Week 6 - Comparisons, Elements, Engines and More

Warm-Up Activity

- https://gizmodo.com/observatories-across-the-world-announce-groundbreaking-1819500578
- Wave shows the increase in energy
- Could have easily missed the wave because of the noisy background
- Designed for the scientific community, not the general public

Assignment Clarification

- Leave unused code in ipynb files
- Comment out notes
- Explain why you did what did, not how you did it in the assignment narrative

Composition

- Pie charts make it difficult to determine relative size to the audience
  - You can add labels, but it's still not clear
  - 3-dimensional pie charts just make it even more difficult to discern proportional
    - Principal of proportional ink
  - Donut charts are slightly better, audience just to have to determine the arc length
- Alternatives:
  - Tree-map, not circular but allows the audience to discern quantity by relative areas
  - Stacked bar chart
    - Audience just has to analyze one dimension of length
    - Doesn't show parts of a whole
  - Waterfall chart
  - Hierarchical data
    - Sunburst
      - Stacked donut chart
    - Nested box area
  - Comparison
    - Comparison bar chart
    - Stacked bar chart
      - Show parts of a whole
    - Stacked line charts
      - Show time-evolving data
      - Builds up to 100%

bqplot

- Similar to matplotlib
- Built to be interactive automatically
- Two principal interfaces:
  - Pyplot-like interface
    - Create, modify, show
  - Object-oriented API for constructing interactive visualizations

Object-oriented programming

- Define objects and classes of objects
  - Example. Objects = vehicles, classes = 3-wheeled objects
  - Classes are defined by attributes
- Classes define traits (i.e. attributes) which we can use for visualizations
  - We will use *traitlets* library
    - Provide methods for datatype-verification
- Once we have an object that has traits, we can watch that object for changes.
- We can use ipywidgets to build interactive widgets that modify changes in traits
- Types of widgets:
  - Type 1: Built-in widgets to ipywidgets
  - Type 2: Widgets that provide indicators and restricted selections
- Linking is how we connect traits to the widgets
- Bqplot objects:
  - *Mark*
    - The mechanism for displaying data
    - Can be represented by *lines*
  - *Scale*
    - Relationship between position and data values
  - *Axis*
    - Where data is placed
  - *Figure*
    - Object that contains the *marks, scale, axis,* and interactions

Live coding demonstration

Will be posted on course site: https://uiuc-ischool-dataviz.github.io/spring2019/week06/