**Data binning / bucketing functional requirements**

**Defining bins**

1. Default bins definition that is ‘reasonable’ (i.e., can be explained *why* it is a good choice) – likely equal interval
2. Default bin definition that is clear to the user (i.e., this is the default method)
3. Default bin definition should be easy to change (to other out-of-the-box solutions, or to manually adjust individual bins)
4. Communication to user how each bin has been defined numerically
5. Graphical communication about bin definition w.r.t. data distribution (histogram or other)
6. ‘out of the box’ pre-calculated definitions for bins (e.g., equal interval, mean/standard deviation, quantiles, some sort of model-based approach)
7. User can manually define bins.
8. Manually defined bins can be based off of statistical / out-of-the-box bins that have been modified (e.g., move sliders on histogram or type in new break values); they do not need to be created from scratch
9. Users can set a ‘center’ point to define break for diverging color schemes
10. Center point can be defined as a break end (‘critical break’) or as an entire bin (‘critical class’). This does not depend on whether there are an even or odd number of bins.
11. Bins must be defined numerically (to identify how data values are assigned to a bin), however, non-numeric aliases can be assigned to give semantic meaning to the bins.
12. Non-numeric aliases can be defined for bin breaks or for bins

|  |  |
| --- | --- |
|  | The bins each have a definition of the range of values, but one bin break has been identified with an additional semantic label |

1. User can set a ‘no data’ value to remove that from the calculations (e.g., -9999)
2. User can set a start/end value to limit which values are used in the classification.
3. Values above or below the defined start/end can be color encoded separately (either to match the encoding for the start/end values OR for encoding as separate ‘out-of-range’ values)

* This, in conjunction with #12 meets the use case of a user wanting to constrain the color range. The start/end values are the constraints on the color range, and the values above/below are set to the same color as the low / high values; semantic labeling can be used to identify the low class as “less than x” and the high class as “more than x”

1. Users can share a bin definition
2. Users can define bins with static values (e.g., ‘400’) or with calculations (‘MAX([field])’)
3. Users can specify when bins are updated.
   1. Filtering
   2. Data updates (add / remove values; change in distribution)
   3. Where else would data change?

**Understanding bins:**

1. Users can see a visualization of the data distribution, with bin breaks and any associated color encoding
2. Users can add basic statistical distribution information (e.g., mean and standard deviations; outliers)
3. Users can adjust the bin breaks in the visualization of the data distribution and see the changes update immediately (e.g., change the bin break, the color encoding changes to match)

**Labeling and communicating bins:**

1. Users can create legends that list absolute ranges of values associated with each bin.
2. Users can create aliases that list semantic definitions for each bin
3. Users can create aliases with numeric or semantic meaning for each bin *break*
4. Users can customize legends easily
   1. Spacing between symbols
   2. Labeling for bins and/or bin break points
5. Users can quickly and easily switch between aliases for bins (e.g., one alias for internal visualizations, one for external with non-numeric description of the bins, etc.)