bin

The bucket command is an alias for the bin command.

Description

Puts continuous numerical values into discrete sets, or bins, by adjusting the value of <field> so that all of the items in a particular set have the same value.

The bin command is automatically called by the chart and the timechart commands. Use the bin command for only statistical operations that the chart and the timechart commands cannot process. Do not use the bin command if you plan to export all events to CSV or JSON file formats.

Syntax

bin [<bin-options>...] <field> [AS <newfield>]

Required arguments

field

Syntax: <field>
Description: Specify a field name.

Optional arguments

bin-options

Syntax: bins | minspan | span | <start-end> | aligntime
Description: Discretization options. See the Bins options section in this topic for the syntax and description for each of these options.

newfield

Syntax: <string>
Description: A new name for the field.

Bin options

bins

Syntax: bins=<int>
Description: Sets the maximum number of bins to discretize into.

minspan
**Syntax:** minspan=<span-length>

**Description:** Specifies the smallest span granularity to use automatically inferring span from the data time range.

**span**

**Syntax:** span = <log-span> | <span-length>

**Description:** Sets the size of each bin, using a span length based on time or logarithm-based span.

**<start-end>**

**Syntax:** start=<num> | end=<num>

**Description:** Sets the minimum and maximum extents for numerical bins. The data in the field is analyzed and the beginning and ending values are determined. The start and end arguments are used when a span value is not specified.

You can use the start or end arguments only to expand the range, not to shorten the range. For example, if the field represents seconds the values are from 0-59. If you specify a span of 10, then the bins are calculated in increments of 10. The bins are 0-9, 10-19, 20-29, and so forth. If you do not specify a span, but specify end=1000, the bins are calculated based on the actual beginning value and 1000 as the end value.

If you set end=10 and the values are >10, the end argument has no effect.

**aligntime**

**Syntax:** aligntime=(earliest | latest | <time-specifier>)

**Description:** Align the bin times to something other than base UTC time (epoch 0). The aligntime option is valid only when doing a time-based discretization. Ignored if span is in days, months, or years.

**Span options**

**log-span**

**Syntax:** [<num>]log[<num>]

**Description:** Sets to log-based span. The first number is a coefficient. The second number is the base. If the first number is supplied, it must be >= 1.0 and < base. Base, if supplied, must be real number > 1.0 (strictly greater than 1).

**Example:** span=2log10

**span-length**

**Syntax:** <int>[<timescale>]
**Description:** A span of each bin. If discretizing based on the `_time` field or used with a timescale, this is treated as a time range. If not, this is an absolute bin length.

```yaml
<timescale>
  **Syntax:** `<sec>` | `<min>` | `<hr>` | `<day>` | `<month>` | `<subseconds>`
  **Description:** Time scale units. If discretizing based on the `_time` field.
  **Default:** `sec`
```

<table>
<thead>
<tr>
<th>Time scale</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;sec&gt;</code></td>
<td><code>s</code></td>
<td><code>sec</code></td>
</tr>
<tr>
<td><code>&lt;min&gt;</code></td>
<td><code>m</code></td>
<td><code>min</code></td>
</tr>
<tr>
<td><code>&lt;hr&gt;</code></td>
<td><code>h</code></td>
<td><code>hr</code></td>
</tr>
<tr>
<td><code>&lt;day&gt;</code></td>
<td><code>d</code></td>
<td><code>day</code></td>
</tr>
<tr>
<td><code>&lt;month&gt;</code></td>
<td><code>mon</code></td>
<td><code>month</code></td>
</tr>
<tr>
<td><code>&lt;subseconds&gt;</code></td>
<td><code>us</code></td>
<td><code>ms</code></td>
</tr>
</tbody>
</table>

**Usage**

The `bin` command is usually a **dataset processing command**. If the `span` argument is specified with the command, the `bin` command is a **streaming command**. See Command types.

**Examples**

**Example 1:**

Return the average "thruput" of each "host" for each 5 minute time span.

```bash
... | bin _time span=5m | stats avg(thruput) by _time host
```
Example 2:

Bin search results into 10 bins, and return the count of raw events for each bin.

```
... | bin size bins=10 | stats count(_raw) by size
```

Example 3:

Create bins with an end value larger than you need, ensure that all possible values are included.

```
... | bin amount end=1000
```

Example 4:

Align the time bins to 3am (local time). Set the span to 12h. The bins will represent 3am - 3pm, then 3pm - 3am (the next day), and so on.

```
... | bin _time span=12h aligntime=@d+3h
```

Example 5:

Align the bins to the specific UTC time of 1500567890.

```
... | bin _time aligntime=1500567890
```

See also

chart, timechart

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has using the bin command.