# Checkpoint IV: First Prototype

**Information Visualization**
**Checkpoint IV**

**Group:  G14**
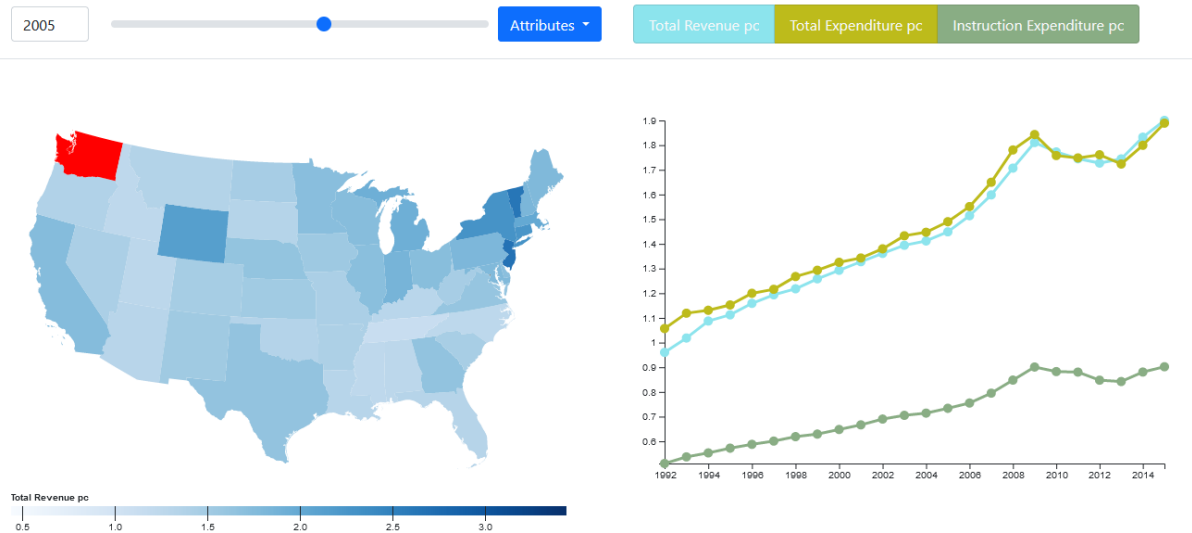**Date:   2021/11/01**

## Layout



*Figure 1: Current layout of the visualization after implementing two idioms*
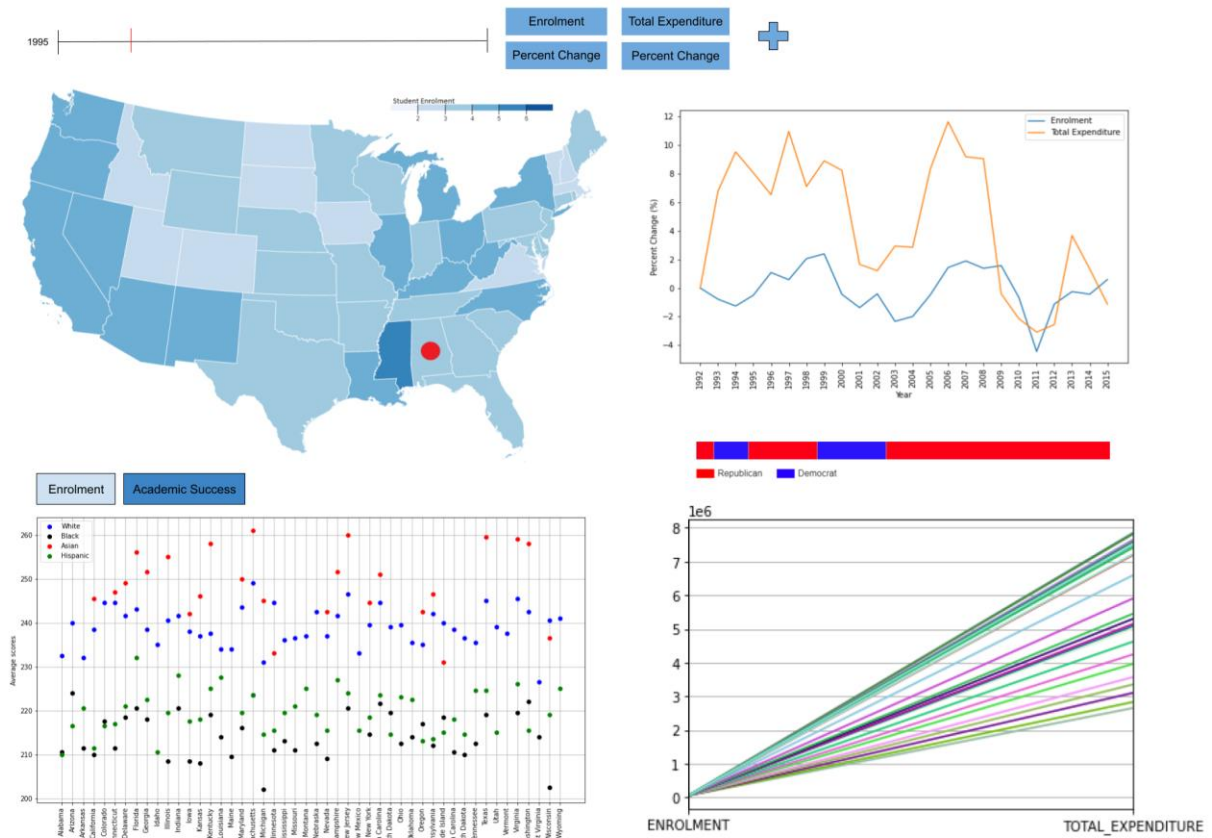


*Figure 2:  Sketch of the Visualization*

We can consider our visualization to be split into three sections:

1. A top section with a slider and a dropdown menu for the user to pick the year and the attributes that they want to visualize. After toggling an attribute in the drop-down menu, it will appear on a list next to it (attributes list), which can be reordered manually either by interacting with the menu or by dragging and dropping each element into the desired order.
2. A left section with, from top to bottom, a choropleth map, and a dot plot to encode data for the selected year. At this point in time only the choropleth map is implemented.
3. A right section with, from top to bottom, a line chart, a Gantt chart, and a parallel coordinates plot to visualize the data throughout the years for the selected state. At this point in time only the line chart is implemented.
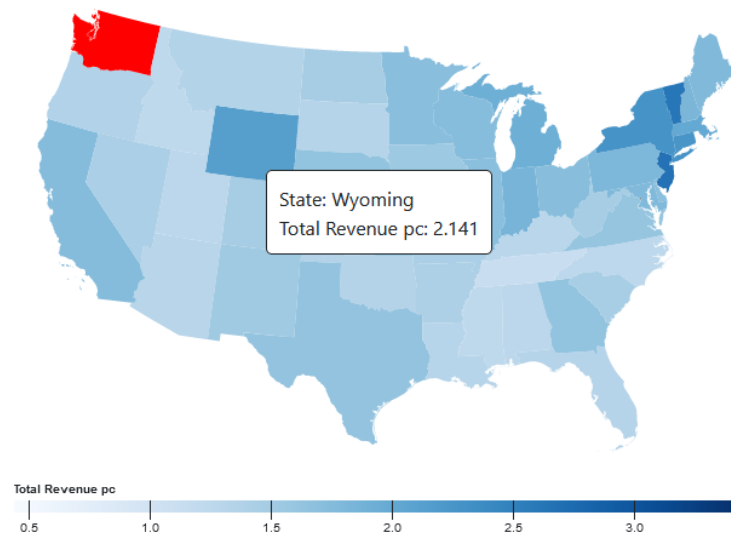
## Visual Encoding



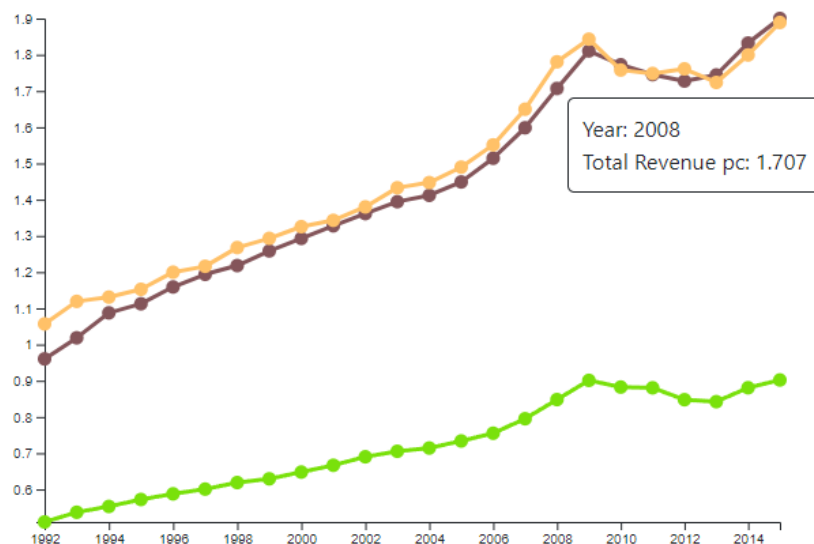*Figure 3: Choropleth map with a state selected (in red)*



*Figure 4: Line chart for the selected attributes and state.*

The choropleth map encodes the values of the first attribute on the attributes list for the selected year, using 2D shapes for marks. The colour saturation encodes the attribute value, and the shape, position and spatial region encode the corresponding state. When the user hovers a state, a tooltip

with the state and encoded value gets rendered. Every state is clickable and gets highlighted with red when clicked.

The line chart represents the evolution through time of the selected attributes for the selected state, using dots connected by lines as marks. The x position encodes the year and the y position the value of the attributes. The line's slope encodes the change in the attribute's values. Colour is used to encode what attribute each line corresponds to and mirrors the colour of the attributes on the attributes list. When the user hovers a dot, a tooltip with the year and encoded value gets rendered. The lines are also clickable and affect what is shown on the choropleth map.

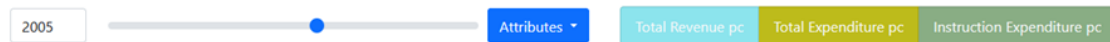# Implementation of Linking Mechanism



*Figure 5: Selection menu*

The Selection Menu allows the user to select a year and a subset of the attributes. This selection is the starting point of the visualization and affects every idiom. It determines the year the data on the state-based idioms (choropleth map and dot plot) is from and the attributes displayed on the line chart, parallel coordinates and choropleth map.

In the choropleth map we can select a state by clicking on it, changing the data shown on the line chart to reflect the selected state.

When the user clicks on a line on the line chart, the attribute that line corresponds to gets selected for the choropleth map.

When we have the dot plot and the parallel coordinates chart implemented, those will also be linked in a similar fashion. In addition to the state selection on the choropleth map, the dot plot will also allow state selection and the parallel coordinates will not only allow year selection by clicking on one of its lines but will also change its axis' order when the user changes the ordering of the attributes on the attributes list.

The linking mechanisms are implemented in the following way:

- Attribute selection is stored in an internal array. When the user toggles an attribute, this attribute is appended/removed from the array. When the user changes the ordering of the attributes this ordering change is reflected on the array.
- Year selection and State selection work in a similar fashion - when a year/state gets selected, the selection is stored in an internal variable and the data gets filtered by year/state (to increase performance, the filtered data is cached until the next filtering happens).
- Every time a configuration change happens - addition/removal of an attribute, reordering of attributes, year selection, and state selection - the affected idioms are updated to reflect these changes. With this mechanism, we can easily and transparently implement new idioms.