INTERACTION & TRANSITION I

RESPONSIVE CHARTS

Georgia Tech Visualization Lab

D3 Event Listeners

- Use .on(`click`, <callback>) or .on(`mouseover`, <callback>)
- Provides access to:
  - **`this`** - the DOM element
  - **`d and i`** - the data element and index

CSS :hover :active (psuedo-classes)

- Use CSS Rules when the state of an element changes
- Quick and easy yet limited
- E.g. the following will add a stroke to all hovered circles

```
circle:hover {
  stroke: #333;
}
```

TRANSITIONS

Georgia Tech Visualization Lab

D3 Transitions

- Just add `.transition()` - its really that easy!
- Can also specify:
  - **`.duration(<milliseconds>)`** - the duration of the animation
  - **`.delay(<milliseconds>)`** - time before animation starts
  - **`.ease(<d3.easeFunction>)`** - the timing of the animation

CSS key-frames

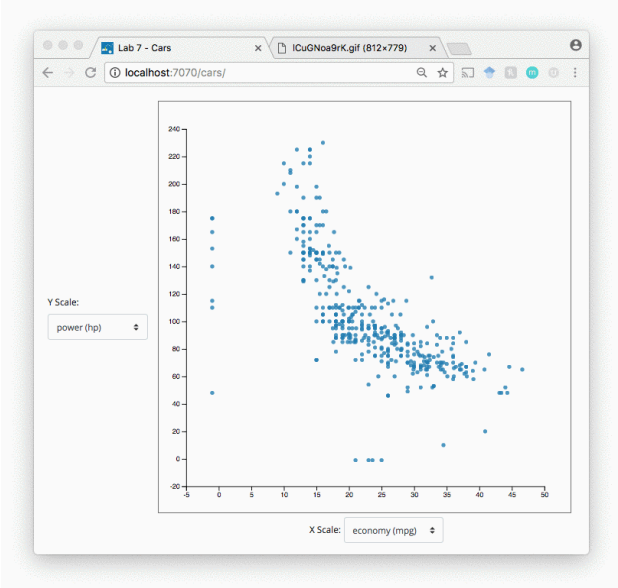
- Use CSS to define the key-frames of an animation
- Easy to find examples on the Web, gives you more control than D3 transitions at times

Georgia Tech Visualization Lab

A LIVE EXAMPLE

TODAY'S ACTIVITY

Georgia Tech Visualization Lab



LAB PROCEDURE

Georgia Tech Visualization Lab

Before Class

- Read Chapter 10 - *Interactive Data Visualization for the Web by* Scott Murray
- *Git pull* example code (https://github.gatech.edu/CS-4460/Labs.git)

In-Class

- Open Lab 7 instruction page (https://github.gatech.edu/CS-4460/Labs/wiki)
- Work through activities
- First things first, start an http server with python at `07_lab` directory

After Class

- Submit "main.js" to the "Lab 7" T-Square assignment
  - Only submit that one file - Deadline is **1:15 PM Today**
- P4 Out **This Sunday - Due Nov. 15 Wednesday**