Build a chart of multiple data series
Build a chart of multiple data series

Splunk **transforming commands** do not support a direct way to define multiple data **series** in your charts (or timecharts). However, you CAN achieve this using a combination of the **stats** and **xyseries** commands.

The **chart** and **timechart** commands both return tabulated data for graphing, where the x-axis is either some arbitrary field or _time, respectively. When these commands are used with a split-by field, the output is a table where each column represents a distinct value of the split-by field.

In contrast, the **stats** command produces a table where each row represents a single unique combination of the values of the group-by fields. You can then use the **xyseries** command to redefine your data series for graphing.

For most cases, you can simulate the results of "... | chart n by x,y" with "... | stats n by x,y | xyseries x y n". (For the **timechart** equivalent of results, x = _time.)

**Scenario**

Let’s say you want to report on data from a cluster of application servers. The events gathered from each server contain information such as counts of active sessions, requests handled since last update, etc. and are placed in the **applications_servers** index. You want to display each server instance and the number of sessions per instance on the same timechart so that you can compare the distributions of sessions and load.

Ideally, you want to be able to run a timechart report, such as:

```
index=application_servers | timechart sum(handledRequests)
avg(sessions) by source
```

However, timechart does not support multiple data series; so instead, you need run a search similar to the following:

```
index=application_servers | bin _time | stats sum(handledRequests) as hRs, avg(sessions) as ssns by _time,source | eval s1="handledReqs sessions" | makemv s1 | mvexpand s1 | eval yval=case(s1=="handledReqs",hRs,s1=="sessions",ssns) | eval series=source+":"+s1 | xyseries _time,series,yval
```
Walkthrough

... | bin _time

The first thing that you need to do, before the `stats` command, is to separate the events by time.

... | stats sum(handledRequests) as hRs, avg(sessions) as ssns by _time,source

The `stats` command is used to calculate statistics for each source value: The sum of `handledRequests` values are renamed as `hRs`, and the average number of `sessions` are renamed as `ssns`.

... | eval s1="handledRequests sessions" | makemv s1 | mvexpand s1

This uses the `eval` command to add a single-valued field "s1" to each result from the `stats` command. Then, the `makemv` command converts s1 into a multivalued field, where the first value is "handledRequests" and the second value is "sessions". The `mvexpand` then creates separate series for each value of s1.

... | eval yval=case(s1=="handledRequests",hRs,s1=="sessions",ssns)

This uses the `eval` command to define a new field, `yval`, and assign values to it based on the case that it matches. So, if the value of s1 is "handledRequests", `yval` is assigned the "hRs" value. And, if the value of s1 is "sessions", `yval` is assigned the "ssns" value.

... | eval series=source+":"+s1

This uses the `eval` command to define a new field, `series`, which concatenates the value of the host and s1 fields.

... | xyseries _time,series,yval

Finally, the `xyseries` command is used to define a chart with _time on the x-axis, yval on the y-axis, and data defined by series.