Statistical and charting functions

http://docs.splunk.com/Documentation/Splunk/6.4.1/SearchReference/CommonStatsFunctions

You can use the statistical functions with the `chart`, `stats`, and `timechart` commands.

- Functions that you use with the `stats` command, can also be used with the `eventstats`, `streamstats`, and `geostats` commands.
- Functions that you use with the `chart`, `stats`, and `timechart` commands can also be used with their respective summary indexing counterparts: `sichart`, `sistats`, and `sitimechart`.
- Functions that you can use to create sparkline charts are noted in the tables below. Sparkline is not a search command, it is a function that applies to only the `chart` and `stats` command and allows you to call other functions. For more information, see Add sparklines to search results in the *Search Manual*.

String and numeric field values

Most of the statistical and charting functions expect the field values to be numbers. All of the values are processed as numbers, and any non-numeric values are ignored.

Some functions process the field values as literal string values, even thought the values are numbers.

- `count`
- `dc`
- `earliest`
- `estdc`
- `estdc_error`
- `first`
- `latest`
- `last`
- `list`
- `max`
- `min`
- `mode`
- `values`

For example, you use the `distinct count` function and the field contains values such as “1”, “1.0”, and “01”. Each value is considered a distinct string value.

The only exceptions are the `max` and `min` functions. These functions process values as numbers if possible. For example, values such as “1”, “1.0”, and “01” are processed the same numeric value.

Types of functions

There are several types of statistical and charting functions:

- Aggregate functions
- Event order functions
- Multivalue functions
- Time functions

Aggregate functions

Most aggregate functions are used with numeric fields. However, there are some functions that you can use with either alphabetic string fields or numeric fields. The function descriptions indicate which functions you can use with alphabetic strings.

Function	Description	Commands	Examples
`avg(X)`	Returns the average of the values of field X. See also, mean(X).	`chart`, `stats`, `timechart`, `sparkline()`	This examples returns the average response time: `avg(responseTime)`

`c(X) |`
`count(X)`
	Returns the number of occurrences of the field X. To indicate a specific field value to match, format X as eval(field="value").
	Processes field values as strings.
		`chart`, `stats`, `timechart`, `sparkline()`	This example returns the count of events where `status` has the value "404":
			`count(eval(status="404"))`
			These generate sparklines for the counts of events. The first looks at the `_raw` field. The second counts events with a `user` field:

			`sparkline(count)`
			`sparkline(count(user))`

`dc(X) | distinct_count(X)`	Returns the count of distinct values of the field X.
	Processes field values as strings.
		chart, stats, timechart, sparkline()	This example generates sparklines for the distinct count of devices and renames the field, "numdevices":
			`sparkline(dc(device)) AS numdevices`
			This example counts the distinct sources for each sourcetype, and buckets the count for each five minute spans:
			`sparkline(dc(source),5m) by sourcetype`

`estdc(X)`	Returns the estimated count of the distinct values of the field X. Processes field values as strings.	chart, stats, timechart	
`estdc_error(X)`	Returns the theoretical error of the estimated count of the distinct values of the field X. The error represents a ratio of abs(estimate_value - real_value)/real_value. Processes field	chart, stats, timechart	

	values as strings.		
`max(X)`	Returns the maximum value of the field X. If the values of X are non-numeric, the max is found using lexicographical ordering. Processes field values as numbers if possible, otherwise processes field values as strings.	`chart`, `stats`, `timechart`, `sparkline()`	This example returns the maximum value of "size": `max(size)`
`mean(X)`	Returns the arithmetic mean of the field X. See also, avg(X).	`chart`, `stats`, `timechart`, `sparkline()`	This example returns the mean of "kbps" values: `mean(kbps)`
`median(X)`	Returns the middle-most value of the field X. **Note:** The median calculation is more accurate with an odd numbers of events. If you have an even number of events, the median is approximated to the higher of the two values.	`chart`, `stats`, `timechart`	
`min(X)`	Returns the minimum value of the field X. If the values of X are non-	`chart`, `stats`, `timechart`	

	numeric, the min is found from lexicographic ordering. Processes field values as numbers if possible, otherwise processes field values as strings.		
`mode(X)`	Returns the most frequent value of the field X. Processes field values as strings.	`chart`, `stats`, `timechart`	

`p<X>(Y) |` `perc<X>(Y) |` `exactperc<X>(Y) |` `upperperc<X>(Y)`	Returns the X-th percentile value of the numeric field Y, where X is an integer between 1 and 99. The percentile X-th function sorts the values of Y in an increasing order. Then, if you consider that 0% is the lowest and 100% the highest, the functions picks the value that corresponds to the position of the X% value.
	The functions `perc`, `p`, and `upperperc` give approximate values for the integer percentile requested. The approximation
		`chart`, `stats`, `timechart`	For the list of values `Y = {10,9,8,7,6,5,4,3,2,1}`:
			`perc50(Y)=6`
			`perc95(Y)=10`

	algorithm that is used, which is based on dynamic compression of a radix tree, provides a strict bound of the actual value for any percentile. The functions `perc` and `p` return a single number that represents the lower end of that range, while `upperperc` gives the approximate upper bound. The `exactperc` function provides the exact value, but will be very expensive for high cardinality fields.		
`range(X)`	Returns the difference between the max and min values of the field X ONLY IF the value of X are numeric.	`chart`, `stats`, `timechart`, `sparkline()`	
`stdev(X)`	Returns the sample standard deviation of the field X.	`chart`, `stats`, `timechart`, `sparkline()`	This example returns the standard deviation of wildcarded fields "*delay" which can apply to both, "delay" and "xdelay". `stdev(*delay)`
`stdevp(X)`	Returns the population standard deviation of	`chart`, `stats`, `timechart`, `sparkline()`	

	the field X.		
sum(X)	Returns the sum of the values of the field X.	chart, stats, timechart, sparkline()	sum(eval(date_hour * date_minute))
sumsq(X)	Returns the sum of the squares of the values of the field X.	chart, stats, timechart, sparkline()	
var(X)	Returns the sample variance of the field X.	chart, stats, timechart, sparkline()	
varp(X)	Returns the population variance of the field X.	chart, stats, timechart, sparkline()	

Event order functions

These functions return events based on chronological or timestamp order.

Function	Description	Commands	Examples
`earliest(X)`	Returns the chronologically earliest seen occurrence of a value of a field X. Processes field values as strings.	chart, stats, timechart	
`first(X)`	Returns the first seen value of the field X. In general, the first seen value of the field is the most recent instance of this field, relative to the input order of events into the stats command. • To locate the first value based on time order, use the `earliest` function. • Works best when the search includes the `sort` command immediately before the statistics or charting command. • Processes field values as strings.	chart, stats, timechart	
`last(X)`	Returns the last seen value of the field X. In general, the last seen value of the field is the oldest instance of this field relative to the input order of events into the stats command. • To locate the last value based on time order, use the `latest` function. • Works best when the search includes the `sort` command immediately before the statistics or charting command. • Processes field values as strings.	chart, stats, timechart	
`latest(X)`	Returns the chronologically latest seen occurrence of a value of a field X.	chart, stats, timechart	

	Processes field values as strings.		

Multivalue functions

Function	Description	Commands	Examples
`list(X)`	Returns the list of all values of the field X as a multivalue entry. The order of the values reflects the order of input events. Processes field values as strings.	chart, stats, timechart	
`values(X)`	Returns the list of all distinct values of the field X as a multivalue entry. The order of the values is lexicographical. Processes field values as strings.	chart, stats, timechart	

Time functions

Function	Description	Commands	Examples
`per_day(X)`	Returns the values of field X per day.	`timechart`	This example returns the values of "total" per day. `per_day(total)`
`per_hour(X)`	Returns the values of field X per hour.	`timechart`	This example returns the values of "total" per hour. `per_hour(total)`
`per_minute(X)`	Returns the values of field X per minute.	`timechart`	This example returns the values of "total" per minute. `per_minute(total)`
`per_second(X)`	Returns the values of field X per second.	`timechart`	This example returns values of "kb" per second: `per_second(kb)`

See also

Evaluation functions, stats, chart, timechart, eventstats, streamstats, geostats