React Chart-js Preprocessor

Parent component should load completely without any subcomponents.

Subcomponents only affect parent on user input. I.e. no subcomponents have onMount side effects.

Presets are not necessary, but they are strongly encouraged. Presets generally work like this:

* Graph reads a number of data points, including 1) layersSelected, and 2) styles.
* Both layersSelected and style are saved in state of the parent component.
* Using the layer selector directly edits layersSelected.
* Using the preSet style editor also directly edits styles.
* When a preSet is selected,
  + its layersSelected are read and completely overwrites any existing layersSelected, and
  + its styles are read and are hydrated with existing styles, specifically Object.assign() at the top level of styles, so if style A is in a preSet, it overwrites style A in state. But if style B is in state, and the preSet does not include style B, then style B is copied directly from state.
* PreSets are only edited when saved. So if you use the preSet editor, but do not save, when you select another preSet, your edits are lost.
* When saving a preSet, the current state’s layersSelected and styles completely overwrite whatever is in a preSet. Usually this occurs as follows:
  + User selects a preSet. State now contains that preSet’s information (with hydrated styles).
  + User makes a few small edits to the preSet.
  + User saves the preSet (which overwrites in full).
* When a user makes any change to layersSelected or styles, the state is in editing mode. This check is done via taking an action at least once, not assigned by continually comparing versions of state. Thus, if the user un-does an action, even though the current and prior preSets match, the state still says editing mode.
  + Alert the user of editing mode so they know they need to save.
* When styles are edited, the entire style object is copied, then a single style is overwritten. This is O(1) for a single style, but the nature of immutably editing styles requires a full immutable copy, which is O(n). Since each style within the style object is also an object, I don’t think the assignment is deep, so it might only be going through pointers.