Prelim

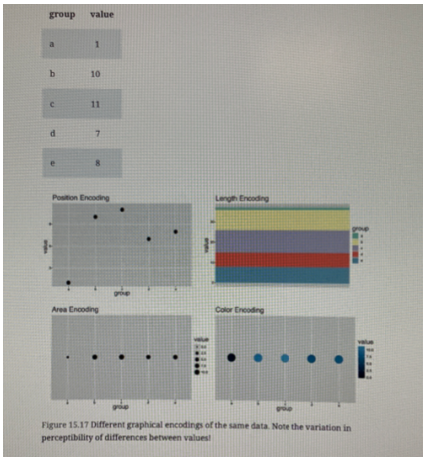
- 15.2 - Selecting visualizations

- The constraints:
  1. Question of interest
  2. The type of data we are trying to visualize
  3. limitations of the human visual processing system
  4. The spacial limitations

- The different visualizations & the questions they answer:
  - bar chart - can be used to label the top 10 of something using a quantity
  - histogram - for continuous vars, visualize the distribution & range of values
  - proportional Representations - show each value relative to the total of that column.
    - Stacked Bar Chart
    - pie chart
  - If your variable of interest is categorical, you will need to aggregate your data eg. number of occurrences/category)
  - Multi-var representations:
    - Scatter plot - for comparing relations between 2 variables & good for estimating linearity
    - Violin plot - comparing relations between 1 continuous var & one categorical var, use a violin plot to display distributions for each category, or use faceting.

- 15.3 - Choosing effective graphical encoding
  - Ask yourself: 'what visual form best exploits the human visual system & space to accurately display your data?'
  - encodings from most to least effective:
    - Position - horizontal/vertical position of an element
    - Length - length of a segment, typically used in a stacked barchart
    - Area - Area of an element such as a circle/rectangle, typically used in a bubble chart/Treemap
    - Angle - Rotational angle of each marker, typically used in a circular layout like a piechart
    - Color - Color of each marker
    - Volume - Volume of a 3D shape, typically used in a 3D barchart

- Sample datasets to show effectiveness of different encodings:



group	value
a	1
b	10
c	11
d	7
e	8

Figure 15.17 Different graphical encodings of the same data. Note the variation in perceptibility of differences between values!

16 - Creating Visuals w/ ggplot2

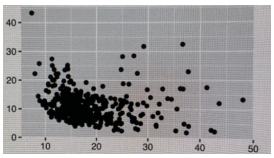
- The Grammar of Graphics to describe a plot:
  - The data being plotted
  - The Geometric Objects on the plot
  - The aesthetic of the Geometric Objects
  - A position adjustment for placing elements on the plot so theres no overlap.
  - A scale for each aesthetic mapping used
  - A coordinate system to organize Geometric Objects
  - The facets of data shown in different plots

- Creating a ggplot:
  - ggplot(data = dataframe) + // pass the df to ggplot, this creates a blank canvas
    geom_point(mapping = aes(x = var1, y = var2)) // call one of the many 'geom_' functions, point() in this case
    - in each 'geom_' function you must specify the (aes)thetic which specifies how the data is displayed, almost all 'geom_' functions require x & y mapping

16.2.1 - Specifying Geometric Objects

- geom_point() - drawing individual points (e.g. scatterplot)
- geom_line() - drawing lines (eg. line chart)
- geom_smooth() - drawing smoothed lines (simple trends / approximations)
- geom_col() - drawing columns (barchart)
- geom_polygon() - drawing arbitrary shapes (drawing an area in a coordinate plane)

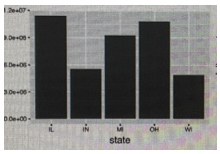
- examples of different plots:



· Point (scatter plot)

- ggplot(data = dataframe) +

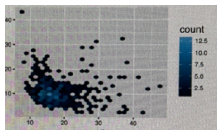
  geom_point(mapping = aes(x = var1, y = var2))



· col (barchart)

- ggplot(data = dataframe) +

  geom_col (mapping = aes(x = var1, y = var2))

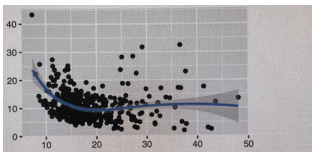


· hex (hexagonal points on a scatterplot)

- ggplot(data = dataframe) +

  geom_hex (mapping = aes(x = var1, y = var2))

· Point w/ smooth (scatter plot, provides a line of best fit)



- ggplot(data = dataframe) +

  geom_point(mapping = aes(x = var1, y = var2))

  geom_smooth(mapping = aes(x = var1, y = var2))