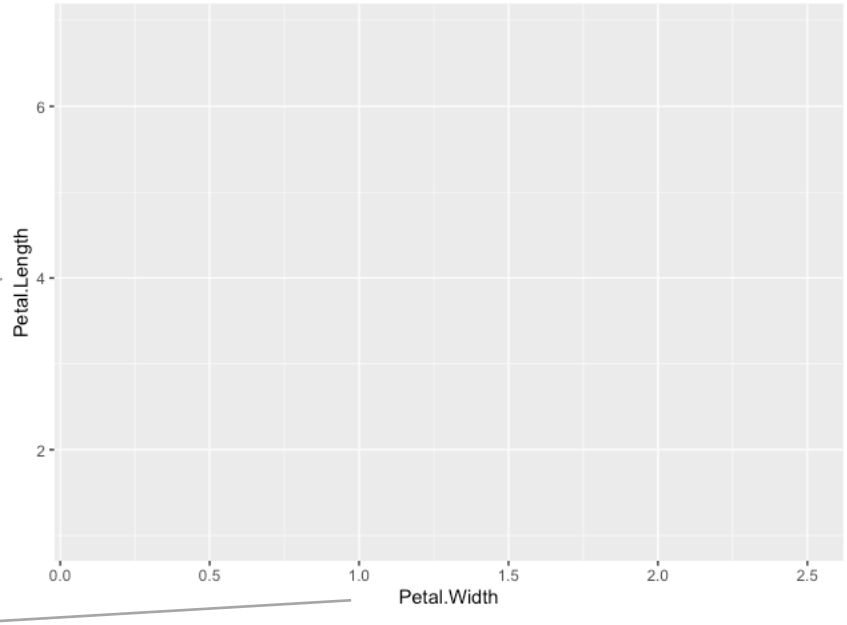
# 2. Map variables to aesthetics

"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,
        aes(x=Petal.Width, y=Petal.Length,
            color=Species))
```



# Step-by-step

1. Set the data
2. Map variables to aesthetics
3. Choose a shape layer

**Main function**

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

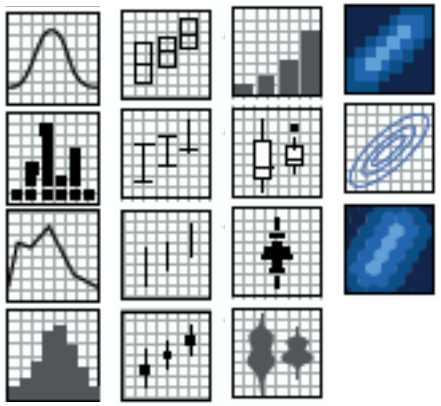
+

**Shape layer**

```
geom_...( )
```

# Types of geoms

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



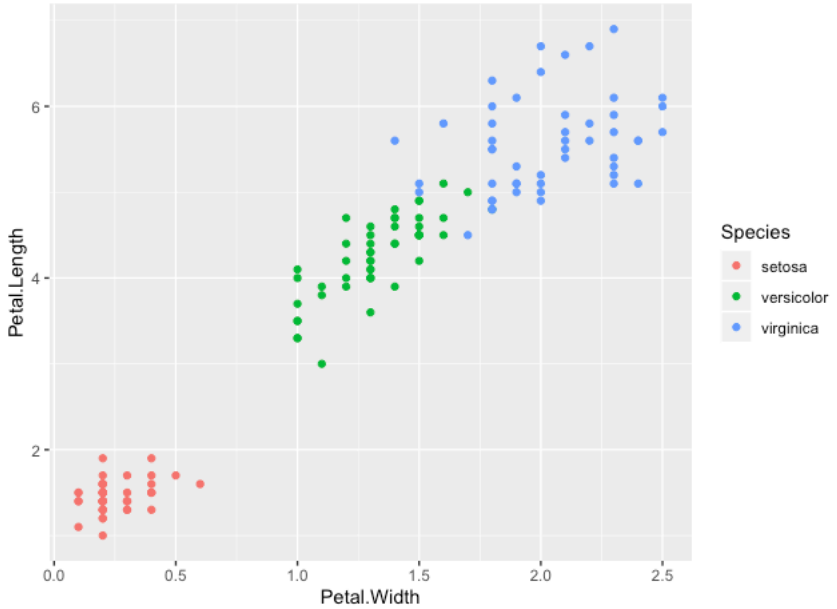
http://bit.ly/ggplot2-cheatsheet

# 3. 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,
        aes(x=Petal.Width, y=Petal.Length,
            color=Species)) +
geom_point()
```



# Step-by-step

1. Set the data
2. Map variables to aesthetics
3. Choose a shape layer
4. Add non-variable adjustments

**Main function**

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

+

**Shape layer**

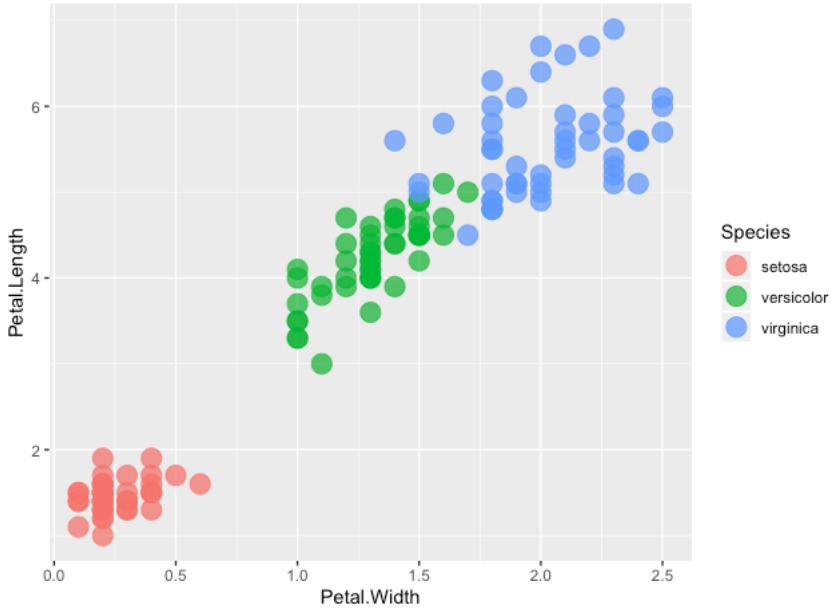
```
geom_...( non-variable adjustments )
```

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



# ggplot2: inheritance

# Template for a simple plot

**Main function**

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

+

**Shape layer**

```
geom_...( non-variable adjustments )
```

# Behind the scenes

**Main function**

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

+

**Shape layer**

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

# Inheritance

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

**Main function**

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

+

**Shape layer**

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

+

**Shape layer**

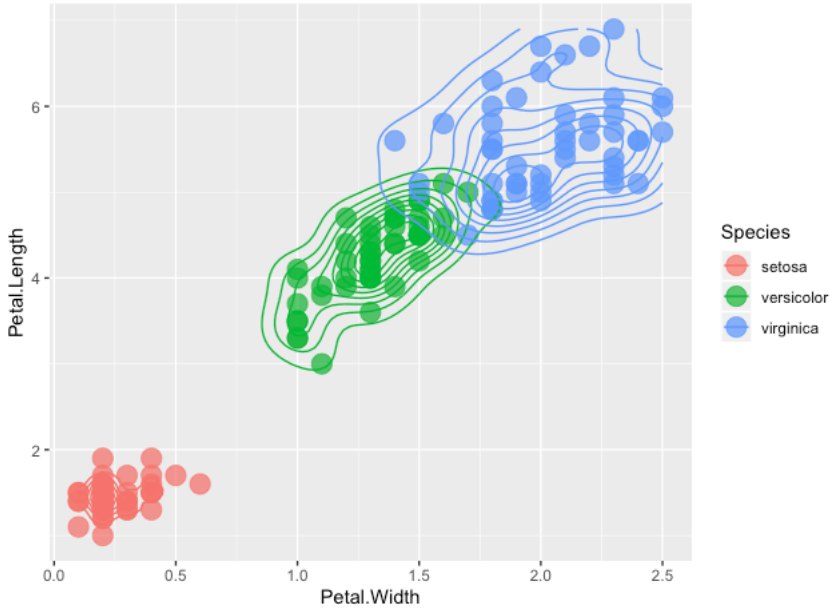
```
geom_...( data = data frame ,
          mapping = aes(variable mappings) ,
          non-variable adjustments )
```

# Adding a new shape layer: geom_density2d()

"iris"

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

```
ggplot(data=iris,
        aes(x=Petal.Width, y=Petal.Length,
            color=Species)) +
geom_point(size=5, alpha=.75) +
geom_density2d()
```



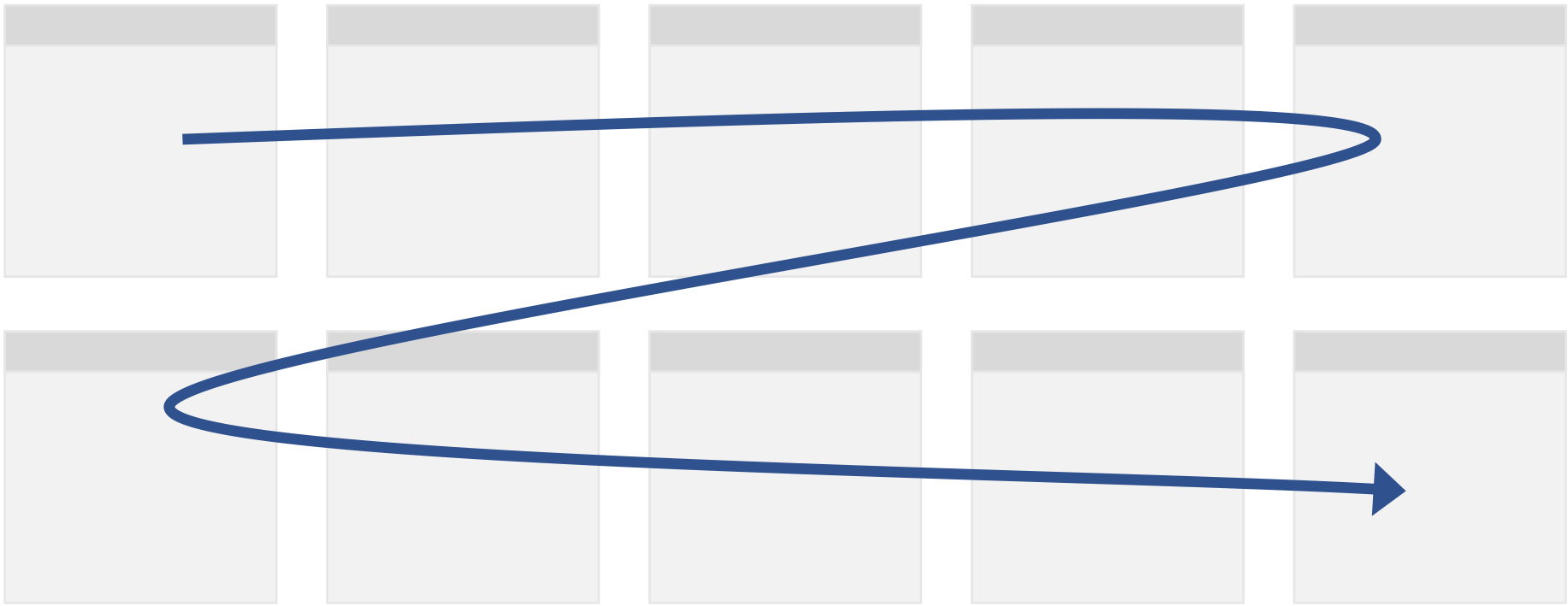
# Dataset 1:
# Game of Thrones character ratings

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

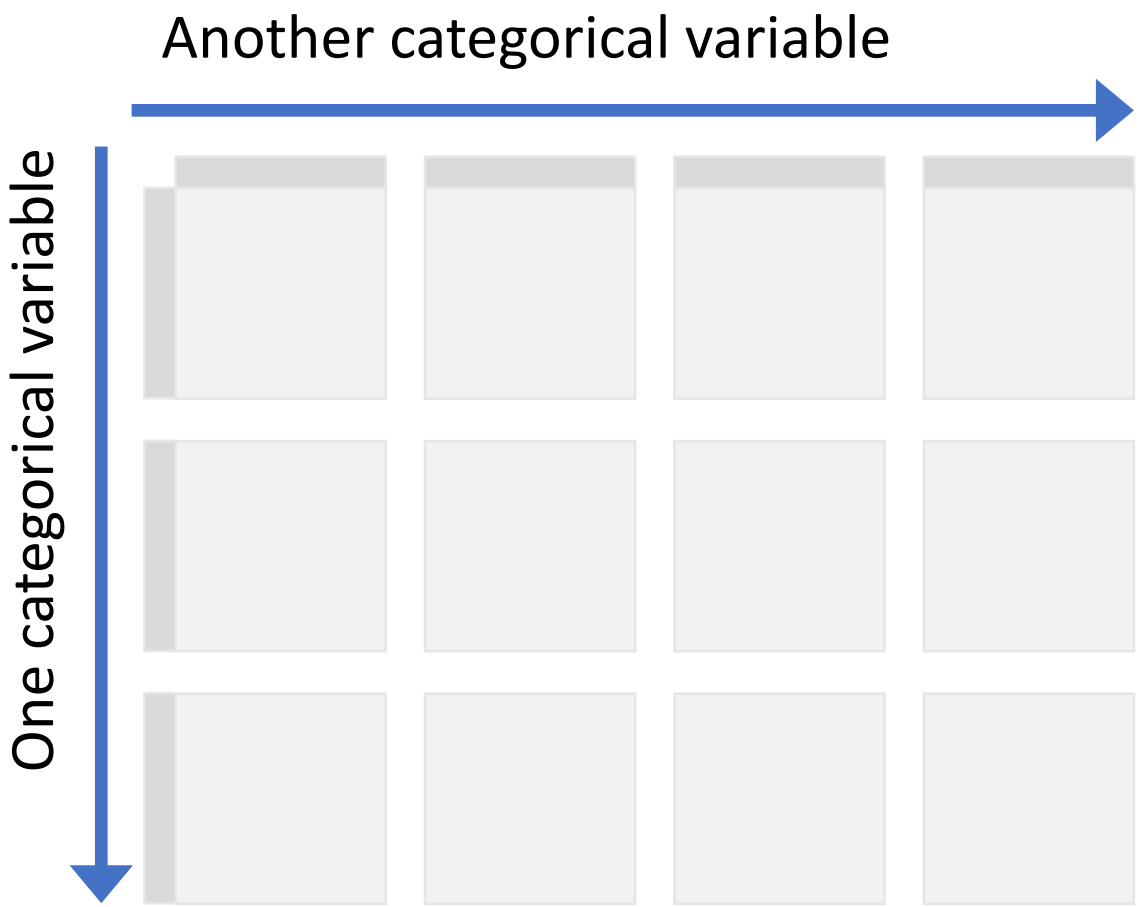
# Creating repeated charts

# facet_wrap()

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



# facet_grid()




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

# Helpful data manipulation

# About %>%

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

```
data_frame %>% function(args)
```

# 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

# 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/#import

# Dataset 2:
# Star Wars character data

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

# 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 |

# 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
```

# 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

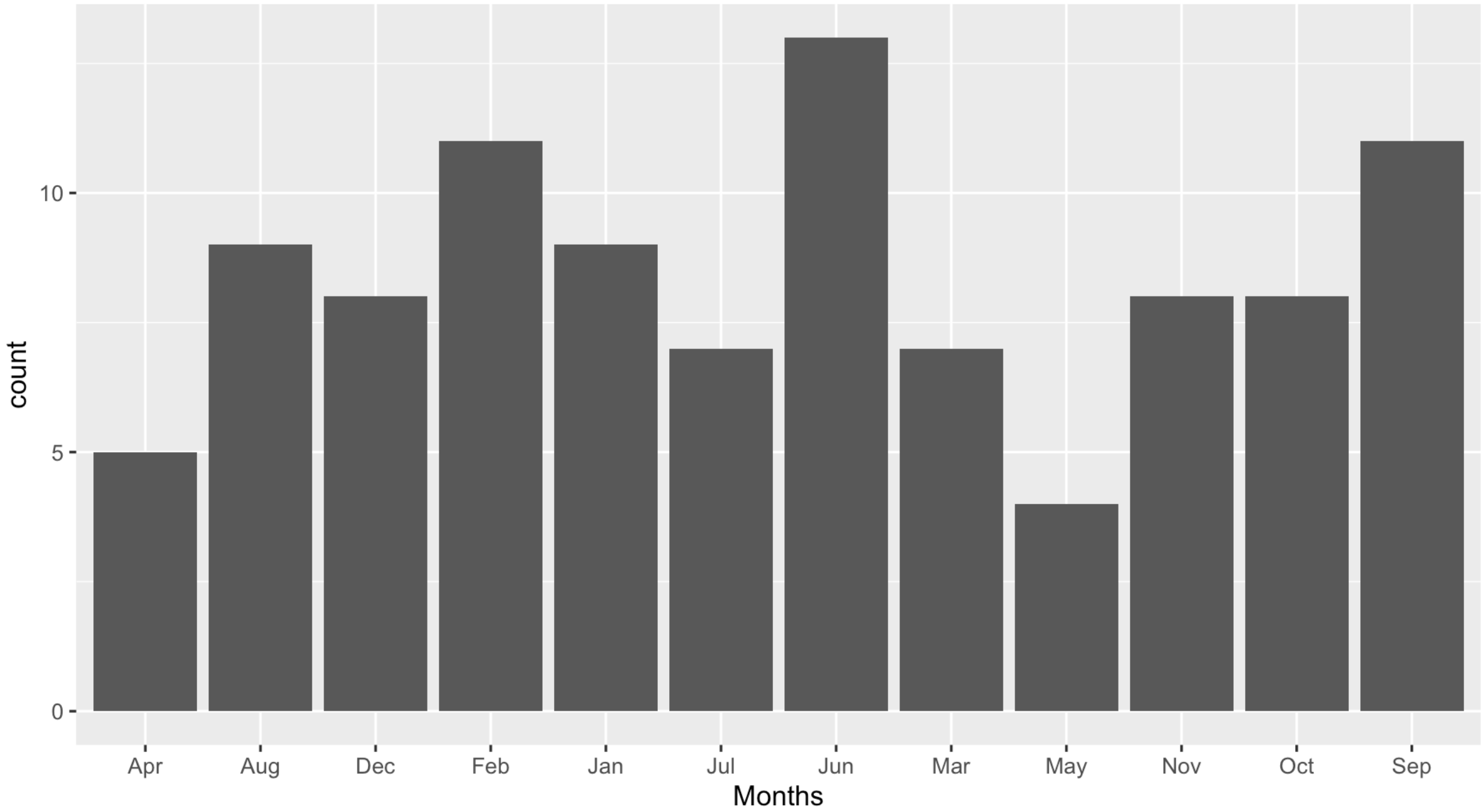
# Factoring resources

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

# Design Principles for Text Variables

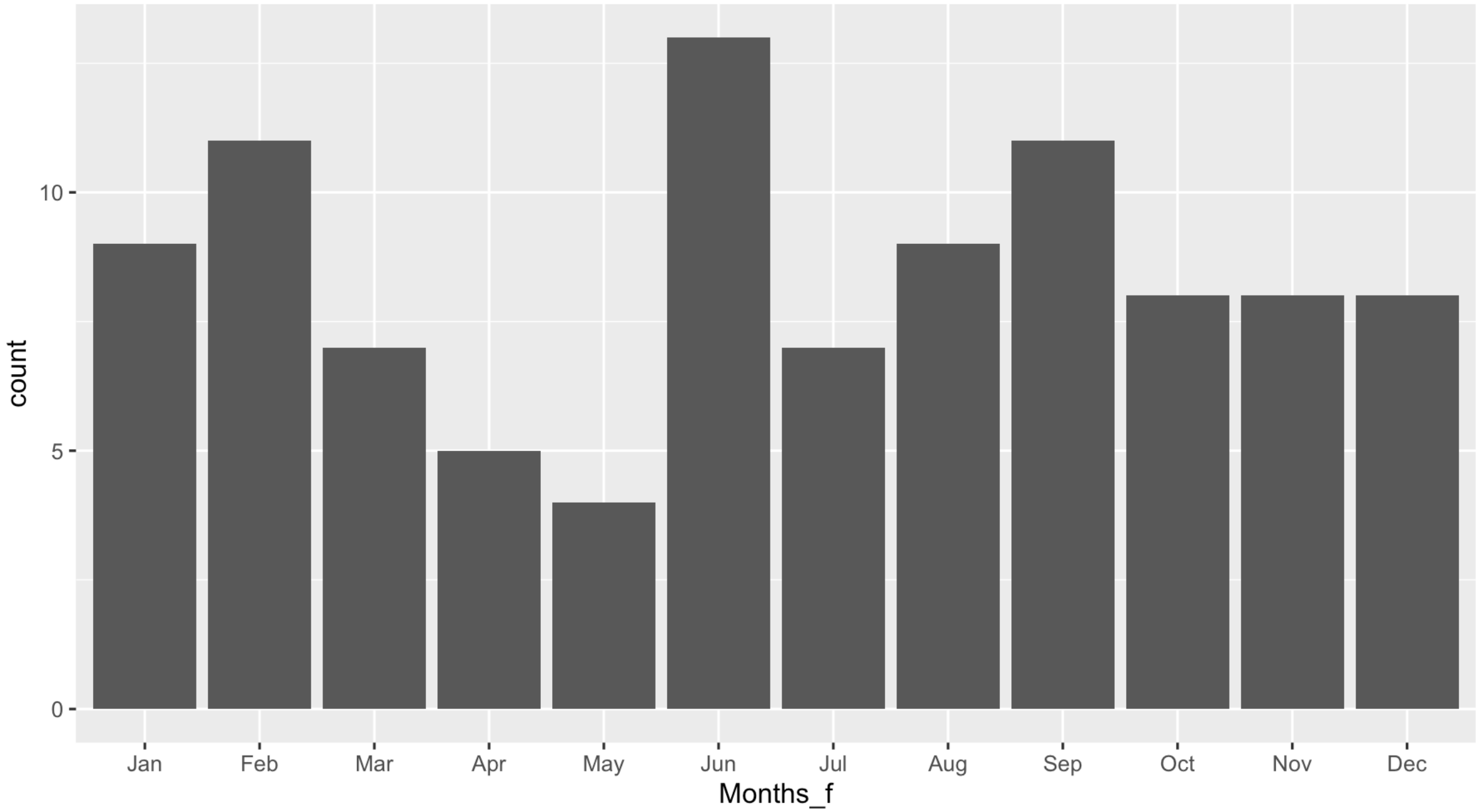
# Principle 1: Order matters

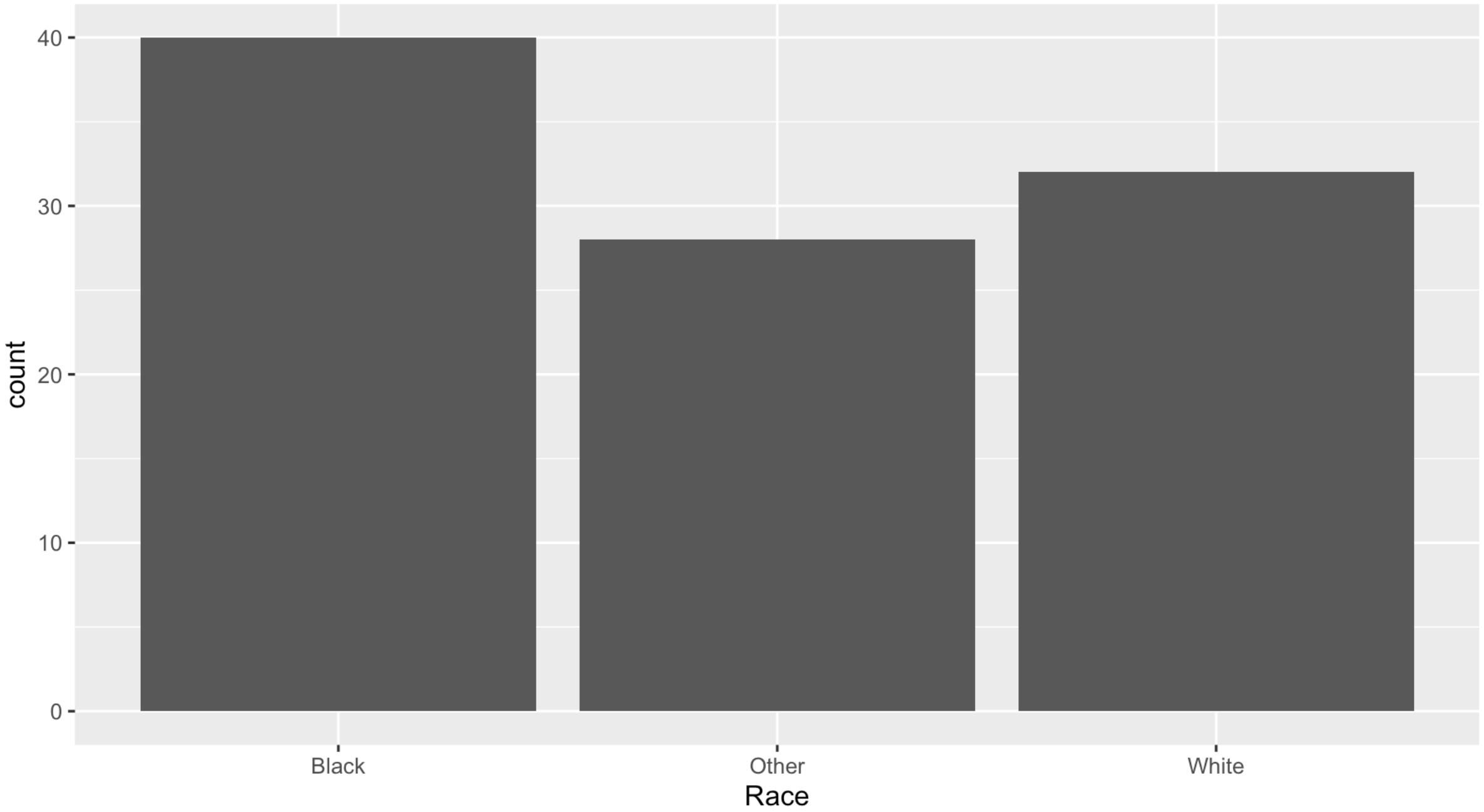

count
10
5
0
Apr
Aug
Dec
Feb
Jan
Jul
Jun
Mar
May
Nov
Oct
Sep
Months

# 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))
```

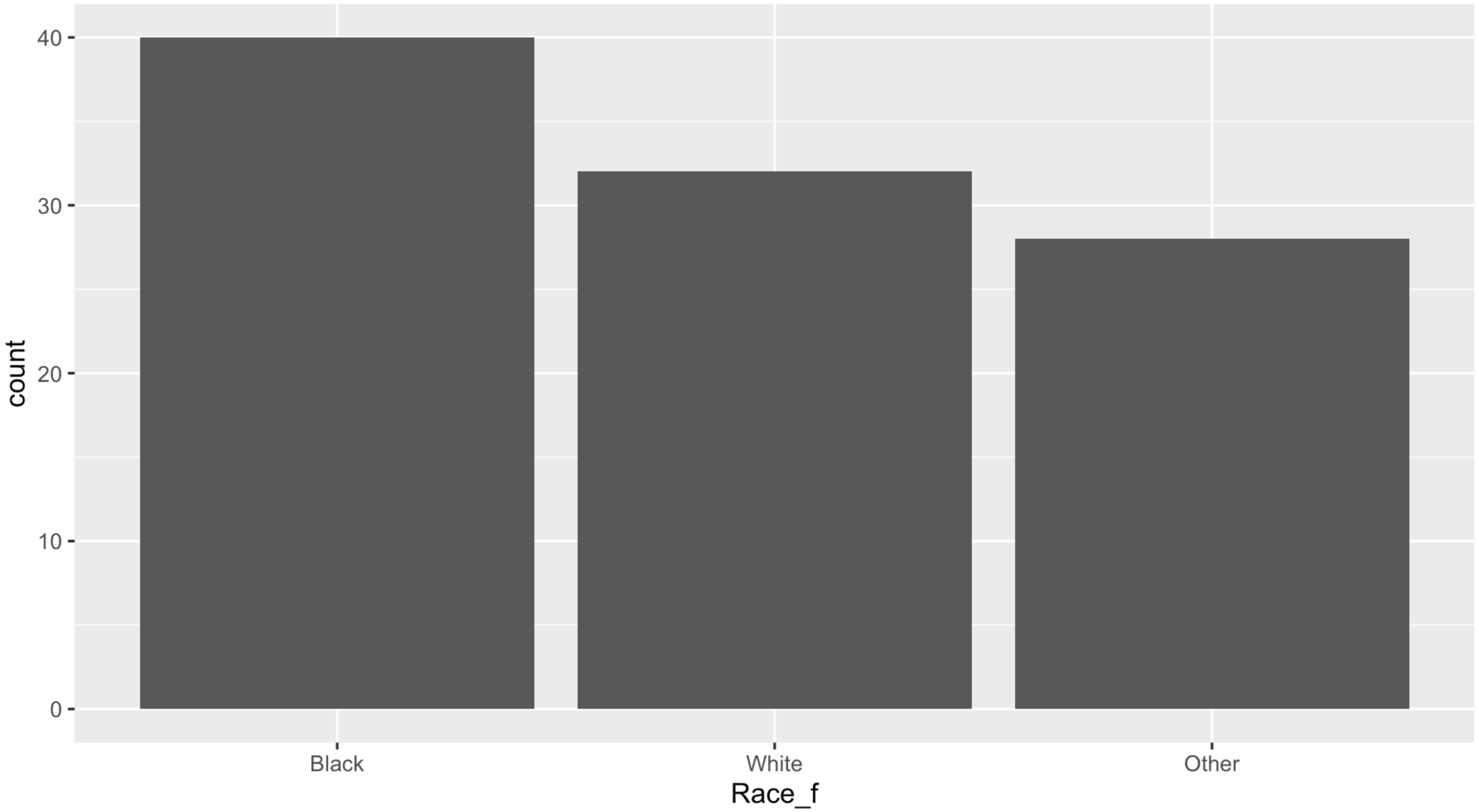

count
10
5
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Months_f


40
30
20
10
0
count
Black
Other
White
Race

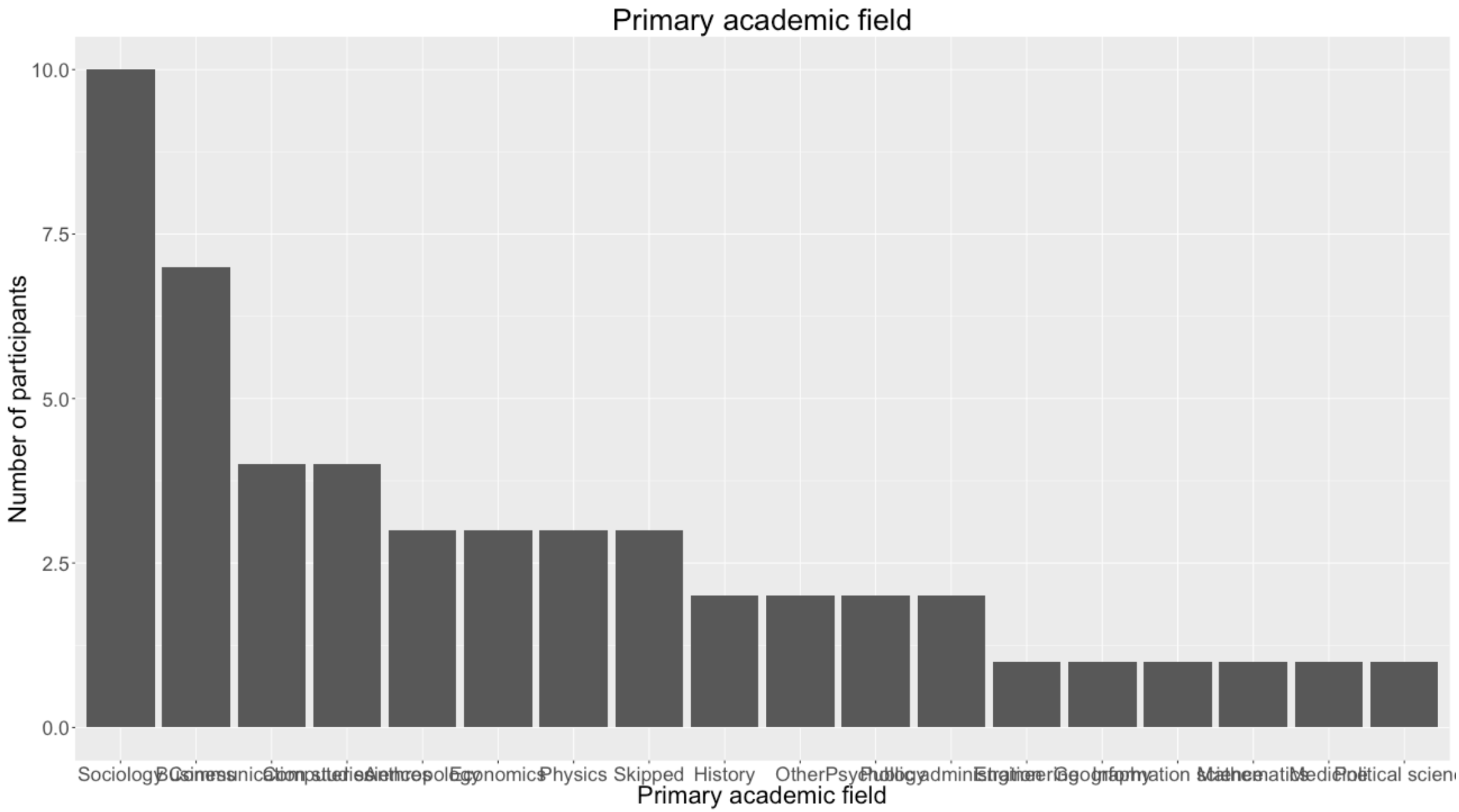
# Order by value (using forcats)

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


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

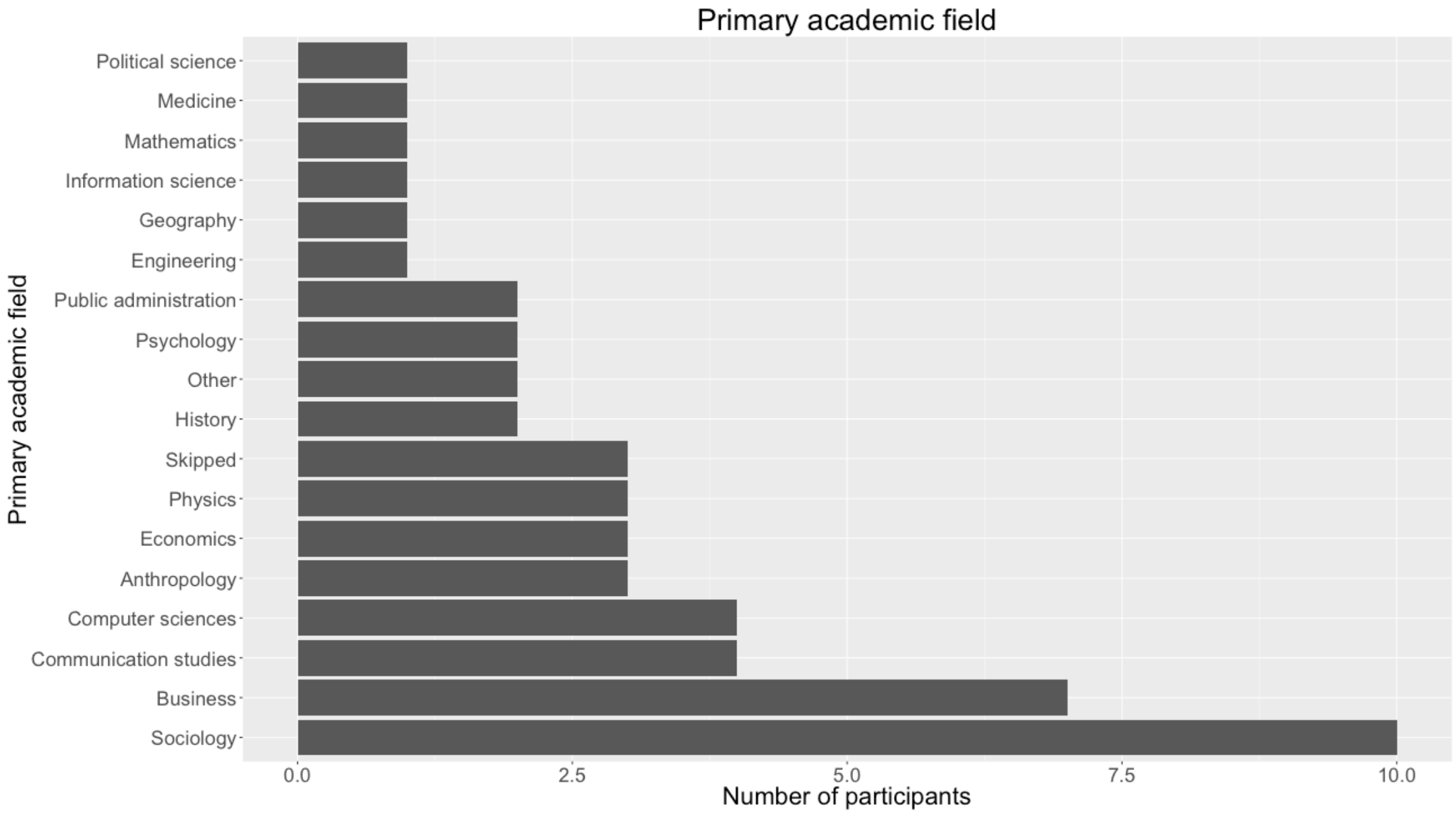

40
30
20
10
0
count
Black
White
Other
Race_f

# 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
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

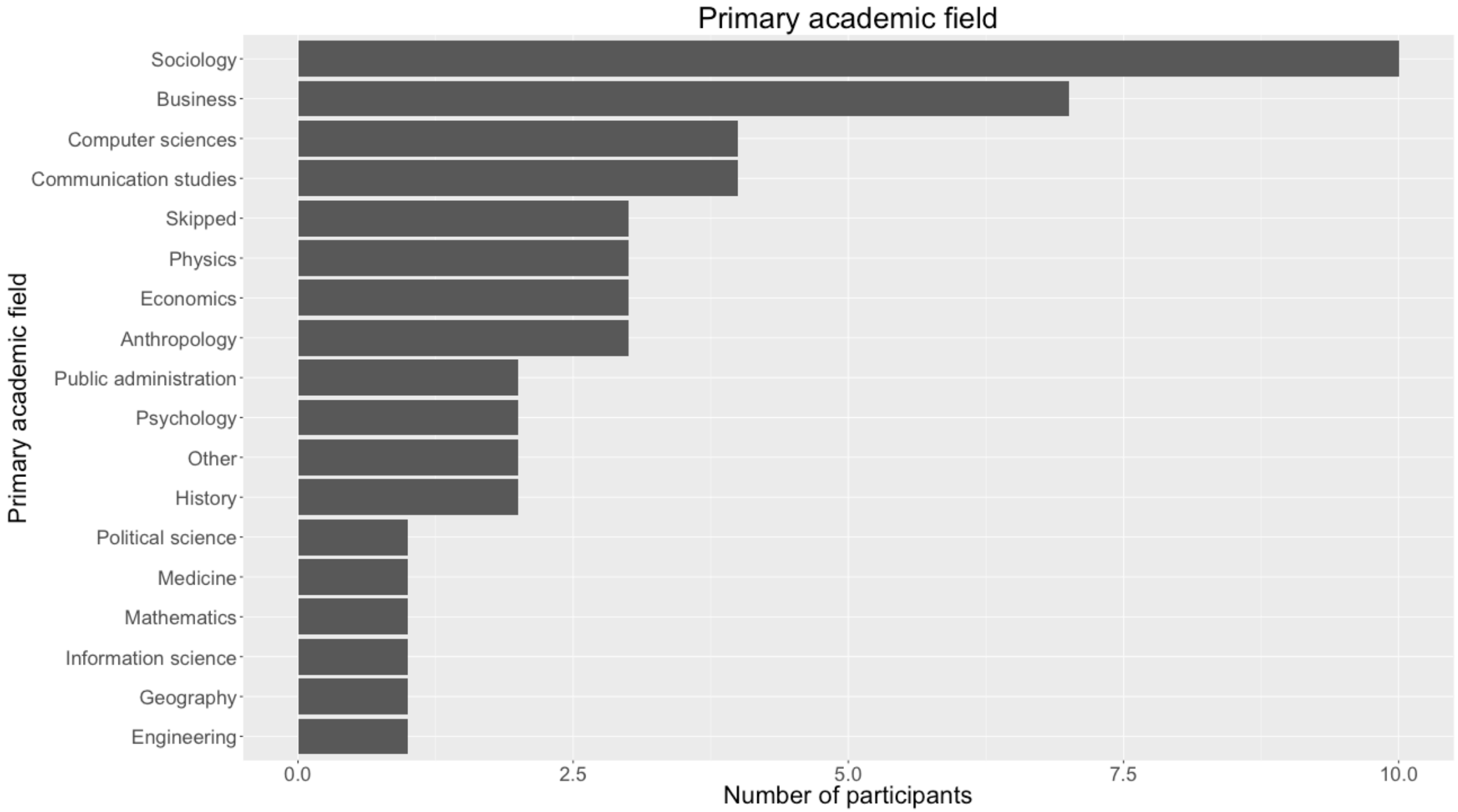
```
+ coord_flip()
```


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

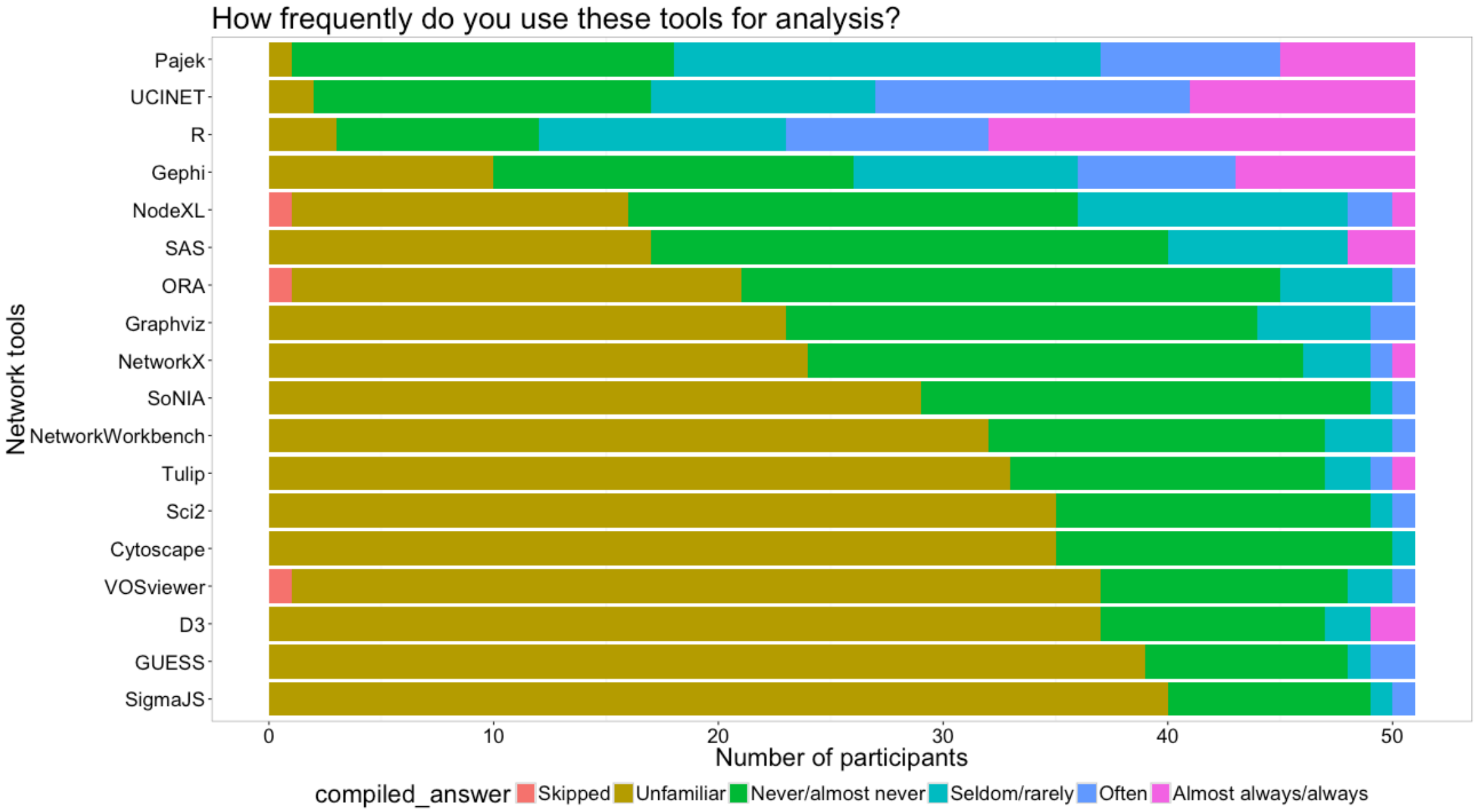
# Oops!

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

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

# Principle 3:
# Select meaningful colors

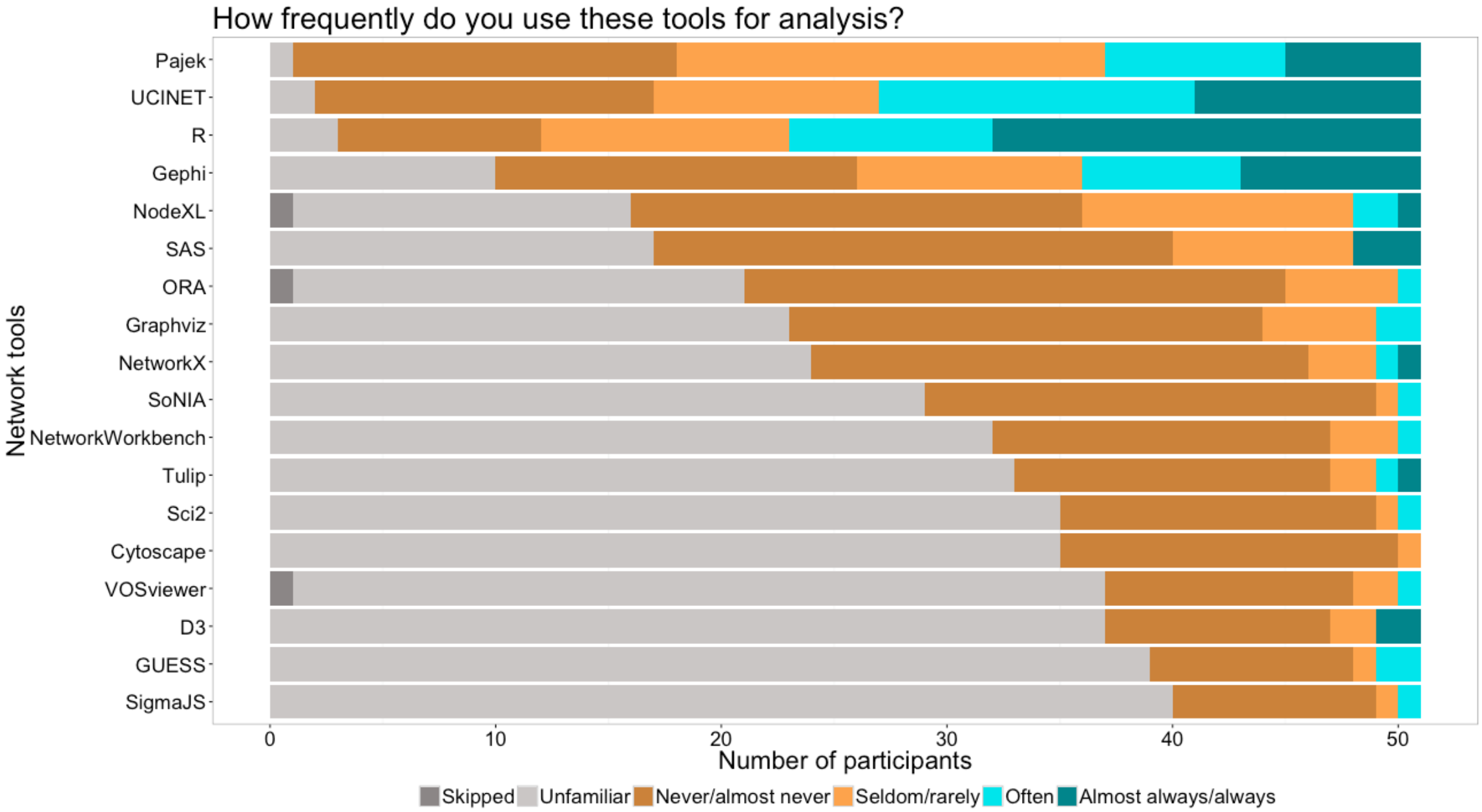

How frequently do you use these tools for analysis?
Network tools
Pajek
UCINET
R
Gephi
NodeXL
SAS
ORA
Graphviz
NetworkX
SoNIA
NetworkWorkbench
Tulip
Sci2
Cytoscape
VOSviewer
D3
GUESS
SigmaJS
0
10
20
30
40
50
Number of participants
compiled_answer
Skipped
Unfamiliar
Never/almost never
Seldom/rarely
Often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

# Select colors manually, or use alternate palette

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



# 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/

# Final advice

# Additional decisions

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!**

# 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.

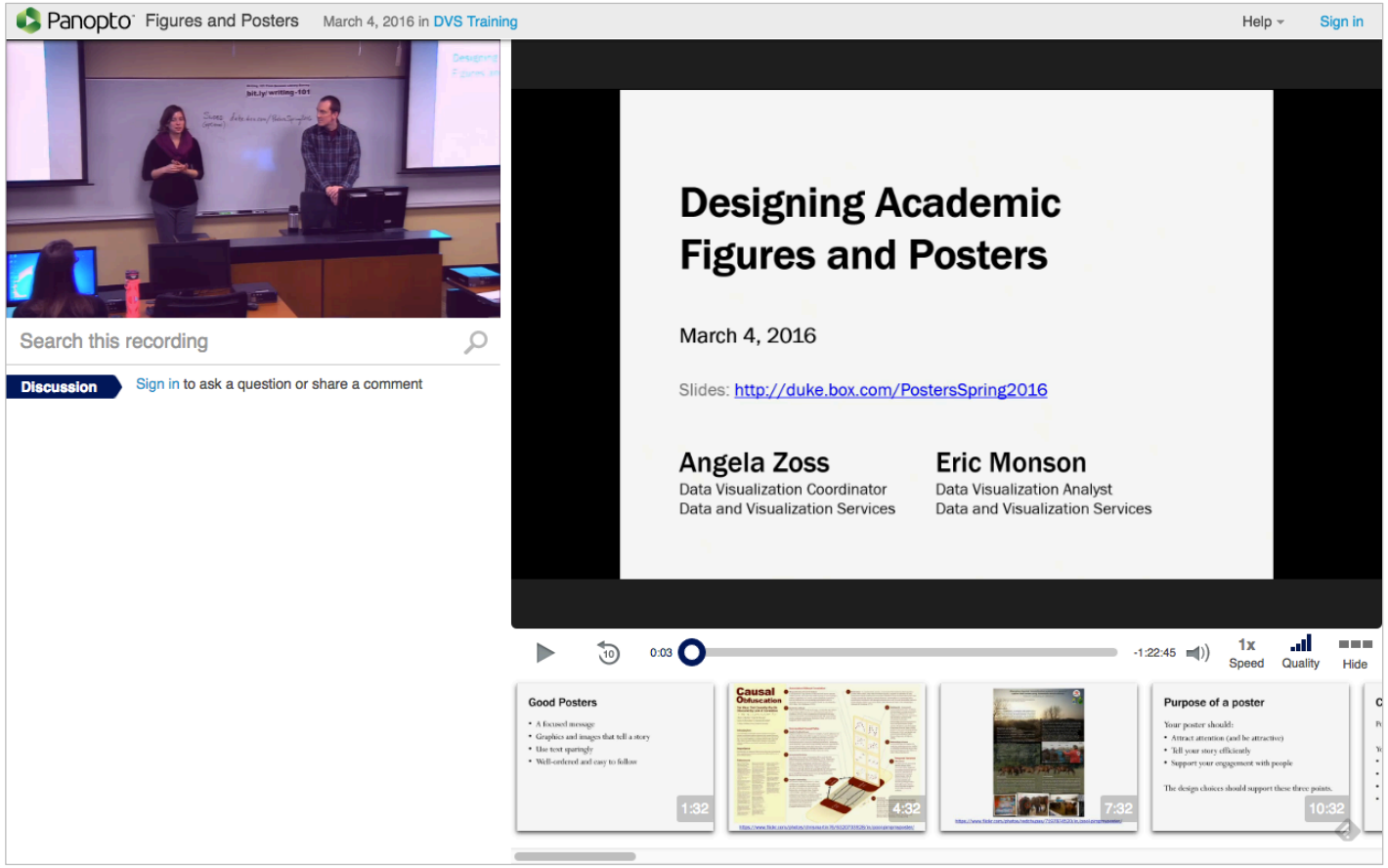
# Other helper packages

- 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

# Videos of past workshops




http://bit.ly/DVSvideos

# Questions?

angela.zoss@duke.edu