`process` Design

`process greenerthumb` data.

Programs

All programs accept JSON messages from STDIN and report results to STD-OUT. Each program terminates with a message printed to STDERR if any message is malformed.

The expected base message format is:

```
{
  "Header": {
    <name>: <value>,...
  },
  <name>: <value>,...
}
```

Each program has different required header keys. These are only parsed as fields are passed, so messages with a malformed header but no fields won't cause errors to be output.

`summarize`

`summarize` reads all input until STDIN is closed and then reports a 5-number-summary for each data-type along with how many instances of that data-type were included.

The required header key is "Name".

`flatten`

`flatten` smooths data by keeping a sliding window of 3 instances of a data-type and replacing it with a weighted average of the 3 instances biased towards the middle instance. The first instance and last instance have a copy of themselves used as the instance to the left and right of them.

The left and right values are weighted by 1/6 each while the middle value is weighted 2/3.

The required header key is "Name".

`filter`

`filter` instances of data-types by specifying a list of ANDing conditions in the set of less than or equal to, less than, equal, greater than, and greater than or equal to and filtering STDIN according to the conditions.

An epsilon value for comparisons can also optionally be passed. The system epsilon is used otherwise.

There are no required header keys.

`clean`

`clean` reads all input until STDIN is closed and filters instances that are more than a passed number of standard deviations away from the mean.

The required header key is "Name".

`select`

`select` messages from STDIN with names in an included set.

The required header key is "Name".