Lecture 5: Data Visualization
Modeling Social Data, Spring 2017
Columbia University

Tara Shui

February 17, 2017

1 Part 1: Guest Lecture

In the first part of today's class, we had Çağatay Demiralp, reseacher from IBM T. J. Watson Research Center, present on data visualization. You can view his slides here. The following is a summary, although much is borrowed from his slides.

1.1 What is visualization?

Why create visualizations? Some examples:

- E.J Marey's sphygmograph (Braun 83)
- John Snow's mapping of cholera cases on Broad St. (Tufte 83)
- Florence Nightingale's Crimean War Deaths (1856)

Jacques Bertin, cartographer and theorist, considered *"visualization as the artificial memory"*:

- Consider time taken for multiplication by mental calculation vs pen and paper.
- Anscombe's quartet: Four datasets have identical summary statistics and linear regression lines, but appear very different when graphed. This is shown in Figure 1.
- Popularity of John W. Tukey (1915-2000) on Wikipedia: After observing the number of page views across time, we notice a spike in January 2011 (Figure 2). Why?
  - Jeopardy Final Round #6063 on Wednesday, January 12, 2011: "John Tukey coined this compound word in 1958 saying it was as important as tubes, transistors, wires, tapes..."
  - Answer: *What is software?*

Bottom line: We create visualizations to

- Record information
- Analyze data to support reasoning
- Communicate information to others

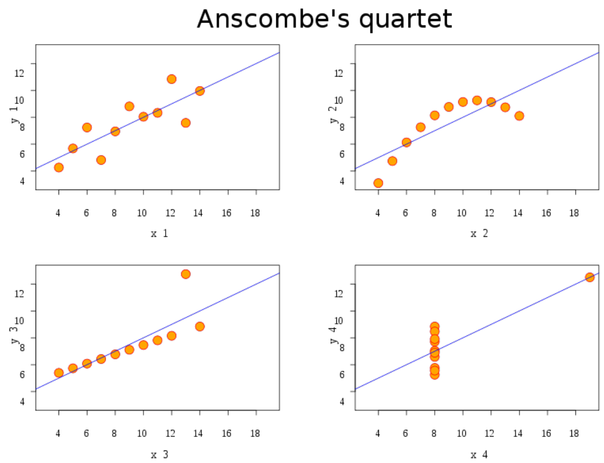

Figure 1: Anscombe's quartet - With proper encoding of the data, we are able to make more sense of it.

Source: Wikipedia.org

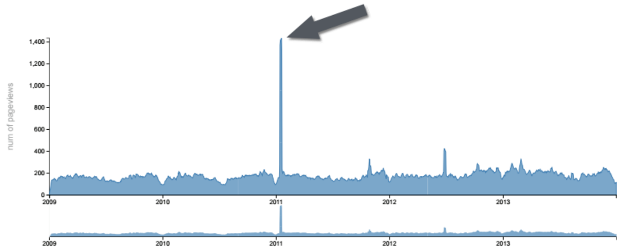

Figure 2: Popularity of John W. Tukey on Wikipedia across time.

Source: Çağatay Demiralp

1.2 What is a ”good” visualization?

Visualization specification involves several choices, i.e. graph type, use of color, etc... How much do these choices matter?

Design Principles (Mackinlay 86)

- Expressiveness - A set of facts is *expressible* in a visual language if the sentences (i.e. the visualizations) in the language express all the facts in the set of data, and only the facts in the data. Watch out for visualizations that:
  - Cannot express the facts, e.g. a one-to-many relation cannot be expressed in a single horizontal dot plot
  - Express facts not in the data, e.g. length of bar in graph says something untrue about the data
- Effectiveness - A visualization is more *effective* than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other visualization.

Design Principles - animation (B. Tversky 02)

- Congruence - The structure and content of the external representation should correspond to the desired structure and content of the internal representation.

- Apprehension - The structure and content of the external representation should be readily and accurately perceived and comprehended.

Design Principles translated:

- Tell the truth and nothing but the truth.
- Use encodings that people decode better.

1.3 Not all visual encoding variables are created equal...

Stevens Power Law (Figure 3) describes the relationship between the magnitude of a physical variable and its perceived sensation:

$$S = I^p \tag{1}$$

where S is perceived sensation, I is physical intensity, and p is an empirically determined exponent. For example, length (such as comparing the length of bars) is perceived linearly, whereas area (such as comparing the area of circles) is underestimated. Stevens' Power Law predicts bias, not necessarily accuracy.

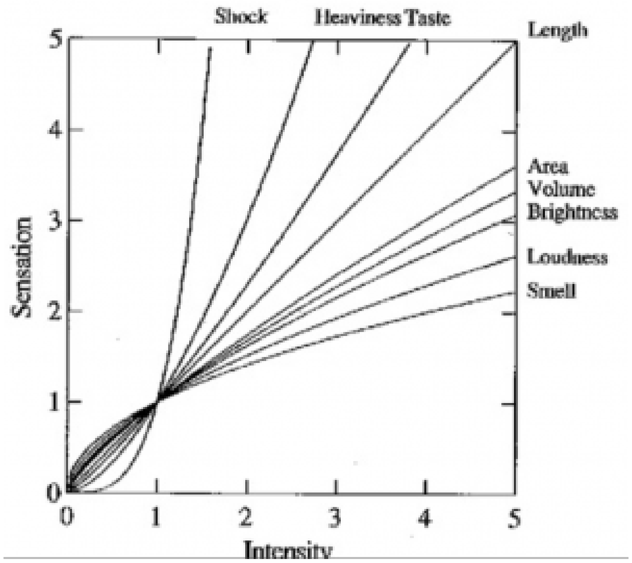

Figure 3: Stevens' Power Law

Source: Stevens, S. The psychophysics of sensory function. Am. Sci. 48, 226253 (1960).

Graphical Perception (Cleveland & McGill 84)

- Task in Experiment I (position - length), Figure 4: "For the two marked bars or divisions, what percent the smaller is of the larger?"
  Type 1, 2, and 3 compare "position" along a common scale, while Type 4 and 5 compare "length".
- Task in Experiment II (position - angle), Figure 5: "What percent each of the other segments or bars is of the largest?"
  The pie chart on the left compares "angle" while the bar chart on the right compares "position".
- Figure 6 shows the results of Cleveland and McGill's Experiment I (top) and II (bottom). Bias was measured by the log absolute estimation error.

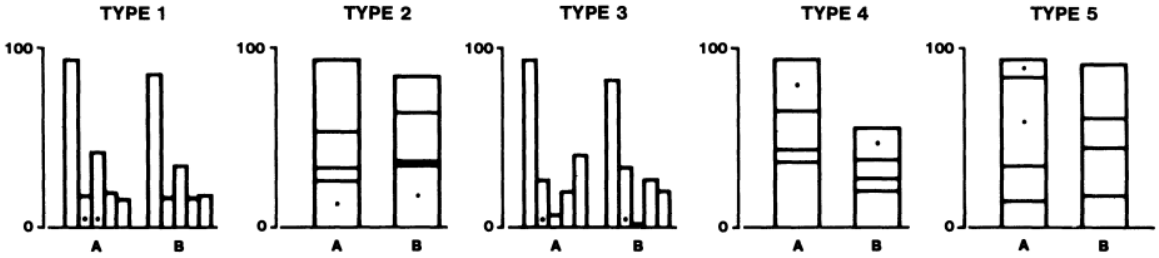

Figure 4: Experiment I (position - length)

Source: Cleveland, W. & McGill, R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. J. Am. Stat. 79, 531554 (1984).

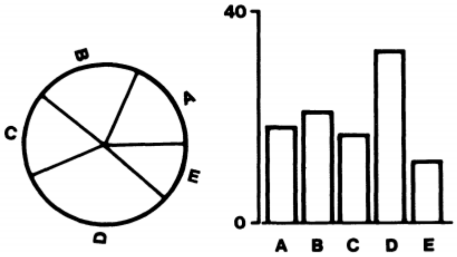

Figure 5: Experiment II (position - angle)

Source: Cleveland, W. & McGill, R.

1.4 Not all data types are created equal...

Data variable types:

- *Nominal* - has two or more categories, but no instrinsic ordering
- *Ordinal* - has two or more categories with natural ordering
- *Quantitative* - numerical variables
- Mackinlay (86) published effectiveness rankings on the different data variable types.

1.5 Color

How should I color my bar chart?

Color Design Guidelines

- Maintain perceptual distinguishability
- Use only a few (¡7) & named, when possible
- Avoid unintended "pop out" of colors
- Get it right in black & white
- Respect cultural norms & the color blind

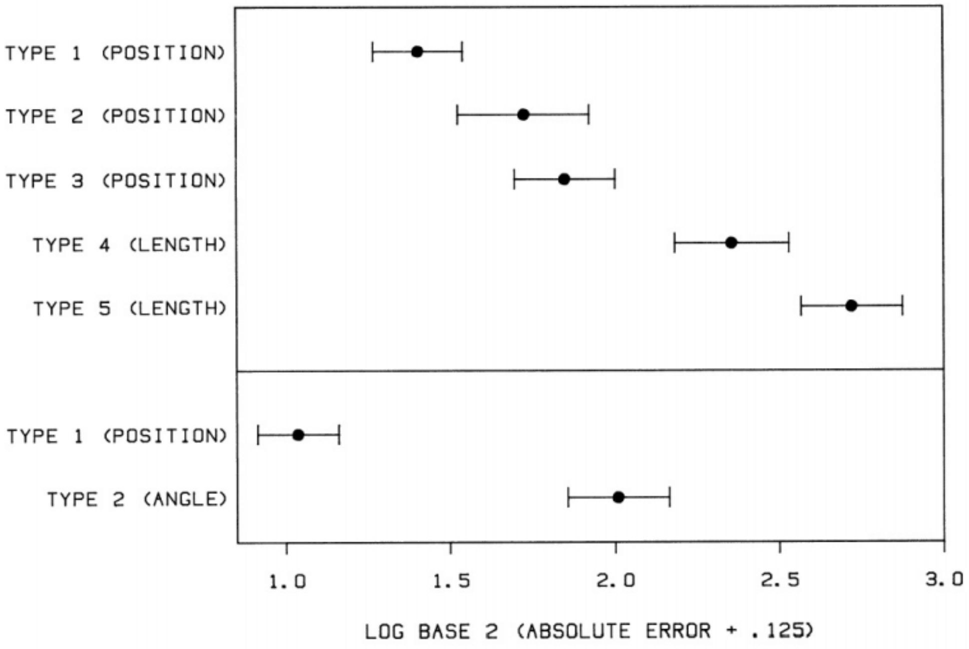

Figure 6: Results

Source: Cleveland, W. & McGill, R.

- Beware of segmentation (rainbows!)
- Consider spatial frequency effects

1.6 Tools

Which tool should I use to create my bar chart?

- Tools come and go, but underlying ideas are important
- Consider the expressiveness vs the speed of tools

Declarative languages:

- Programming by describing *what*, not *how*
  - In contrast to imperative programming, where you must give explicit steps
- Separate specification (what you want) from execution (how it should be computed)
- Examples include HTML/CSS, SQL, and D3
- Advantages:
  - Faster iteration
  - Less code
  - Larger user base than imperative languages
  - Potential to create better visualization - Smart defaults
  - Reusable - Write-once, then re-apply

  - Easier programmatic generation - Write programs which output visualizations e.g., automated search & recommendation

The Grammar of Graphics (Wilkinson)

- Algebra is useful! (Sets, operators, rules)
  - Operators: + (blend), * (cross), / (nest)
- Geometric primitives (marks)
  - "Don't give a pie, give primitives to make a pie and more!"

1.7 The Upshot

Always consider what the one point you are trying to make with this visualization is. Then, how can you make that point the most obvious thing when your visualization is seen?

2 Part 2: Intro to ggplot2

In the second part of today's class, Professor Hofman went over some basic ideas in ggplot2 and how to effectively use visualization tools like ggplot2 to convey a point. To do this, we revisited the Movielens data from Lecture 2. You can find his code in this Jupyter notebook. The following are some pointers and best practices that Professor Hofman mentioned throughout the demo:

- Convert timestamps to datetime objects using package *lubridate* for easy manipulation
  - For example, `year(datetime_object)` can quickly extract the year
- General idea of ggplot2:
  `ggplot(<dataframe>, 'aes' aesthetic-mapping('x' variable = column_name)) + geom_type`
- Use piping operator %>% (similar to command line) so that code is easier to follow
  - But in ggplot2, remember to use + to precede `geom_histogram/geom_point/geom_line` etc. instead of %>% because geoms implicitly compute y variable counts.
- If x is categorical, i.e. `as.factor(rating)`, ggplot will automatically bin by category instead of breaking up by numerical intervals
- `scale_y_continuous(label = comma)` adds commas to give you easily readable numbers on your y axis
- Good practice for making a plot with discrete data: Plot data as points, your model as the line, and optional confidence band as an underlying shade
- Changing the window of your data:
  - Using `coord_cartesian(xlim = c(starting_value, ending_value))` will only change the visual window of your graph, not the data your graph is using
  - Use `xlim(c(starting_value, ending_value)` instead, which limits your data before any transformation occurs
- Be careful when ignoring warnings!
- Playing around with log scales may give the data a completely different story
- `geom_density(fill = <color>)` can be used to smooth and fill the area under a graph
- Consider using a vertical bar to depict mean: `geom_vline(aes(intercept = mean(num_ratings)), linetype = 'dashed')`

- Keep in mind when you are specifying something constant (which you would want to keep outside the aesthetic mapping) vs transforming something IN the data (keep inside the aesthetic mapping)
- `cumsum` gives a running sum of values down the column, whereas `sum` gives one value
- `coord_flip` is good for seeing long x-axis labels like movie titles
- Be aware of the ordering of factors (`mutate(title = reorder(title, mean_rating)))`)
- Every design choice depends on the point you want to make!!
- If you want to split up data and visualize individual plots together, you can use `title` to break out facets: `facet_wrap(~ title)`
- Add standard errors to a line graph with `geom_ribbon`
- `extract` can pull out specific texts (regex)