

DATA VISUALIZATION

SICSS-NDSU

<https://slides.com/zoltanpm/sicss-ndsu-viz/>

OVERVIEW

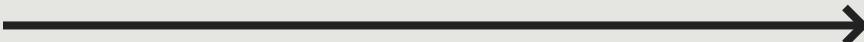
ggplot & the Grammar of Graphics



Color theory



Maps



Networks



WHY DO WE VISUALIZE



EXPLORE

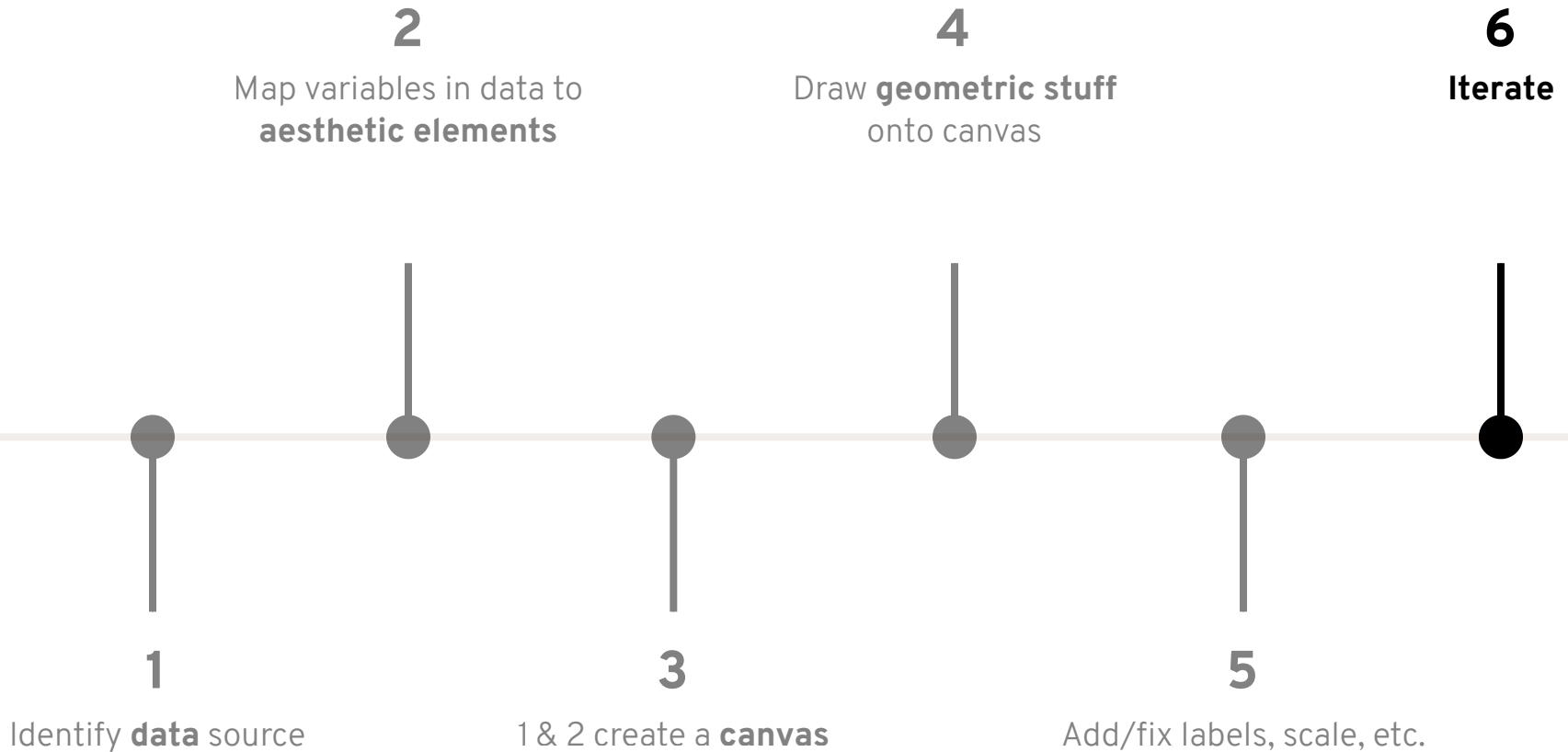


ANALYZE



COMMUNICATE

GRAMMAR OF GRAPHICS



```
1 library(tidyverse)
2
3
4 p <- ggplot(data,
5               mapping = aes(x, y, color, size, etc))
6
7 p + geom_point/line/boxplot/etc(visual variables like opacity or color) +
8     labs() +
9     coordinate systems() +
10    etc.
```



SOME COLOR THEORY

... and how to color in R

HOW HUMANS SEE COLOR: HSL

- **Hue**
 - color wheel
 - qualitative
- **Saturation**
 - pure through muted
 - color to gray
- **Lightness**
 - contrast
 - white - color - black

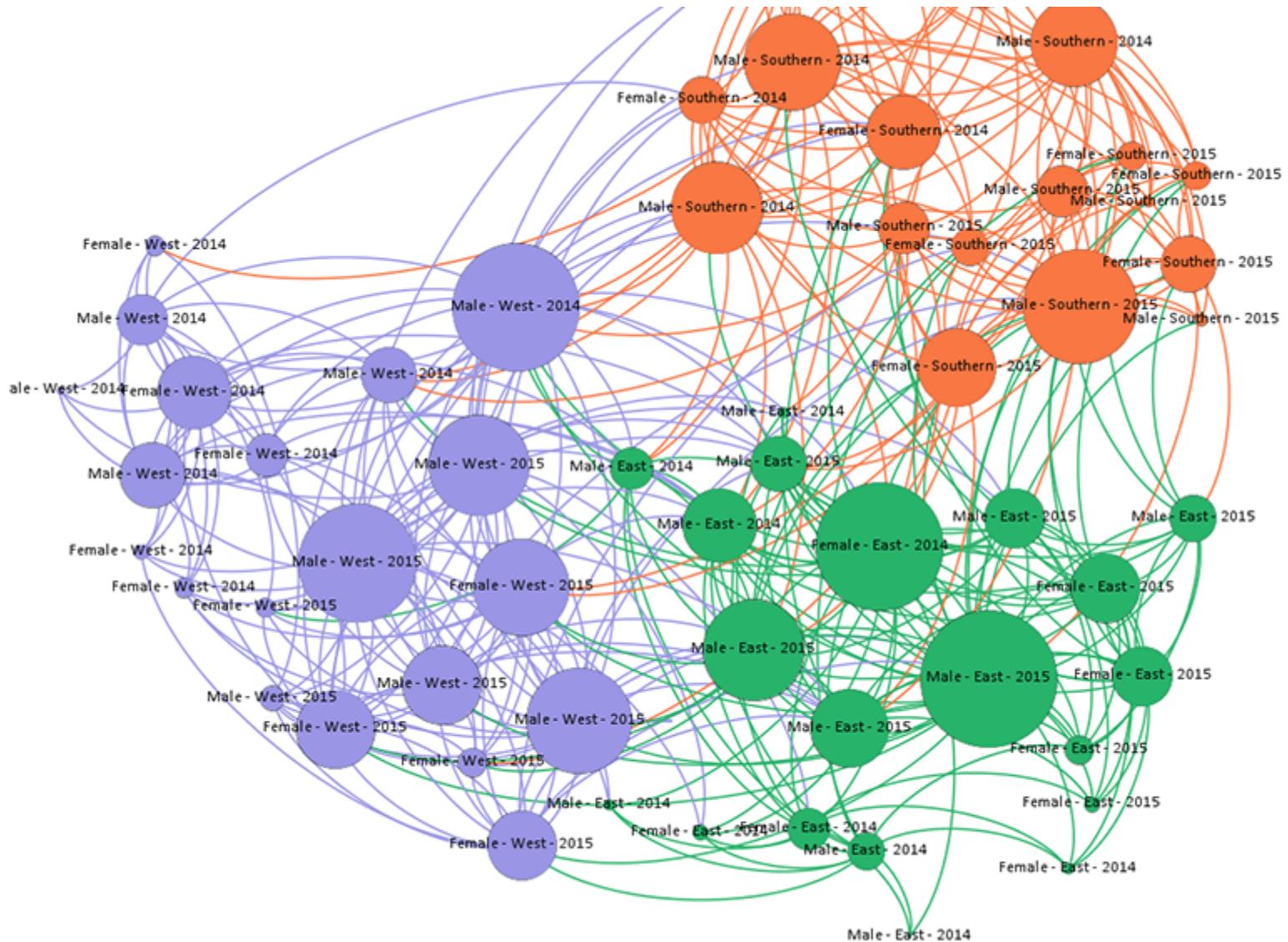
HOW COMPUTERS SEE COLOR

- Must be binary
- Types of representation
 - RGB
 - CMYK
 - HEX

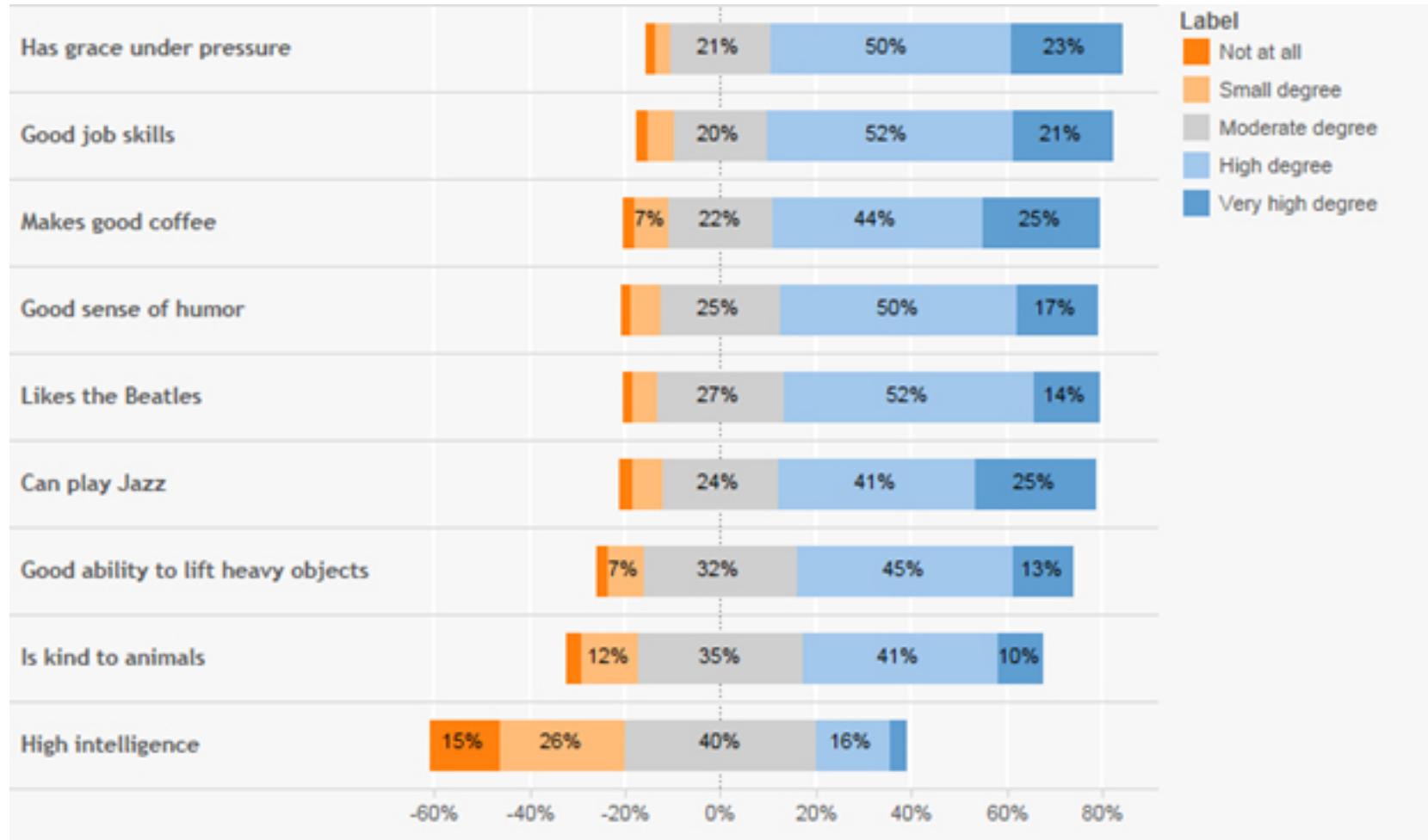


HEX values of the same three colors

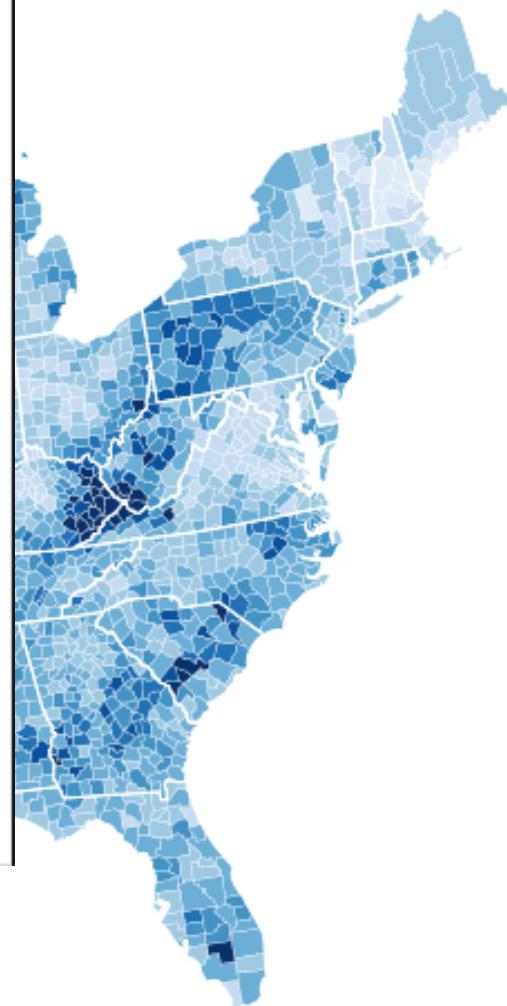
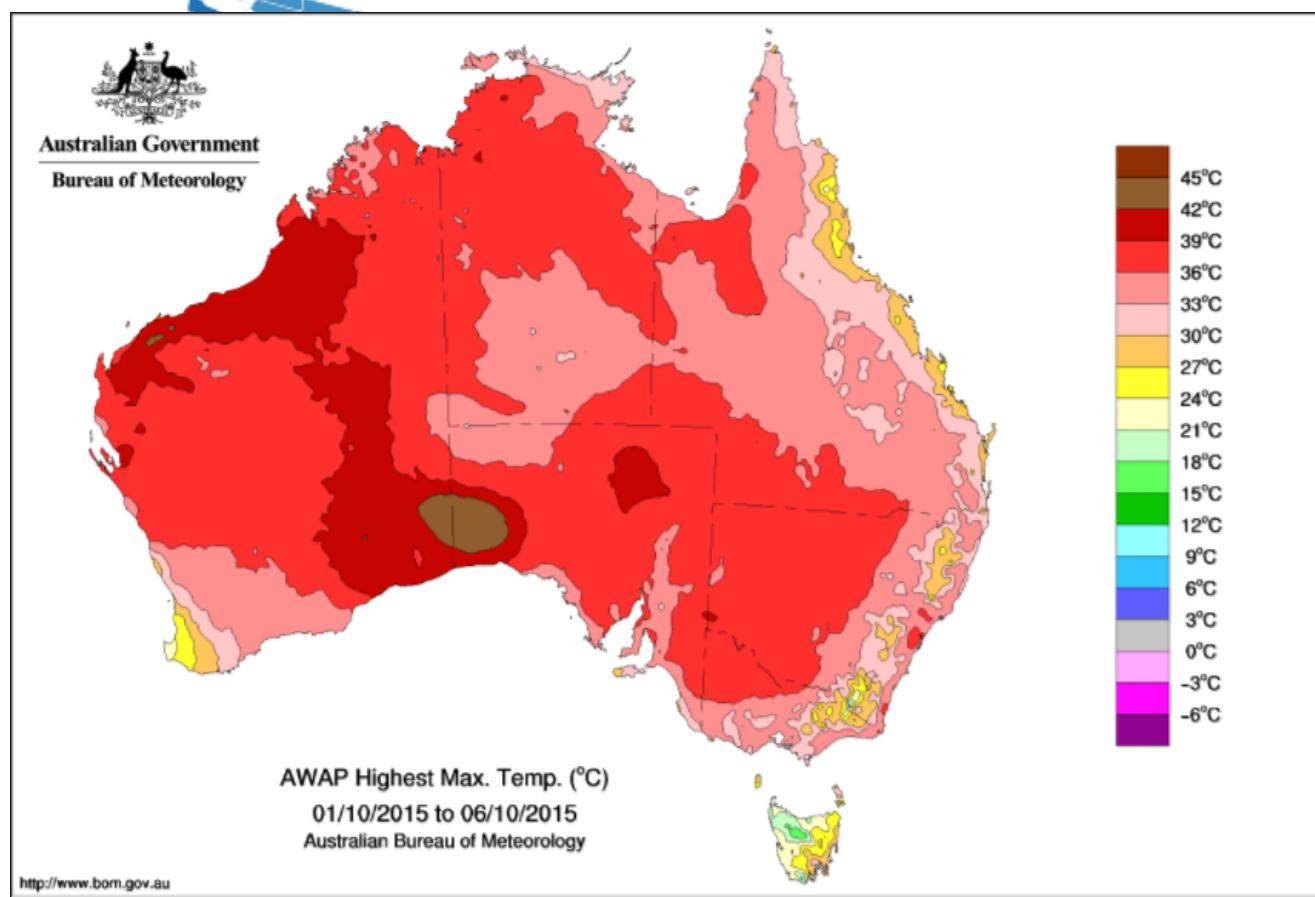
DISCRETE/QUALITATIVE DATA



ORDINAL DATA

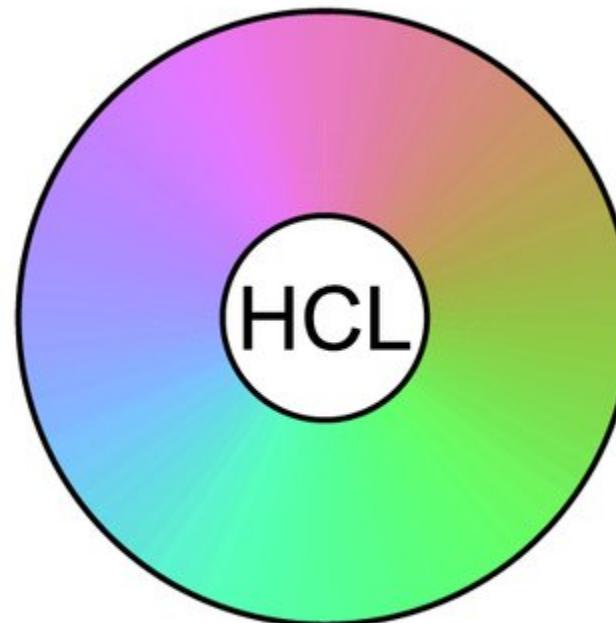


CONTINUOUS/QUANTITATIVE DATA



ADD COLOR THE THE HARD WAY

- Control *Hue, Chroma, Luminance* manually
- ```
+ scale_fill_hue(
 h=c(0, 360),
 c=50,
 l=50)
```

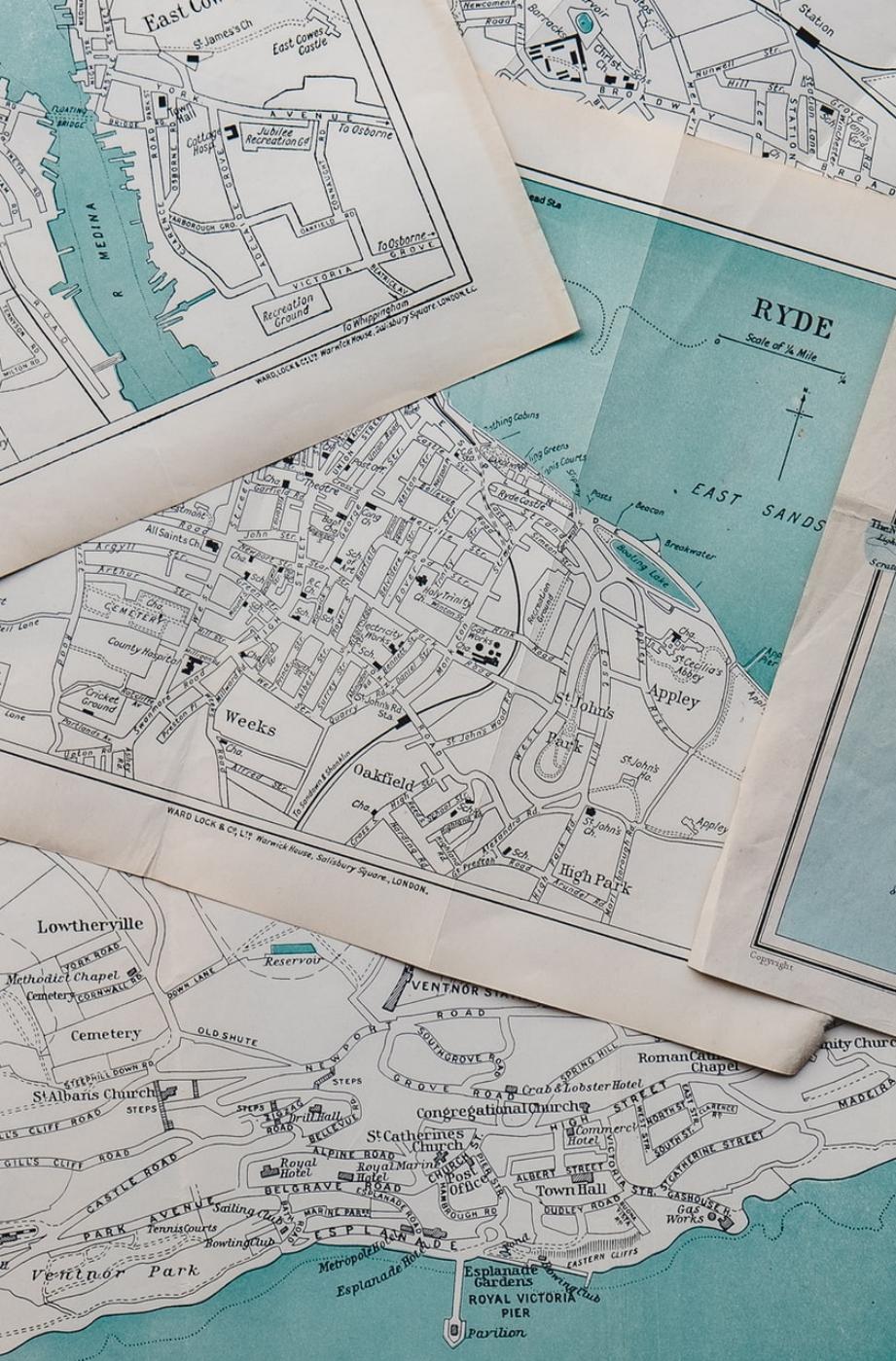


# THE TOO-BASIC WAY

- Use built-in base colors
- **+ scale\_fill\_manual(values = c('color1', 'color2'))**  
or `scale_color_manual` for points and lines  
can also use hex colors (e.g., `#FF00FF`) here
- `demo('colors')` to see options

# THE RCOLORBREWER WAY

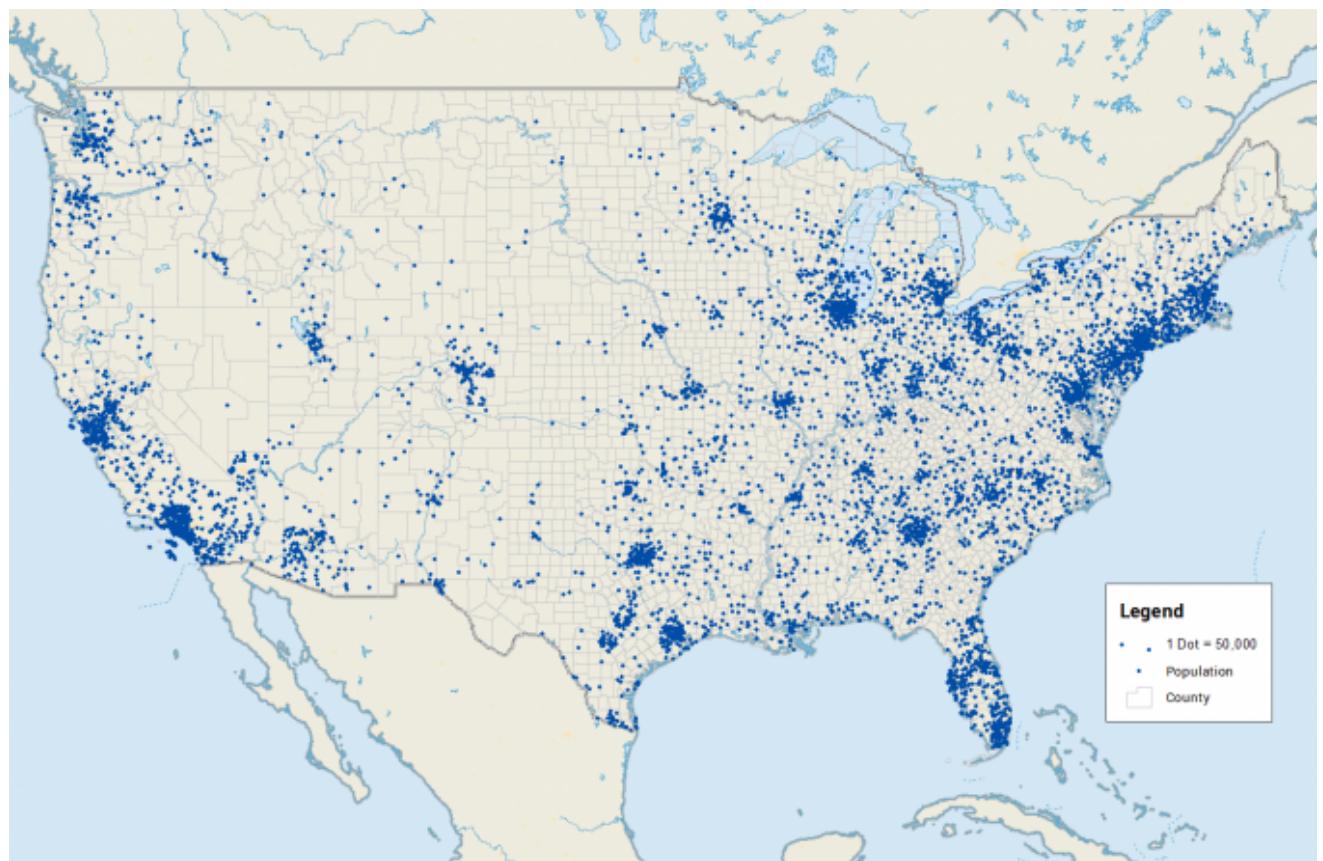
- Pre-selected palettes: **display.brewer.all()**
  - Map onto data types
    - qualitative
    - ordinal/quantitative
    - divergent
  - Access with **scale\_fill\_brewer(type="seq/div/qual", palette)**  
or **scale\_color\_brewer** for points and lines
  - Extract hex with **brewer.pal(n, "xyz")**  
add **display.** to beginning to see the colors
- number of colors  
from palette*



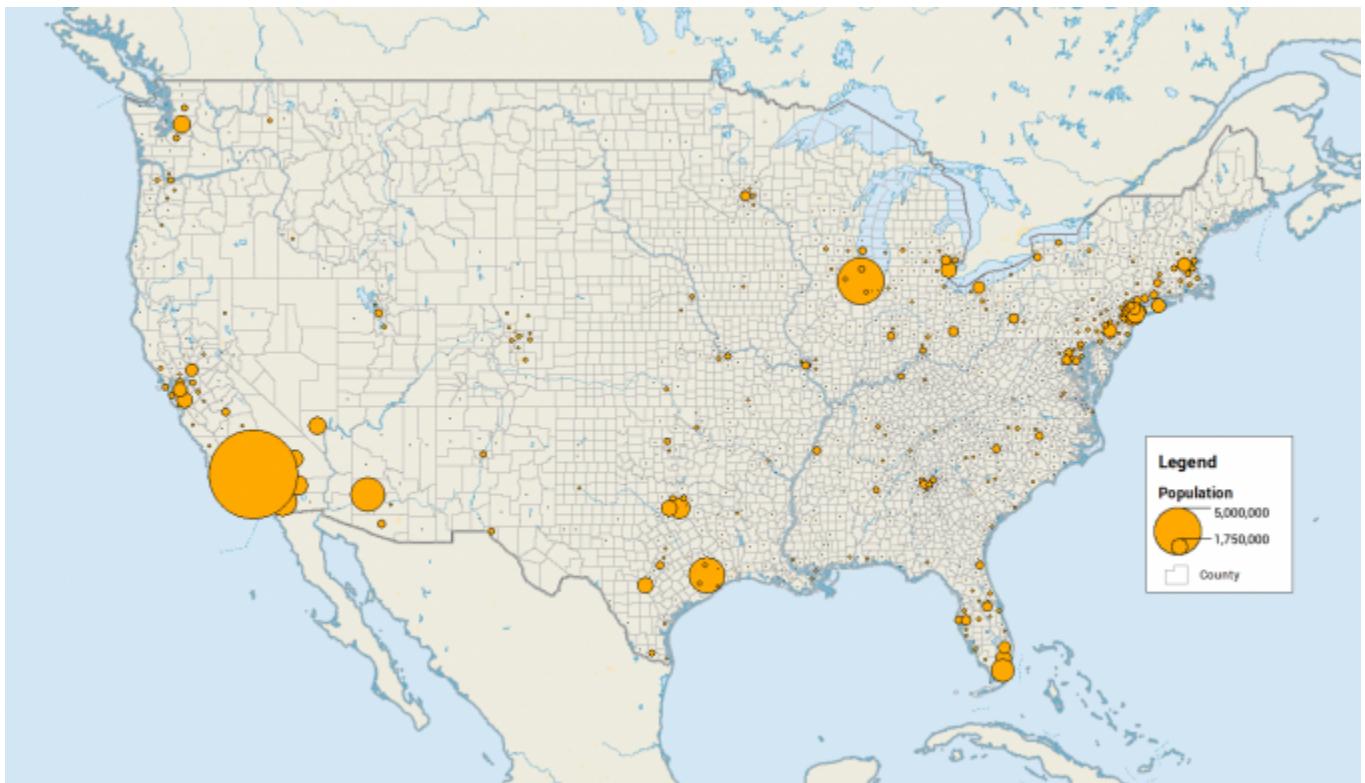
# MAPS

---

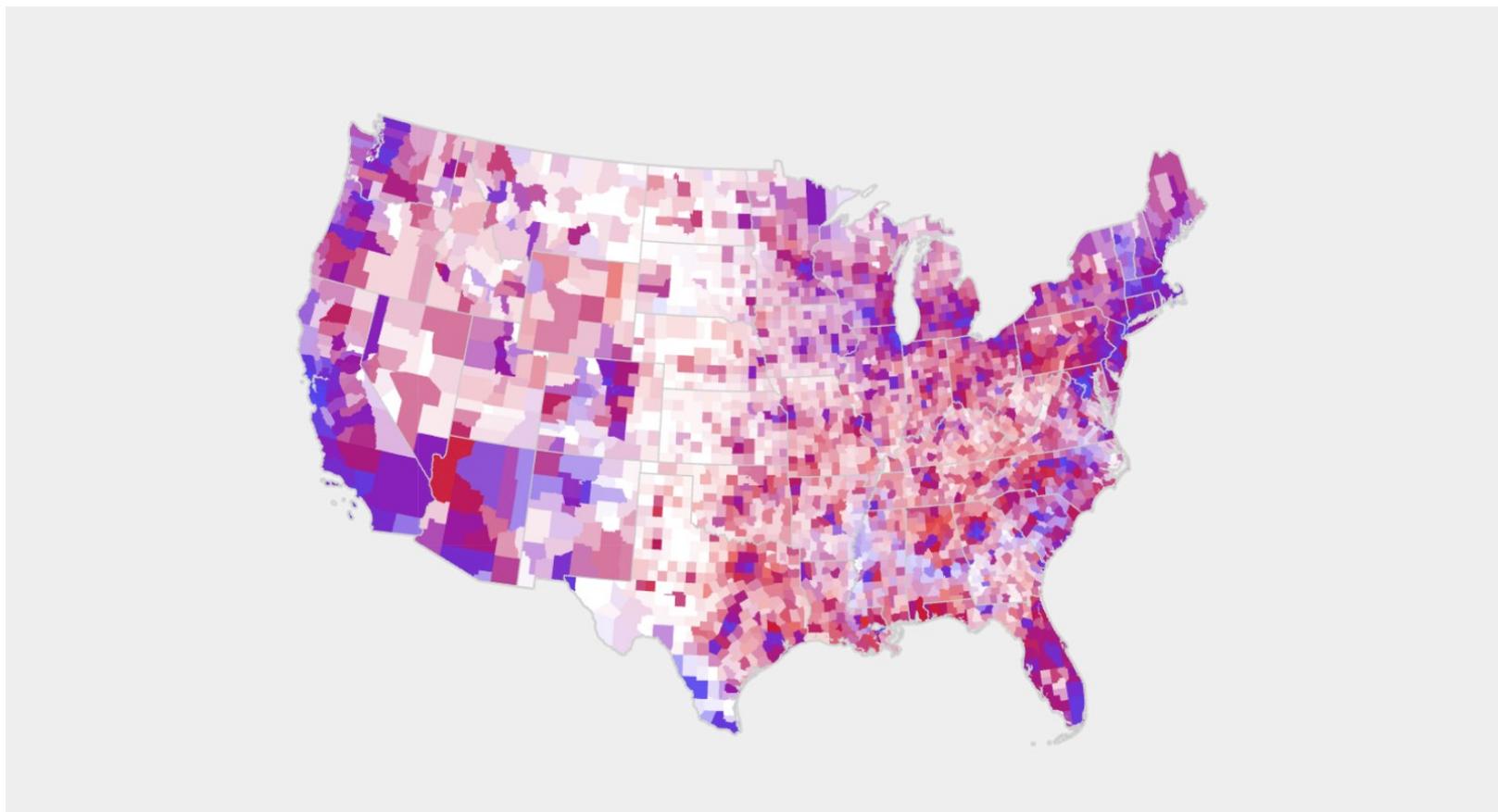
# DOT DISTRIBUTION



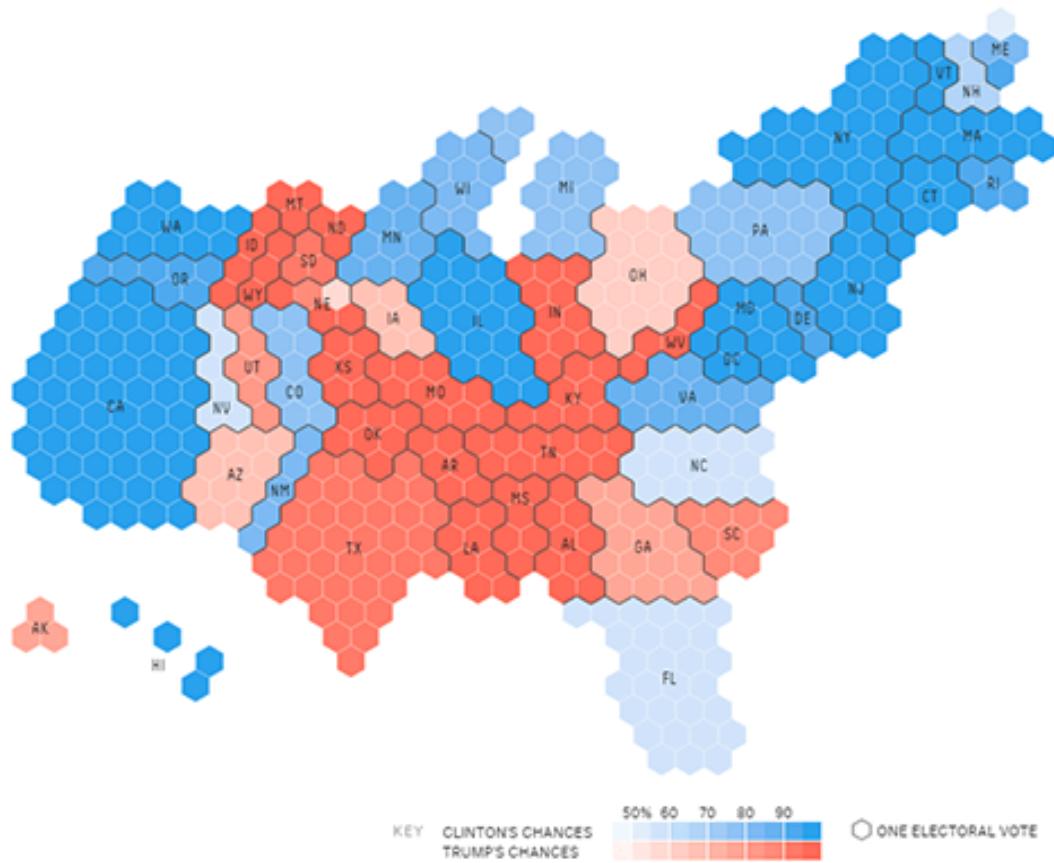
# GRADUATED SYMBOL



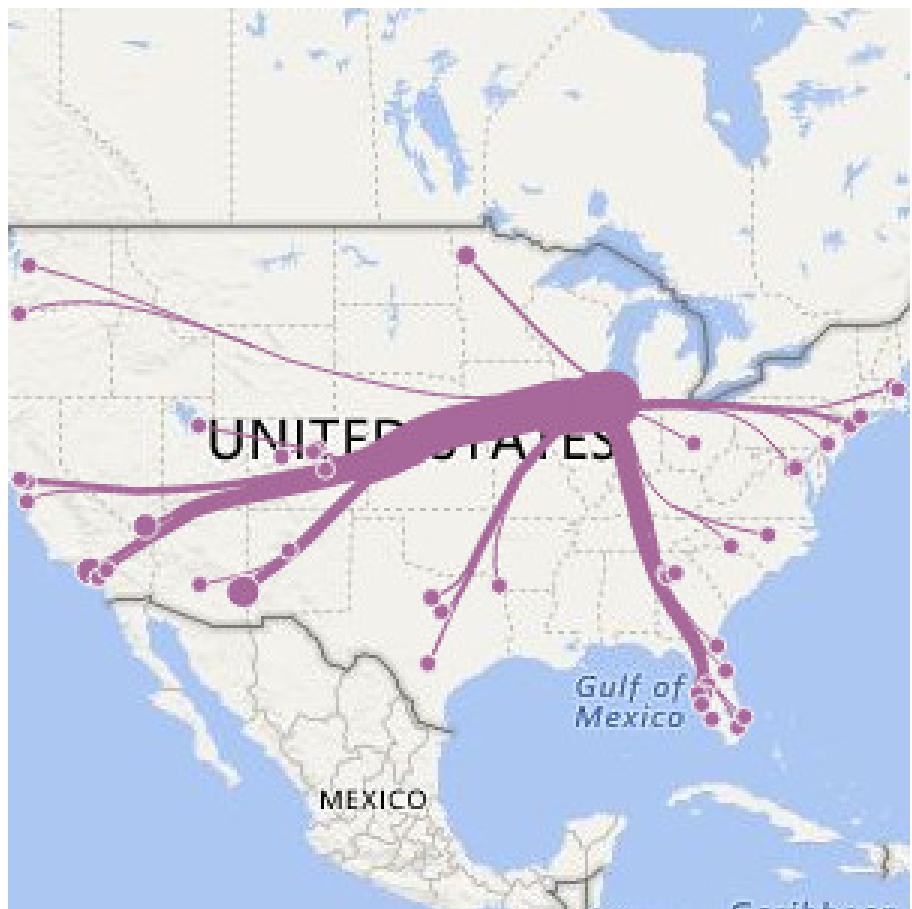
# CHOROPLETH



# CARTOGRAM

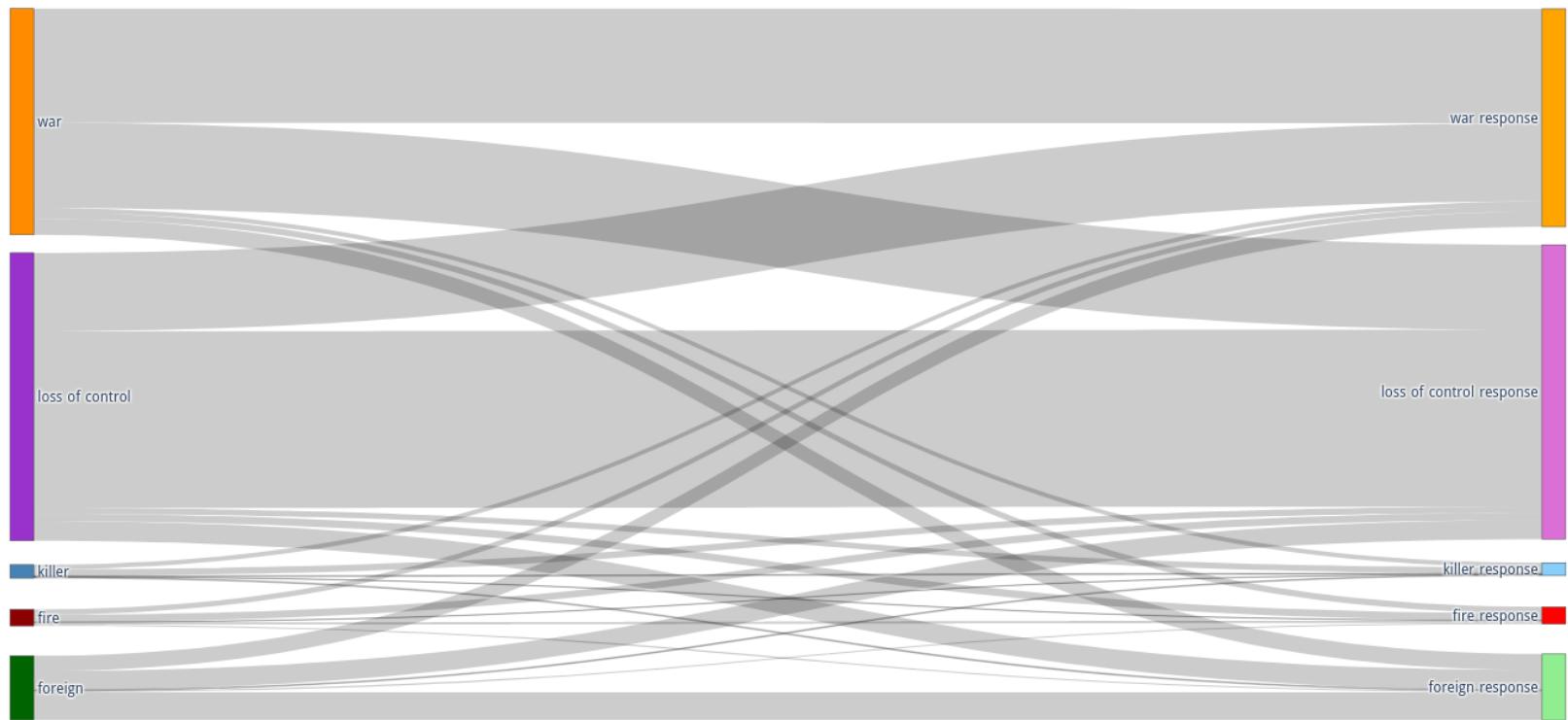


# FLOW/NETWORK MAPS



# NETWORKS

---



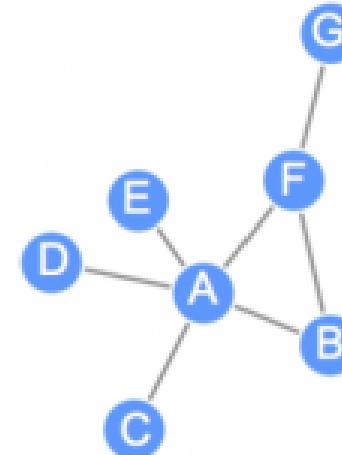
# EDGE LISTS

| <b>source</b> | <b>target</b> | <b>opt. variables (# of flights, delay, etc.)</b> |
|---------------|---------------|---------------------------------------------------|
| JFK           | LAX           | 18                                                |
| ORD           | FAR           | 5                                                 |
| MSP           | SEA           | 21                                                |
| FAR           | DEN           | 1                                                 |
| LAX           | SFO           | 19                                                |

# Undirected

## MATRICES

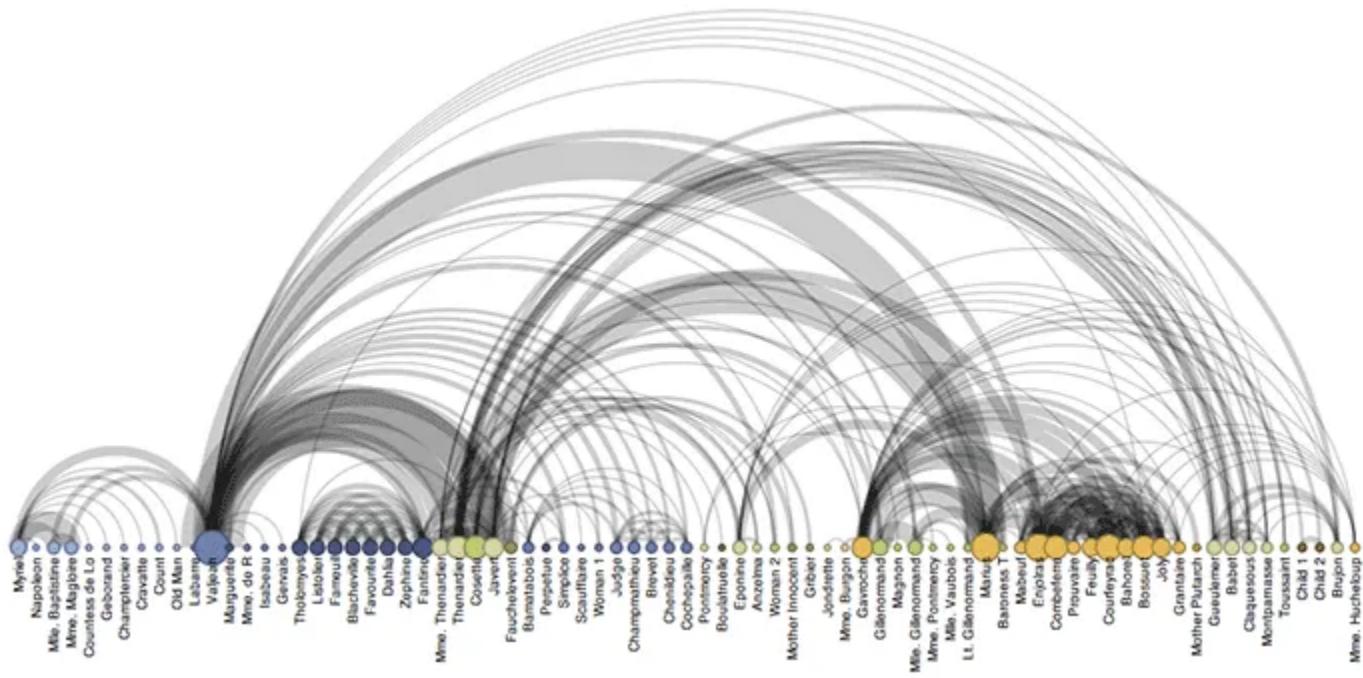
Binary representation of linkages



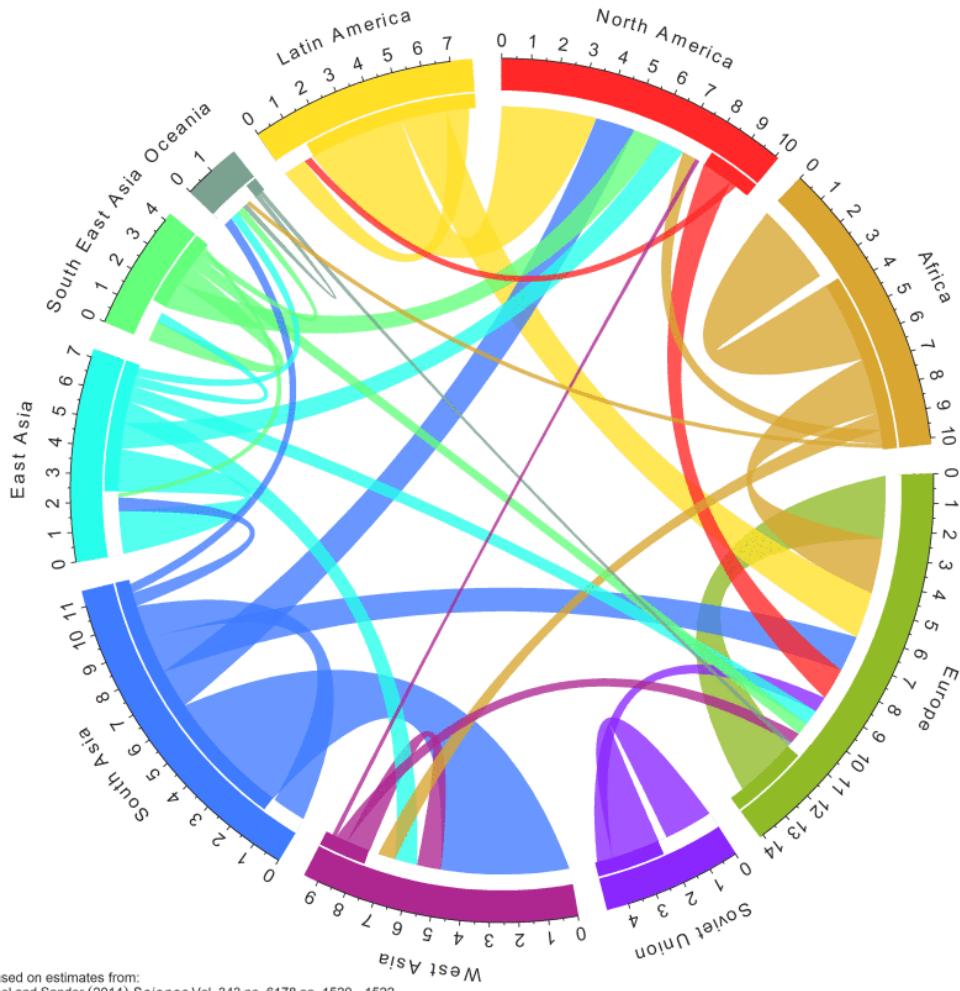
|   | A | B | C | D | E | F | G | Degree |
|---|---|---|---|---|---|---|---|--------|
| A | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 5      |
| B | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2      |
| C | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1      |
| D | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1      |
| E | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1      |
| F | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 3      |
| G | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1      |

Adjacency matrices

# ARCH



# CIRCULAR



# SANKEY