Introduction to histograms and heatmaps

介绍直方图和热点图

 › [Getting started](https://grafana.com/docs/grafana/latest/getting-started/) › Intro to histograms

A histogram is a graphical representation of the distribution of numerical data. It groups values into buckets (sometimes also called bins) and then counts how many values fall into each bucket.

一个直方图，是对于数值化数据的图形化表示。它把数据都分组到桶（有时候叫箱）里，并计算出每个桶里都有多少个数据。

Instead of graphing the actual values, histograms graph the buckets. Each bar represents a bucket, and the bar height represents the frequency (such as count) of values that fell into that bucket’s interval.

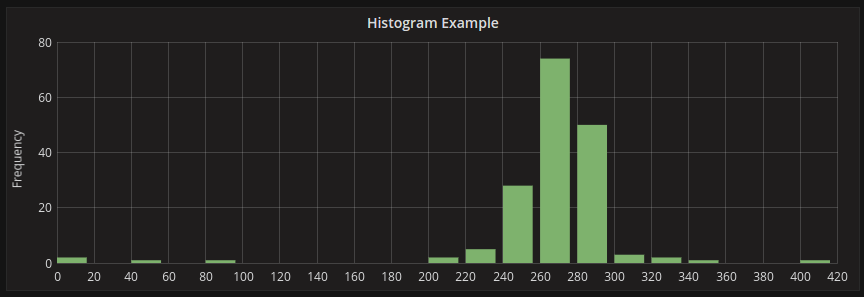
代替了去画出实际的值，直方图画的是桶。每个方块都代表了一个桶，而桶的高度代表了落入到桶的间隔中，值的频率（比如个数）。

Histogram example

直方图例子

This histogram shows the value distribution of a couple of time series. You can easily see that most values land between 240-300 with a peak between 260-280.

这个直方图展示了一组时间序列值的分布。你可以很容易的看出来，大多数据的值都分布在了240-300之间，高峰是在260-280之间。



Histograms only look at *value distributions* over a specific time range. The problem with histograms is you cannot see any trends or changes in the distribution over time. This is where heatmaps become useful.

直方图只能看出来指定时间范围内值的分布。直方图的问题是，你不能随着时间看出数据的任何走向和改变。这就是热点图有用的地方。

Heatmaps

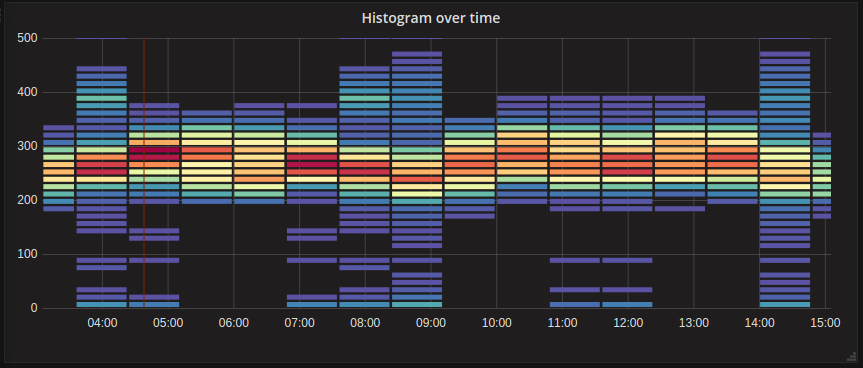
热点图

A *heatmap* is like a histogram, but over time where each time slice represents its own histogram. Instead of using bar height as a representation of frequency, it uses cells and colors the cell proportional to the number of values in the bucket.

热点图很像是直方图，但是随着时间的推移，每个时间片代表它自己的直方图。而不是使用方块来代表着频率，它使用单元格，并根据桶中值的数量为单元格着色。

In this example, you can clearly see what values are more common and how they trend over time.

在这个例子中，你可以很清楚的看出来，随着时间的推移，哪些值是最普遍的，趋势是怎么样的。



Pre-bucketed data

预装数据

There are a number of data sources supporting histogram over time like Elasticsearch (by using a Histogram bucket aggregation) or Prometheus (with [histogram](https://prometheus.io/docs/concepts/metric_types/#histogram) metric type and *Format as* option set to Heatmap). But generally, any data source could be used if it meets the requirements: returns series with names representing bucket bound or returns series sorted by the bound in ascending order.

有不少的数据源都支持时间化的直方图，像Elasticsearch（使用直方图来聚合）或Prometheus（使用直方图型的指标和格式，并将热点图做为备选）。但通常来说，如果遇到以下的需求时，任何数据源都是可以使用的：返回使用names来做为边界的series，或返回升序的series。

Raw data vs aggregated

原始数据 vs 聚合

If you use the heatmap with regular time series data (not pre-bucketed), then it’s important to keep in mind that your data is often already aggregated by your time series backend. Most time series queries do not return raw sample data but include a group by time interval or maxDataPoints limit coupled with an aggregation function (usually average).

如果你使用有规律的时间序列数据来做热点图，要注意你的数据经常被时间序列聚合过了。大多数的时间序列查询不会返回原始示例数据，但是会返回一个按时间间隔的集合，或使用聚合函数（通常为平均值）产生的最大数据值。

This all depends on the time range of your query of course. But the important point is to know that the histogram bucketing that Grafana performs might be done on already aggregated and averaged data. To get more accurate heatmaps it is better to do the bucketing during metric collection or store the data in Elasticsearch, or in the other data source which supports doing histogram bucketing on the raw data.

当然，这完全取决于你查询的时间范围。但重要的是要知道，Grafana所做的直方图可能是在已经聚合和平均的数据上完成的。为了获得更准确的热点图，最好是在指标收集期间进行映射，或将数据存储在Elasticsearch中，或存储在支持对原始数据进行直方图映射的其他数据源中。

If you remove or lower the group by time (or raise maxDataPoints) in your query to return more data points your heatmap will be more accurate but this can also be very CPU and memory taxing for your browser and could cause hangs and crashes if the number of data points becomes unreasonably large.

如果在你查询返回更多的数据点时，你删除或降低了时间的分组（或提高了最大的数据点），你的热点图会更精确，但是当你的数据量变的非常大的时候，会让你的浏览器使用更多的CPU和内存，导致挂起或崩溃。