

Overview

A Martini-Glass layout was used to tell the story of Domestic Hate Crime. The intent was to tell a story in a contextual and directed manner, while at the same time allowing the audience to undertake self-exploration at the end of the guided story so they might gain additional insight into story's dimensions.

The D3 JavaScript library in combination with HTML5 and CSS are utilized to create the visualizations. D3 add on libraries, created by the open source community, are incorporated into the visualization in in order to provide dynamic and static annotation capabilities. The Bootstrap JavaScript open source library plays an important role in the presentation. It provides a consistent HTML framing and a consistent navigation metaphor. Complete transparency motivates me to acknowledge Ashish Kumar for turning a me onto Bootstrap; and for providing me with the HTML template that I used for this presentation. I significantly modified the template but I retained navigation bar on the top of each page and the navigation buttons on the bottom of each page.

Scene Layout and Design

The first three scenes of the presentation represent the stem of the Martini Glass layout as they provide author directed content. They provide the empirical context and overview of the main drivers of the story:

- The bias motivations for the hate crime incidents.
- The perpetrators who commit these hate crime incidents.

The fourth scene recaps the learnings from the three prior scenes, and sets the stage for user exploration in fifth scene – the “cone bowl” of the Martini Glass.

The fifth scene is where users can explore the interaction between bias motivated hate crimes and the perpetrators who commit these crimes. The user has the ability to select a specific hate crime bias and then see which perpetrator cohort is the most likely to commit the selected hate crime type.

Annotation Usage

Three types of annotations are used.

- Static textual annotations that are embedded directly into the visualization. Their purpose is to highlight key points of interest, such as historical events, that shape the story I am trying to communicate. Static annotations are used in the first three scenes because this is where I am trying to communicate a point of view. I used the d3-Annotation JavaScript library, developed by Susie Lu, to create static annotations. A

consistent annotation style, fill color and text font are employed for all of these static annotations.

- Dynamic annotations (e.g., tool tips) that change context as the viewer hovers over each visual mark in a chart. Tool tips provide a means to visualize the quantitative and categorical values of each data point in a particular chart. They are employed on four of the five scenes of the story. I employed the d3-tip.js JavaScript library, developed by David Goltz, Constantin Gavrillete and Justin Palmer, to create these dynamic annotations (e.g., tool tips). A consistent tooltip style, fill color and font are used for all of these dynamic annotations.
- Legends that tie colors on a multiline chart to a particular ordinal data categorization. Without the use of legends, it would be difficult for the viewer to understand charts where multiple colors are used. I employed the d3-Legend JavaScript library, developed by Susie Lu, to create legends on the two scenes where colorized multiline charts are used.

The static textual annotations, tooltips and legends are cleared on leaving each scene; as each scene is has a unique context.

Parameter Usage

Parameters are used to filter subsets of the data in order to create the multiline charts in scene 2 and scene 5. The specific parameters used in scene 2 and scene 5 are “bias motivation category” and “bias motivation subcategory”, respectively.

The “bias motivation subcategory” data attribute is included in each data point on the colorized multi-line charts in scene 5. This data attribute serves as a parameter for the scene’s on-click events. Each on-click event passes the “bias motivation subcategory” value as a parameter to a on-click event handler. The on-click event handler uses the parameter to select the appropriate data to display on the bar chart that accompanies each of the multiline charts in scene 5.

Triggers

As mentioned above, the on-click events in the multiline charts in scene 5 trigger a data value (e.g., parameter) to be passed to the on-click event handler. The on-click event handler in turn uses the data value (e.g., parameter) to select the data that is visualized in the accompanying bar charts.

Other window events, such as mouse over and mouse, out are used to trigger the toggling on and off of tool tips. The “X” and “Y” parameters are passed by these window events to tell the windowing system where to render the tool tips.

Viewers are made aware of the available triggering events by helpful hints that are displayed when the page load event is triggered on each scene. Each helpful hint displays for 10 seconds

when a page is loaded and then it dissolves. A window timer is employed to control when a helpful hint fades from view on a scene.