Interactive Visualization

Problems with Static Visualization

- Many real world datasets are too large to visualize with a single static visualization
  - Remember the hairball graphs we saw
- In general, this means that we have to show a subset of the data set, rather than the whole thing
- Vis designers/developers are usually not domain experts, so it's probably not a good idea to make them decide what to show/hide
- If the end user makes that choice, then we need to have interactive visualizations

View Manipulation

- We'll look at 4 aspects of view manipulation
  - changes of view
  - selection
  - navigation
  - attribute reduction

Changes of View

- During exploration of data, users may need to view the data using different encodings, or organized differently
- An example of changing encoding would be using one of the many different plot types in matplotlib or excel (same data, different encoding)
- An example of re-organizing the data would be sorting by different attributes
- After the view has changed, the user will need to "reorient" themselves to the updated visualization
- Animating the transition between views can help reduce the mental cost of reorienting

Selection

- Some view changes require users to select part of the data/space, like filtering or zooming
- Allowing users to select elements is therefore a fundamental operation we need to support
- Some important questions about selection that you'll need to answer are:
  - which elements can be selected?
  - can multiple items be selected?
  - can the selection be empty?
  - can you select data points with similar values (indirect selection)?
  - how will you indicate which elements are selected (eg highlight with color, size, motion, etc)

Navigation

- Navigation allows us to "move around" the data set to view subsets of it in varying levels of detail
- We generally use "camera" lingo do describe this
- "Panning" means to translate the "camera" (left/right/up/down)
- "Zooming" means to keep the center of the frame the same, but see more/less data
- "Geometric Zooming" works like zooming a physical camera
- "Semantic Zooming" means we change what information is displayed as we zoom. Maybe we show more attributes as data points get bigger, or as we see fewer of them

Attributes

- Navigation and filtering can help us reduce the number of data points shown on screen at a time
- However, if we have many attributes associated with each data point, we can't show them all at the same time without clutter
- We need to allow users to control which attributes are displayed, and also filter which data points are shown based on attributes
- A common technique is "detail on demand" where you can see a lot of attributes of a data point when you select it, for example

Multiple Views

- The previous techniques allow a single view to change over time so that at a user can explore their data set in a flexible way
- However, it is not very effective for comparison tasks
- Imagine trying to compare two different regions of a data set. With an interactive view, you'd have to navigate back/forth and use your memory to try to pick out differences/similarities
- Imagine trying to use 2 different encodings to understand a subset of the data. You'd have to use your memory to help understand the connection between the two encodings
- Both of these issues can be mitigated by multiple linked views of the data

Faceting

- Faceting means providing multiple visualizations of different parts/encodings of a data set simultaneously
- Facets can vary based on which part of the data they include, which encoding they use, or which attributes they show, or combinations of all of the above
- For facets which show different encodings/attributes of the same data points, linked highlighting can be used to indicate correspondences between views
- Another linking technique is shared navigation: panning/zooming in one view also pans/zooms in the other

Juxtaposition vs Superimposition

- Multiple views can be juxtaposed (drawn near each other) or superimposed (drawn on top of each other)
- Juxtaposition requires more pixels (maybe a less significant cost than it used to be, but still a limited resource)
- Superpositions have the tendency to increase clutter and are better suited when "upper" layers are "sparse"