005 - Data Graphics

EPIB 607 - FALL 2020

Sahir Rai Bhatnagar
Department of Epidemiology, Biostatistics, and Occupational Health
McGill University

sahir.bhatnagar@mcgill.ca

slides compiled on September 11, 2020

Objective

- Understand the building blocks of visualizing data

Visualizing data: Mapping data onto aesthetics

Scales

Color scales

What is Data Visualization?

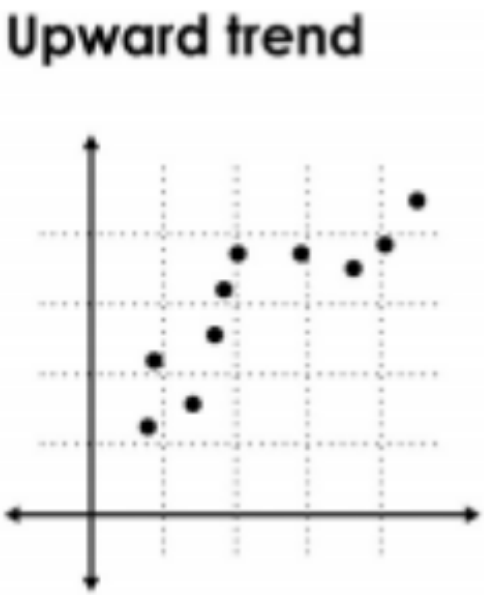
- In its most basic form, visualization is simply mapping data to geometry and color.
- It works because your brain is wired to find patterns, and you can switch back and forth between the visual and the numbers it represents.
- This is the important bit. You must make sure that the essence of the data isn't lost in that back and forth between visual and the value it represents because if you can't map back to the data, the visualization is just a bunch of shapes.

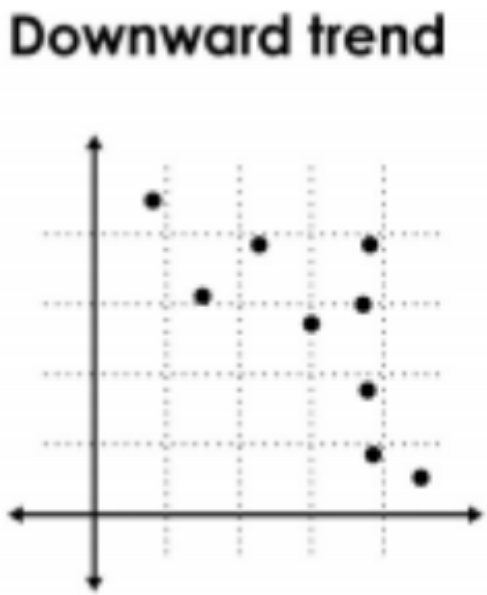
Aesthetics (aka Visual Cues)

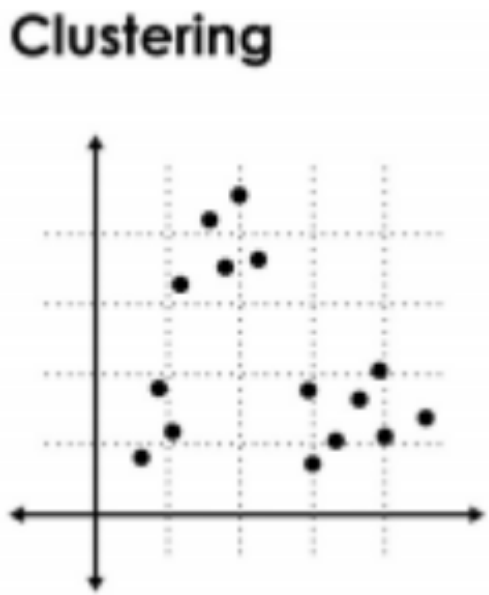
- All data visualizations map data values into quantifiable features of the resulting graphic.
- We refer to these features as aesthetics, also known as Visual Cues.

Example: Scatterplot

- When you use position as a visual cue, you compare values based on where others are placed in a given space or coordinate system







Aesthetics (Visual Cues): The Building Blocks

1. Position (numerical): where in relation to other things?
2. Length (numerical): how big (in one dimension)?
3. Angle (numerical): how wide? parallel to something else?
4. Direction (numerical) at what slope? In a time series, going up or down?
5. Shape (categorical) belonging to which group?
6. Area (numerical) how big (in two dimensions)?
7. Volume (numerical) how big (in three dimensions)?
8. Shade (either) to what extent? how severely?
9. Color (either) to what extent? how severely? Beware of red/green color blindness

Visual Cues: When you visualize data, you encode values to shapes, sizes, and colors

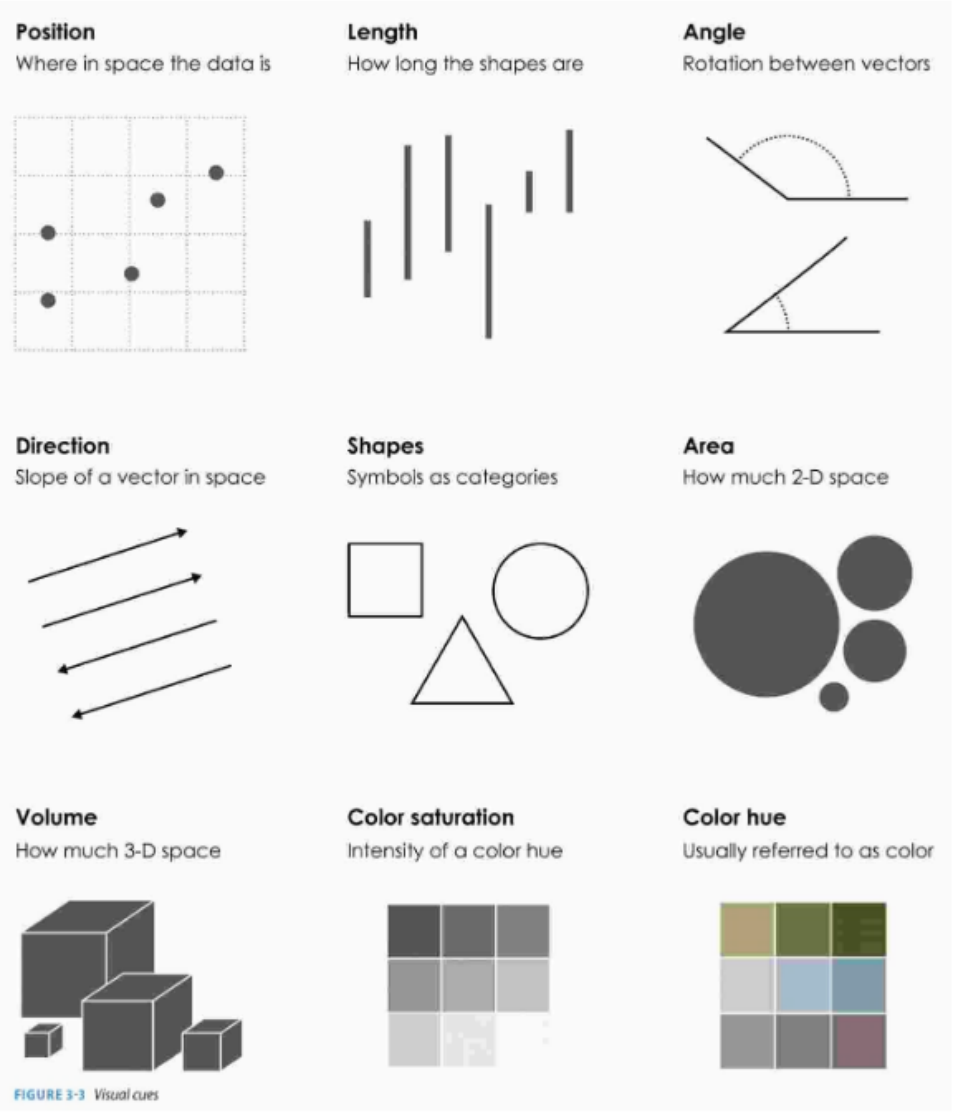
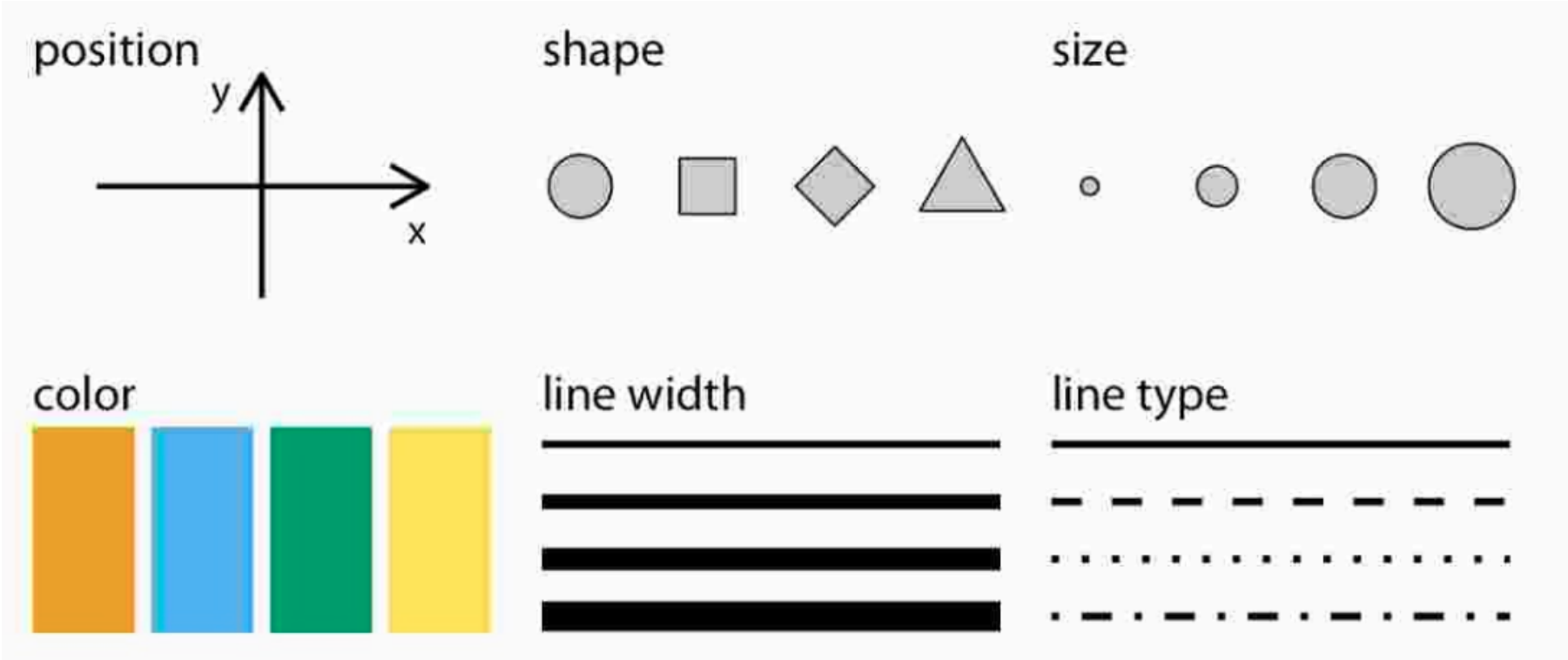


FIGURE 3-3 *Visual cues*

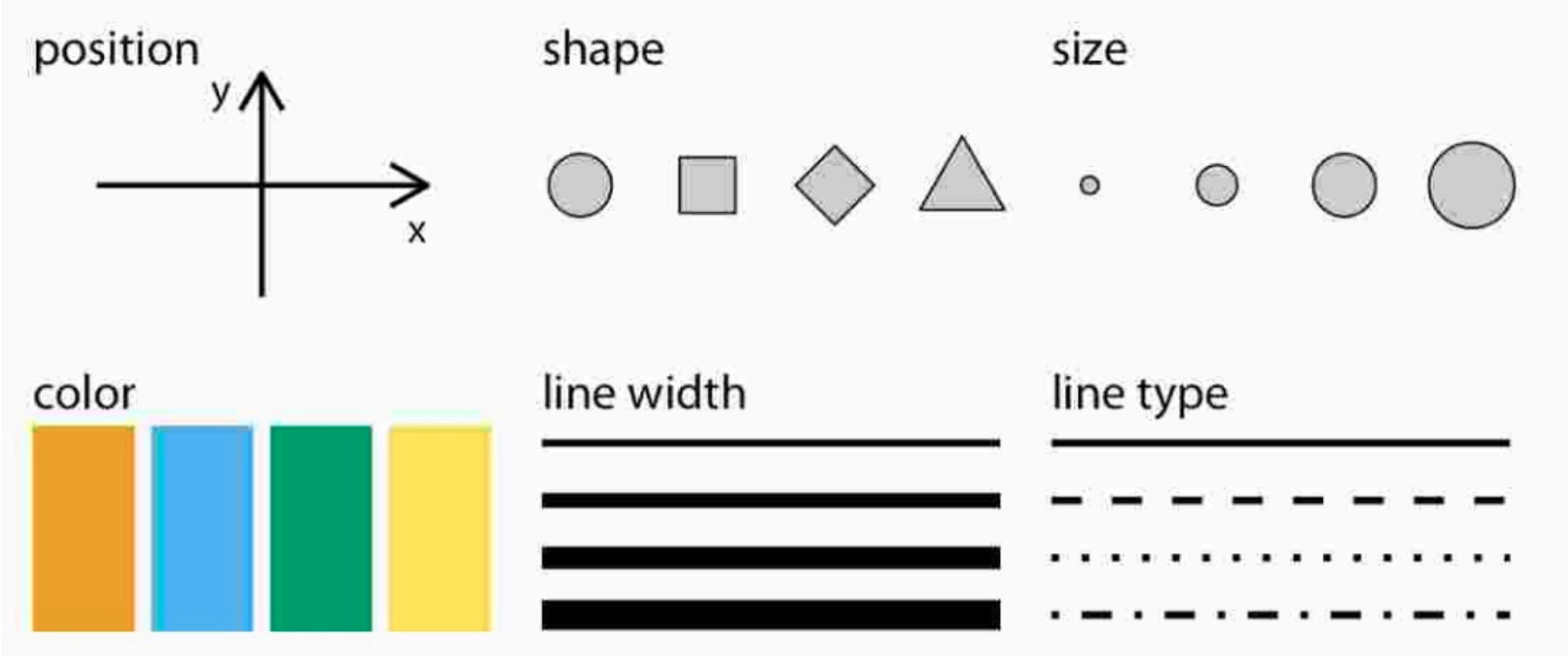
Commonly Used Visual Cues



All visual cues fall into one of two groups

- Those that can represent continuous data and those that can not

Which of the following can represent continuous data? Discrete data?



Visualizing data: Mapping data onto aesthetics

Scales

Color scales

Scales

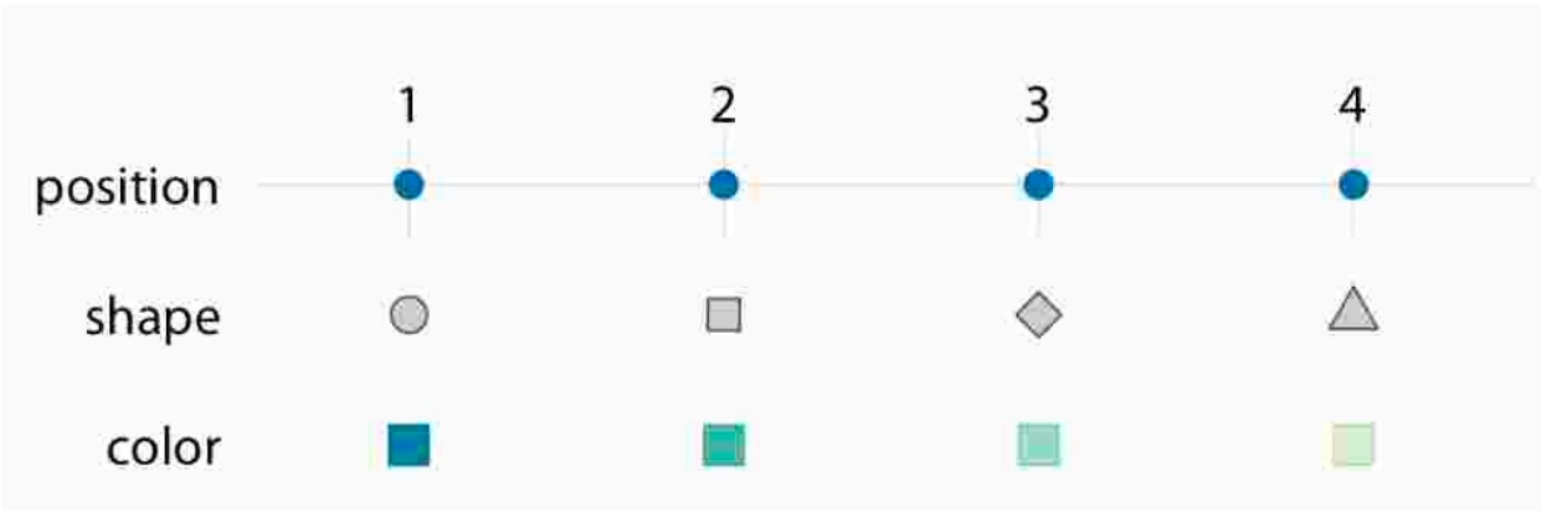
- To **map** data values onto **aesthetics**, we need to specify which data values correspond to which specific aesthetics values.
- For example, if our graphic has an x axis, then we need to specify which data values fall onto particular positions along this axis.
- Similarly, we may need to specify which data values are represented by particular shapes or colors.

Scales

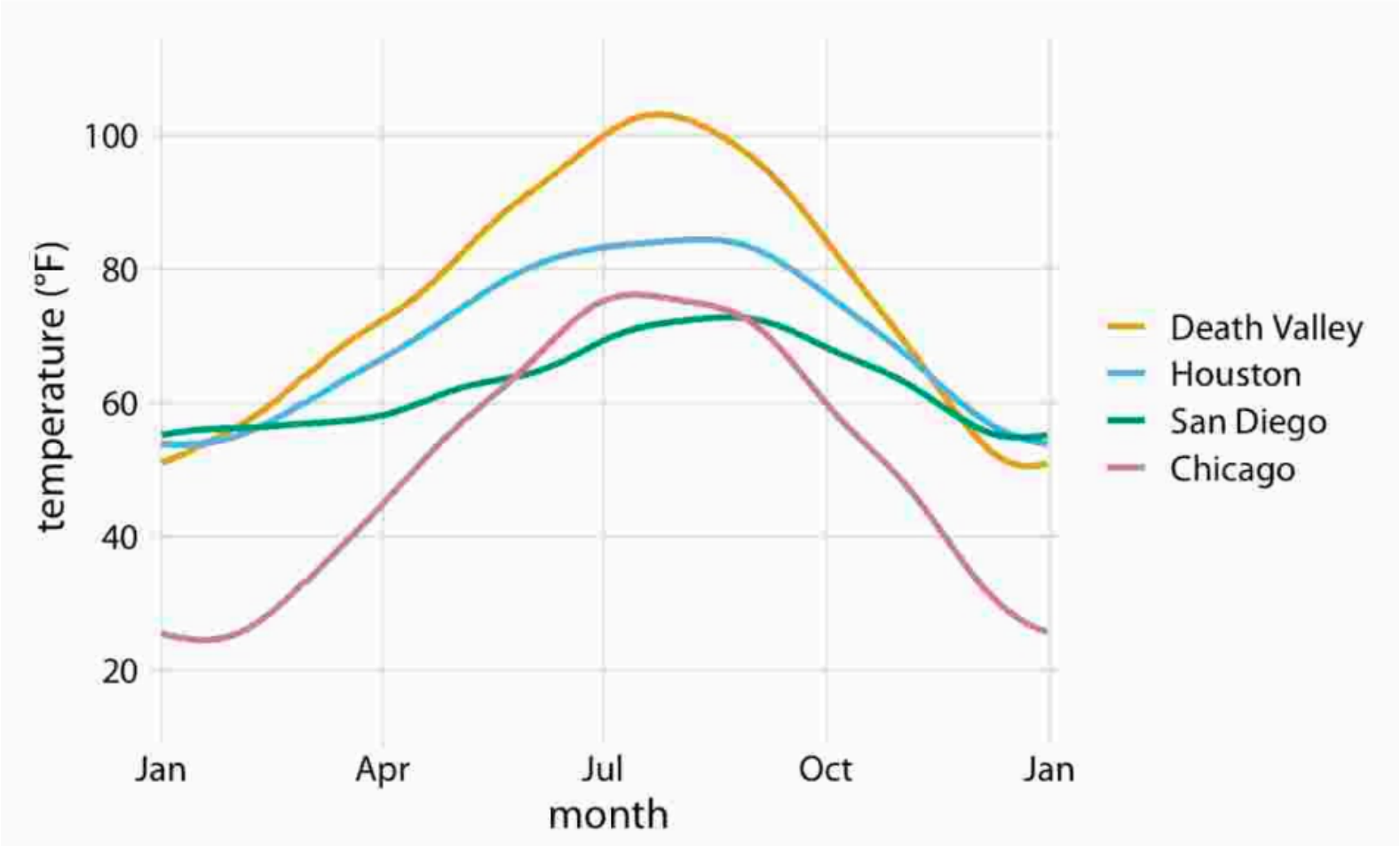
- This mapping between data values and aesthetics values is created via scales.
- A scale defines a unique mapping between data and aesthetics.
- Importantly, **a scale must be one-to-one**, such that for each specific data value there is exactly one aesthetics value and vice versa.
- If a scale isn't one-to-one, then the data visualization becomes ambiguous.

Scales

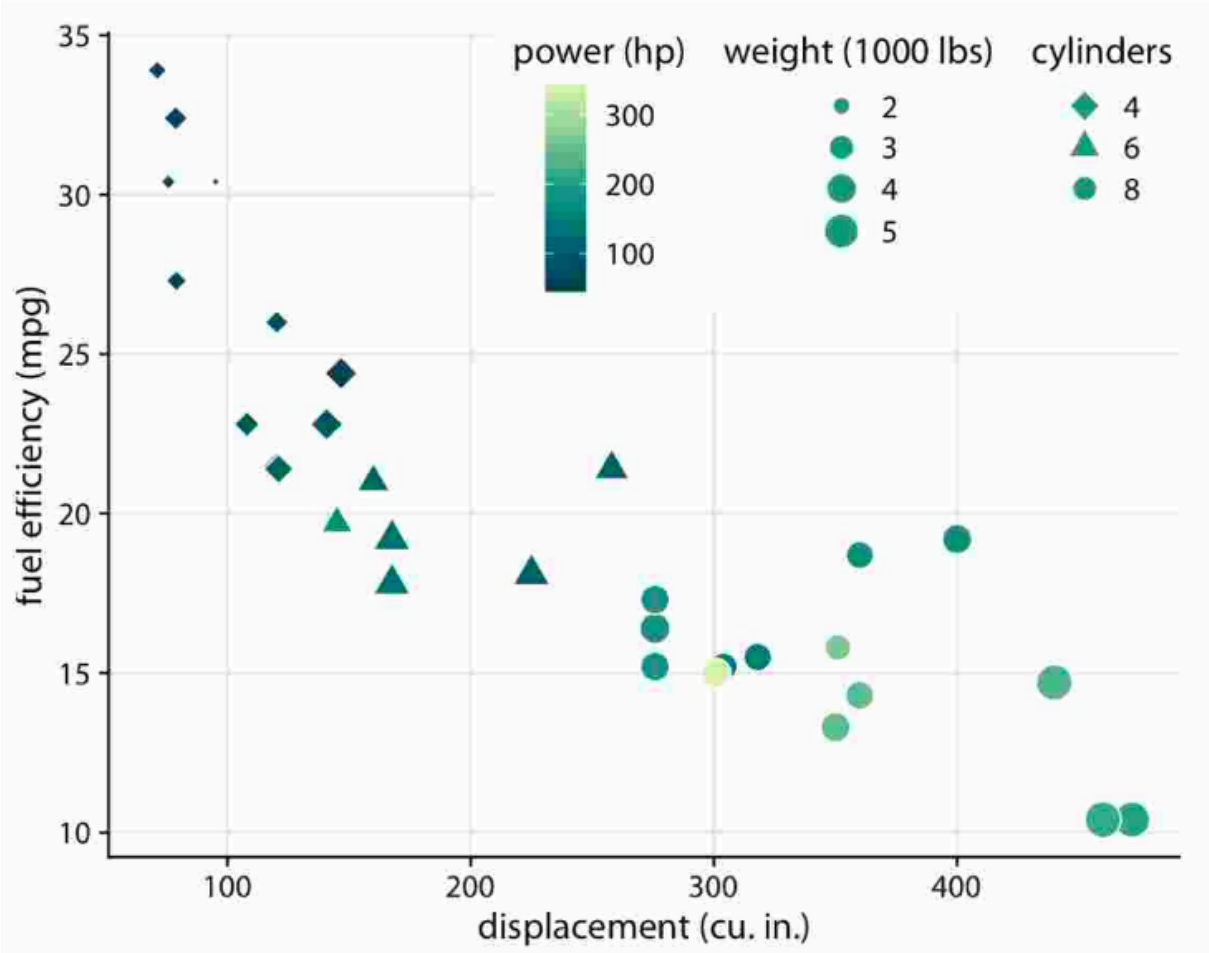
- Scales link data values to aesthetics.
- Here, the numbers 1 through 4 have been mapped onto a position scale, a shape scale, and a color scale.
- For each scale, each number corresponds to a unique position, shape, or color and vice versa



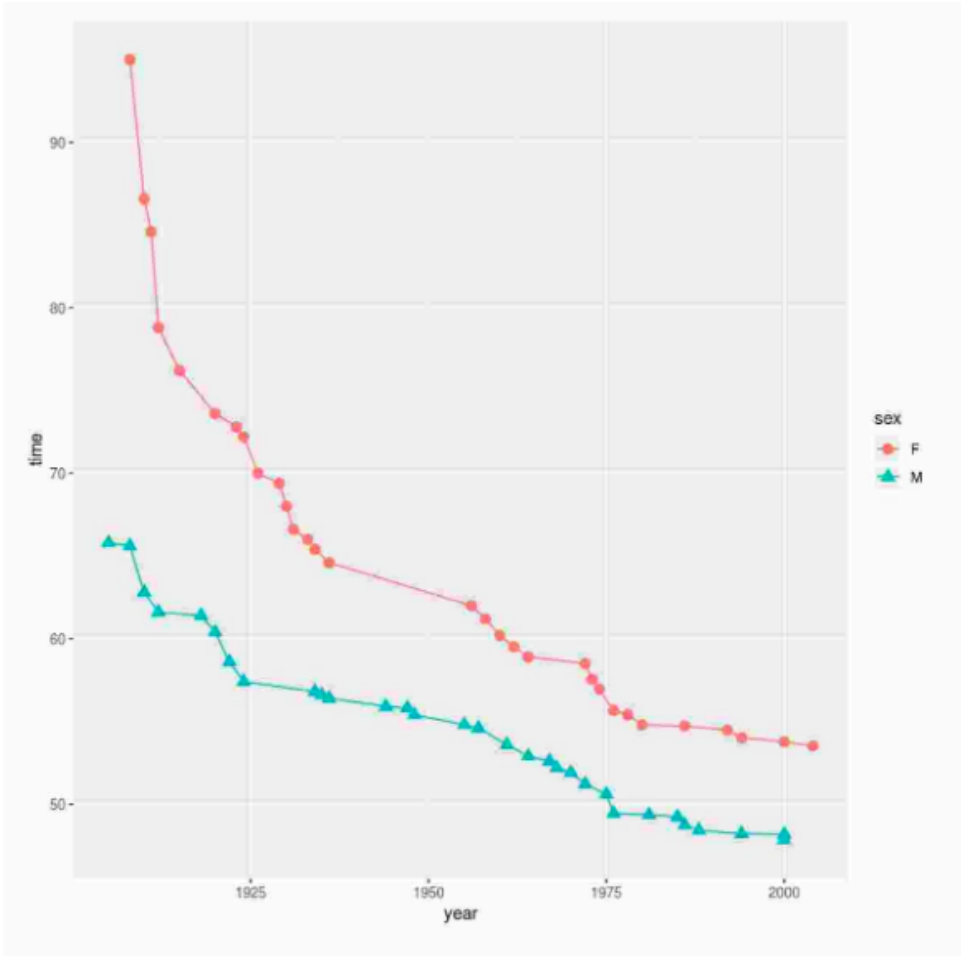
How many scales are being used?



How many scales are being used?



How many scales are being used?



Difference between Aesthetics (Visual Cues) and Scales ?

- **Aesthetics:** describe every aspect of a given graphical element.
- **Scale:** defines a unique mapping between data and aesthetics.
- A scale is a visual cue with data attached to it

Visualizing data: Mapping data onto aesthetics

Scales

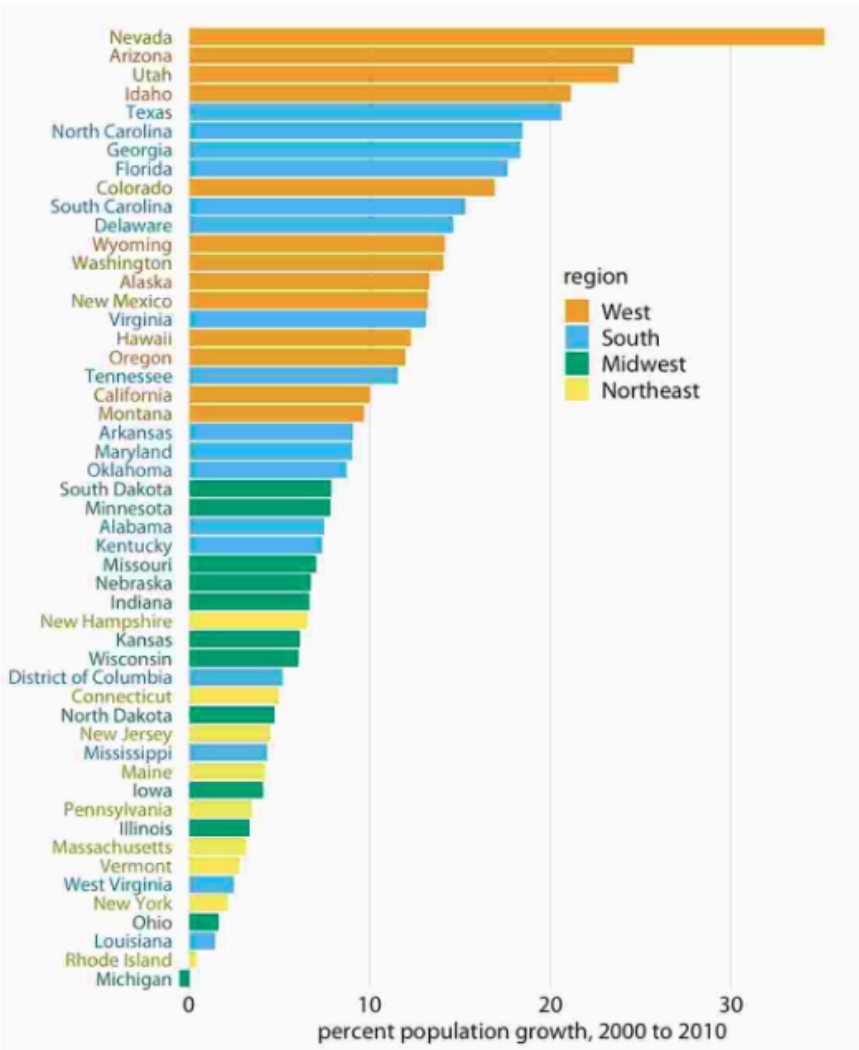
Color scales

Color scales: 3 use cases

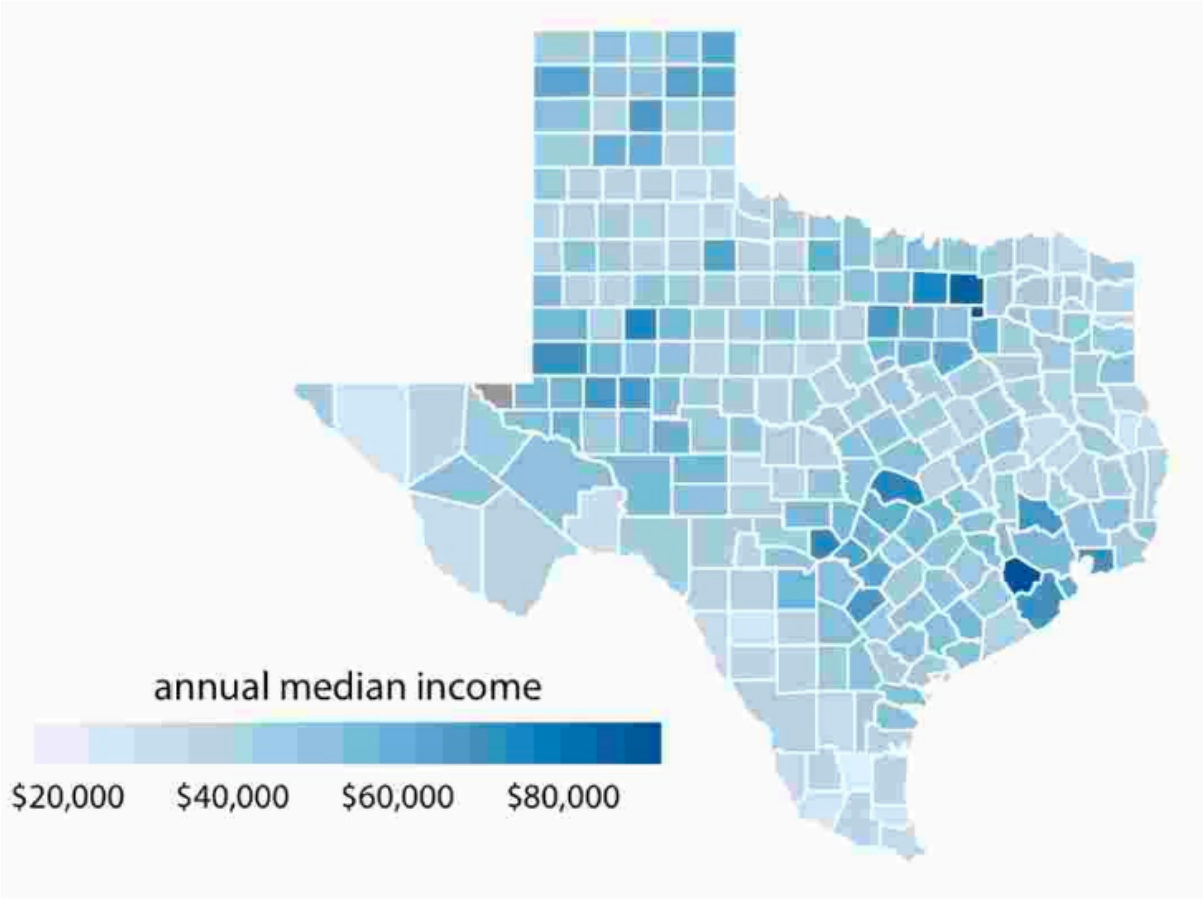
1. To distinguish groups of data from each other
2. Represent data values
3. To highlight

The types of colors we use and the way in which we use them are quite different for these three cases.

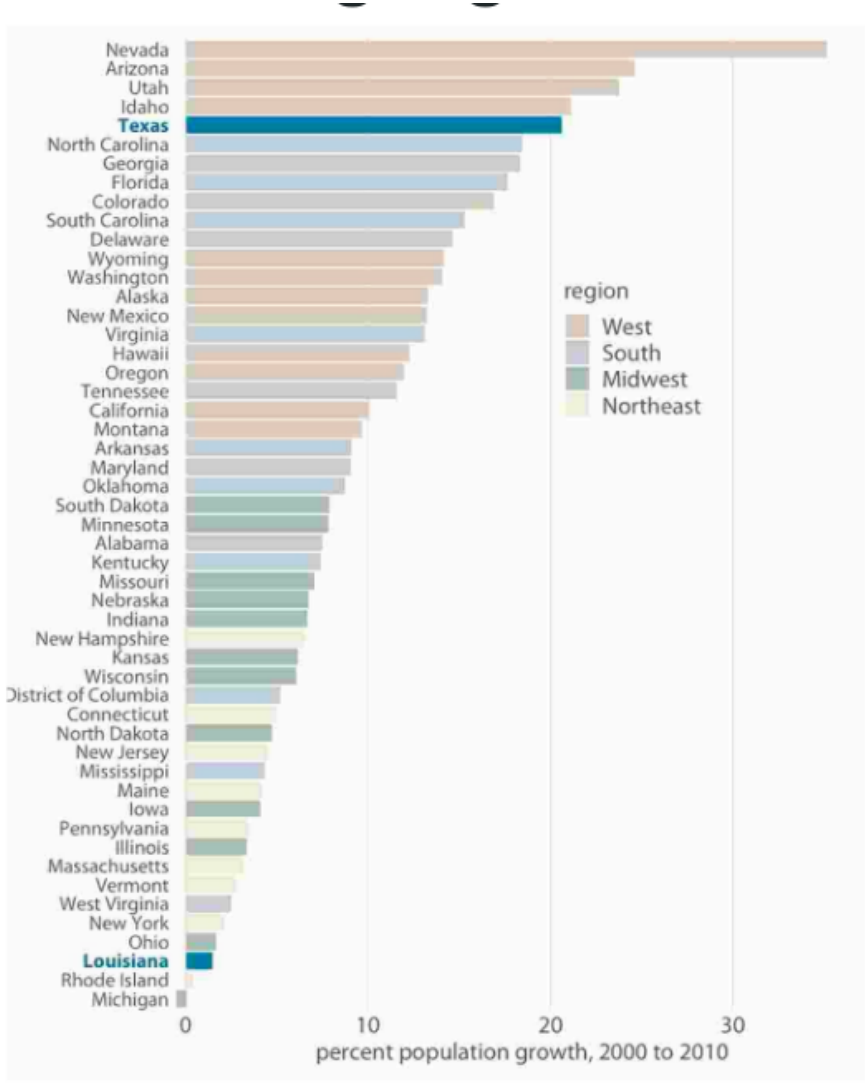
Color as a tool to distinguish



Color to represent values



Color as a tool to highlight

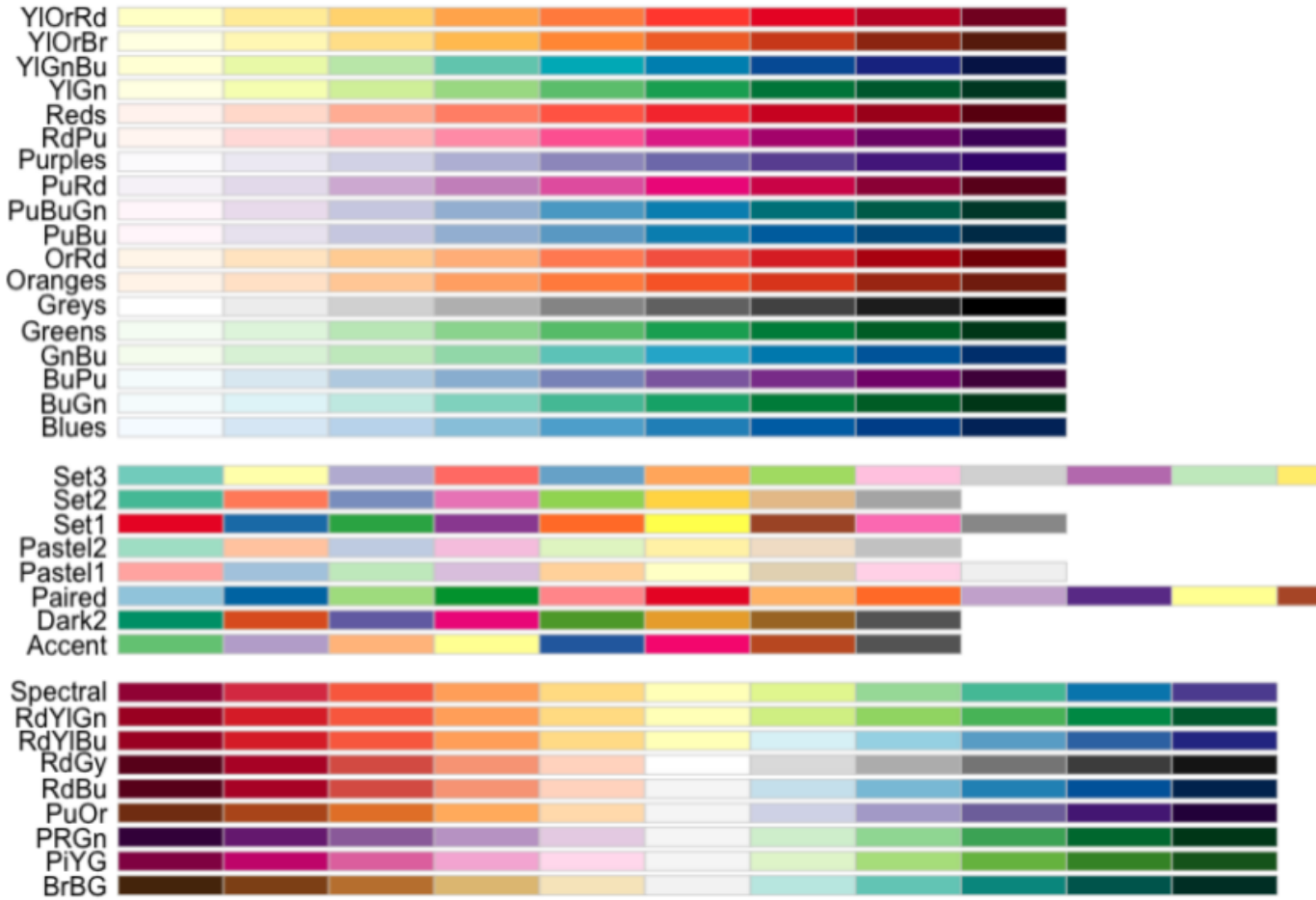


Cynthia Brewer Palette

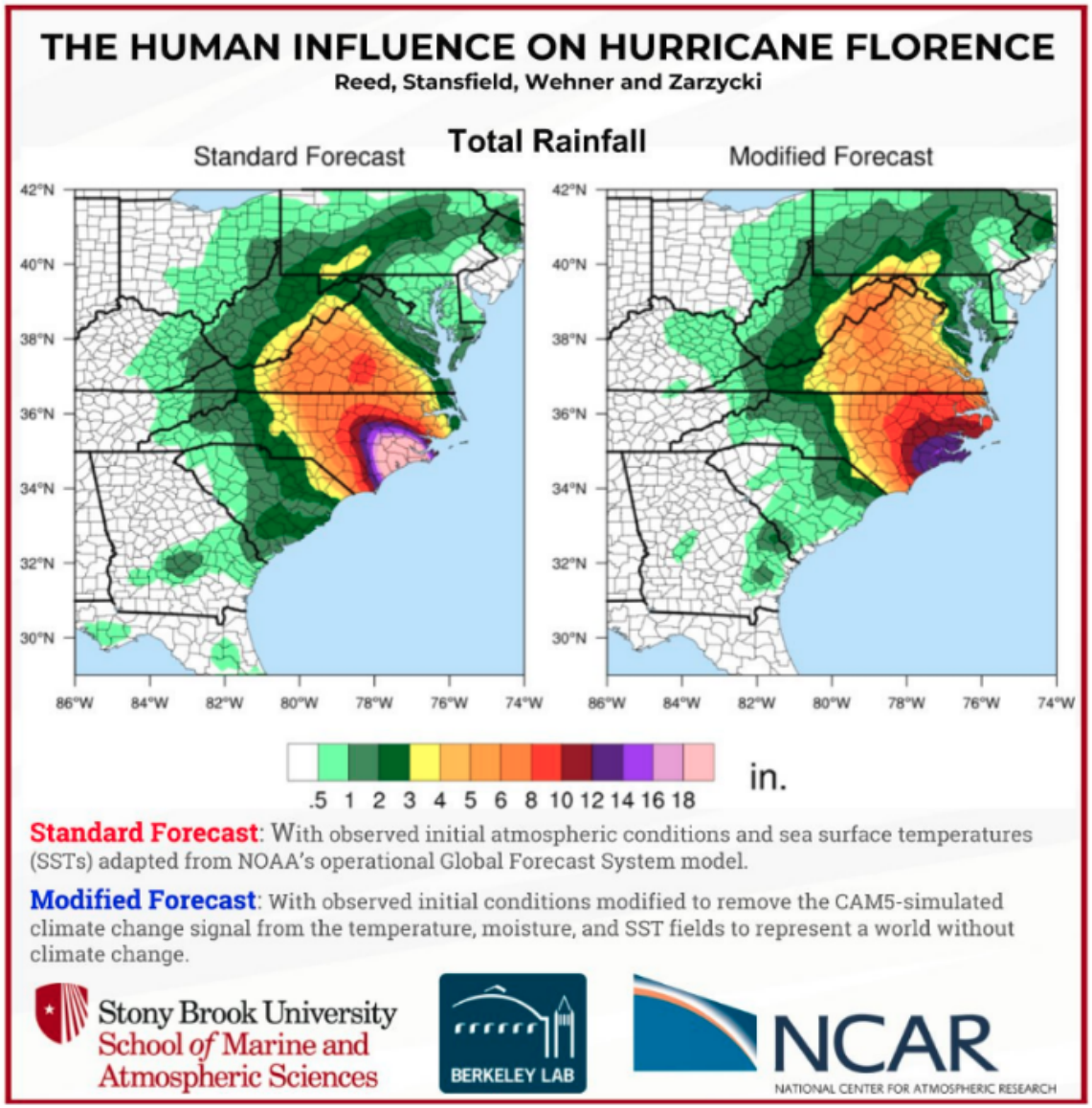
Sequential
(data values)

Qualitative
(distinguish)

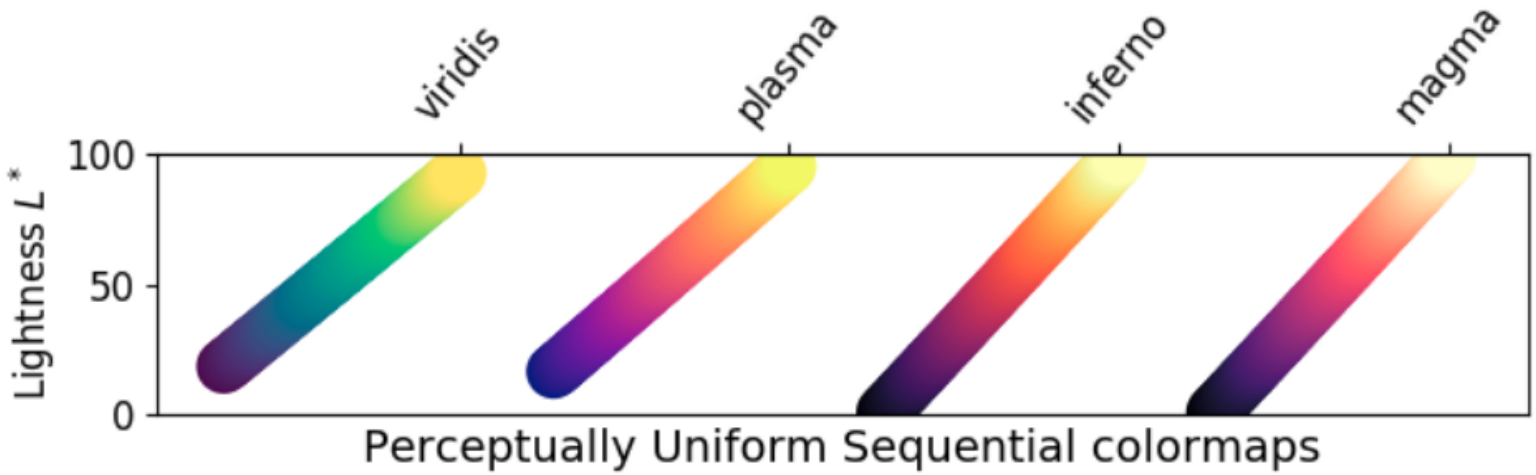
Diverging
(two directions)



Good choice of colors?



Perceptually Uniform Palettes

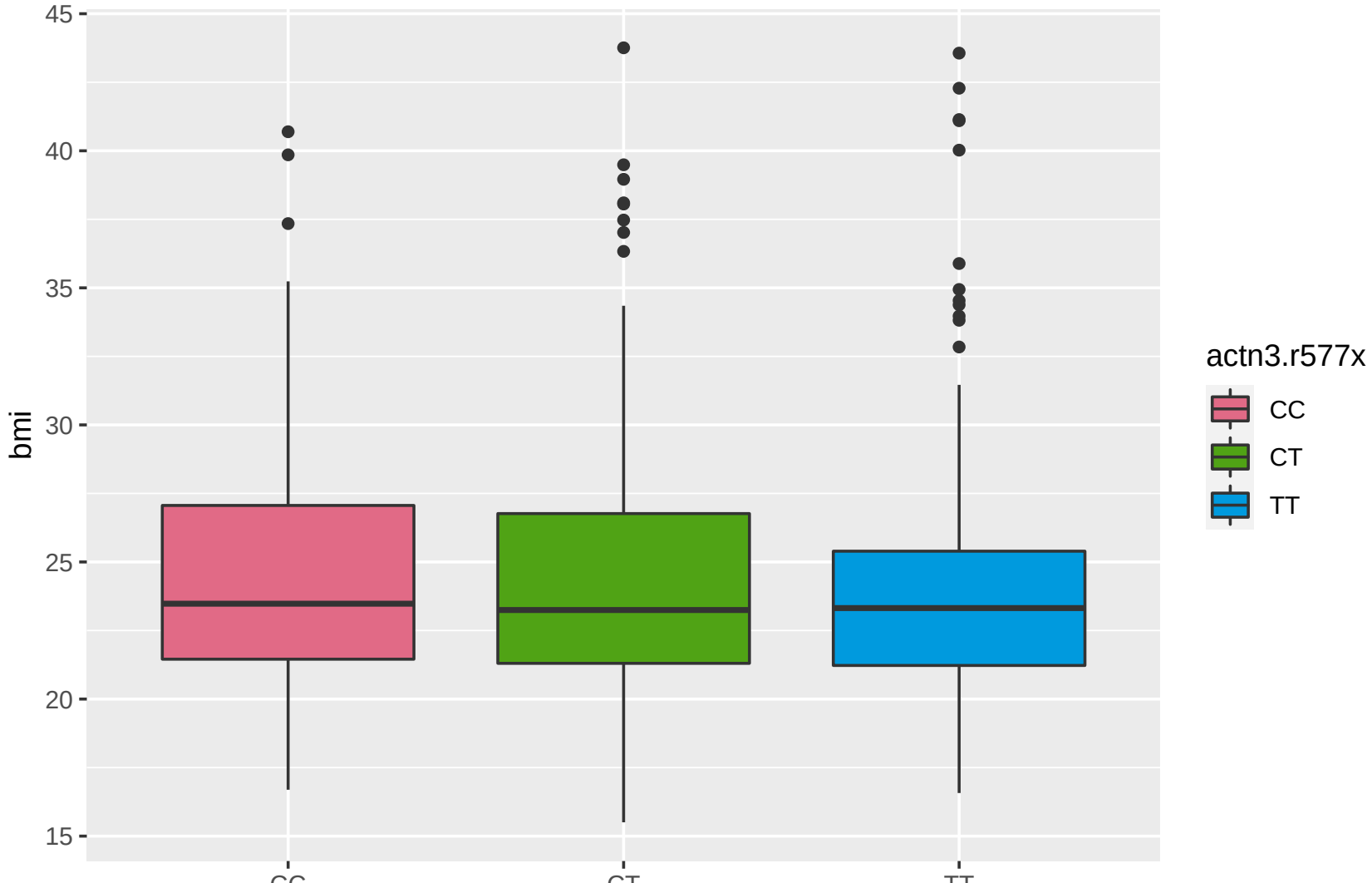


- https://cran.r-project.org/web/packages/colorspace/vignettes/colorspace.html
- https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

Qualitative palette

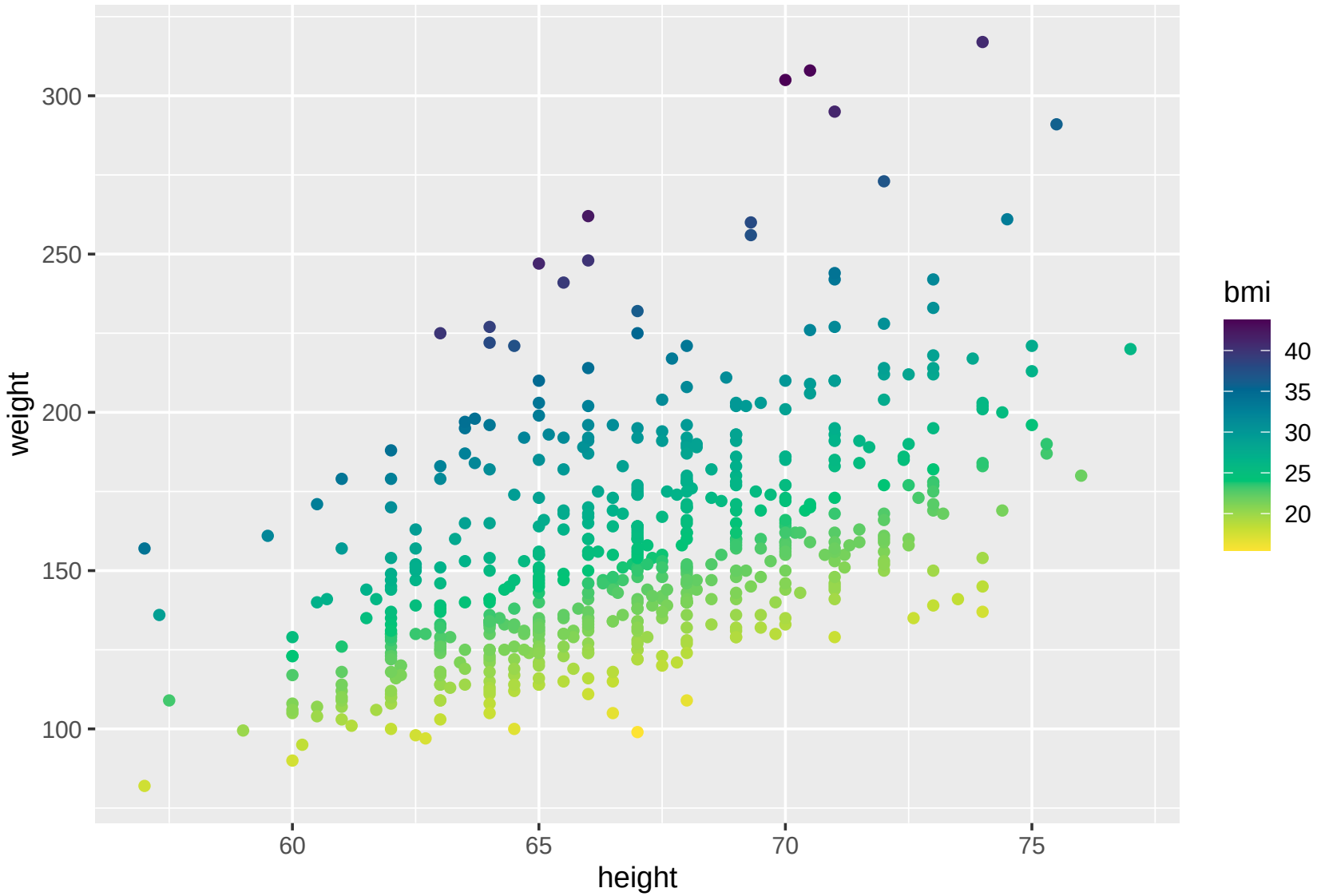
```
library(oibiostat); data("famuss")
library(ggplot2)
library(colorspace)

ggplot(famuss, aes(x = actn3.r577x, y = bmi, fill = actn3.r577x)) +
  geom_boxplot() +
  colorspace::scale_fill_discrete_qualitative()
```



Sequential palette

```
ggplot(famuss, aes(x = height, y = weight, color = bmi)) +
  geom_point() +
  colorspace::scale_color_continuous_sequential(palette = "Viridis")
```



Session Info

```
R version 3.6.2 (2019-12-12)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Pop!_OS 19.10

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.3.7.so

attached base packages:
[1] tools     stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] colorspace_1.4-1   oibiostat_0.2.0    NCStats_0.4.7      FSA_0.8.30
 [5] forcats_0.5.0      stringr_1.4.0      dplyr_1.0.2        purrr_0.3.4
 [9] readr_1.3.1        tidyr_1.1.2        tibble_3.0.3       ggplot2_3.3.2.9000
[13] tidyverse_1.3.0    knitr_1.29

loaded via a namespace (and not attached):
 [1] tidyselect_1.1.0   xfun_0.16          haven_2.3.1        vctrs_0.3.4
 [5] generics_0.0.2     rlang_0.4.7        pillar_1.4.6       glue_1.4.2
 [9] withr_2.2.0        DBI_1.1.0          dbplyr_1.4.2       modelr_0.1.5
[13] readxl_1.3.1       lifecycle_0.2.0    plyr_1.8.6         munsell_0.5.0
[17] gtable_0.3.0       cellranger_1.1.0   rvest_0.3.5        evaluate_0.14
[21] labeling_0.3       fansi_0.4.1        highr_0.8          broom_0.7.0
[25] Rcpp_1.0.4.6       scales_1.1.1       backports_1.1.9    jsonlite_1.7.0
[29] farver_2.0.3       fs_1.3.2           TeachingDemos_2.12 hms_0.5.3
[33] digest_0.6.25      stringi_1.4.6      grid_3.6.2         cli_2.0.2
[37] magrittr_1.5       crayon_1.3.4       pkgconfig_2.0.3    ellipsis_0.3.1
[41] xml2_1.3.0         reprex_0.3.0       lubridate_1.7.4    assertthat_0.2.1
[45] httr_1.4.1         rstudioapi_0.11    R6_2.4.1           compiler_3.6.2
```