

Stackdriver Monitoring in the Cloud Console is now Generally Available and the default option. For a limited period of time, you also have the option to use the classic driver Monitoring console. For more information, see [Monitoring in the Cloud Console](/monitoring/docs/monitoring_in_console) (/monitoring/docs/monitoring\_in\_console).

This page explains how to read metric data, also called time-series data, using the [timeSeries.list method](/monitoring/api/ref_v3/rest/v3/projects.timeSeries/list) (/monitoring/api/ref\_v3/rest/v3/projects.timeSeries/list) in the Monitoring API. There are several ways to use this method:

- To run the `list` method without writing any code, the examples in the **PROTOCOL** tabs on this page use the forms-based API Explorer. (See [API Explorer](/monitoring/api/apis-explorer) (/monitoring/api/apis-explorer) for more information on this tool.)
- To learn how to use the `list` method from selected programming languages, see the runnable code samples on this page.
- To view the metrics for a monitored resource using Metrics Explorer, do the following:

1. In the Google Cloud Console, go to **Monitoring** or use the following button:

[Go to Monitoring](https://console.cloud.google.com/monitoring) (https://console.cloud.google.com/monitoring)

2. If **Metrics Explorer** is shown in the navigation pane, click



**Metrics Explorer**. Otherwise, select  **Resources** and then select **Metrics Explorer**.

3. Enter the monitored resource name in the **Find resource type and metric** text box.

For an introduction to metrics and time series, see [Metrics, Time Series, and Resources](/monitoring/api/v3/metrics) (/monitoring/api/v3/metrics).

Each call of the [timeSeries.list method](/monitoring/api/ref_v3/rest/v3/projects.timeSeries/list) (/monitoring/api/ref\_v3/rest/v3/projects.timeSeries/list) can return any number of time series from a single metric type. For example, if you are using Google Compute Engine, then the `compute.googleapis.com/instance/cpu/usage_time` metric type has a separate time series for each of your VM instances.

You specify which time series data you want by supplying the following:

- A filter expression that specifies the metric type. Optionally, the filter selects a subset of the metric's time series by specifying the resources producing the time series or specifying values for certain labels in the time series.
- A time interval that limits how much data is returned.
- Optionally, a specification of how to combine multiple time series to produce an aggregate summary of the data. For more information, see [Aggregating data](#) (#aggregating) for some examples.

You specify which time series to retrieve by passing a [time series filter](/monitoring/api/v3/filters#time-series-filter) (/monitoring/api/v3/filters#time-series-filter) to the `list` method. Following are the commonly-provided filter components:

- The filter must specify a single metric type. For example:

To retrieve custom metrics, change the `metric.type` prefix in the **filter** to `custom.googleapis.com` or another prefix if used; `external.googleapis.com` is frequently used.

- The filter can specify values for the metric's dimension labels. The metric type determines which labels are present. For example:

Note that `label` is correct, although the actual metric object uses `labels` as its key.

- The filter can limit the time series to those that contain a specific monitored resource type:

The filter components can be combined into a single time series filter, such as the following:

If you do not specify values for all the metric labels, then the `list` method returns a time series for each combination of values in the unspecified label(s). The method returns only time series that have data.

You must specify **`interval.startTime`** and **`interval.endTime`** when calling the `timeSeries.list` method. The resulting time period includes the end time but *not* the start time, unless they are both the same time. This can be confusing. For example, consider these examples of `(start, end]` intervals:

`(T, T+1]`

This interval *does not* contain T.

`(T-1, T]`

This interval *does* contain T.

`(T, T]`

As a special and unusual case, this interval includes *only* the time T. If you omit the start time, this is the interval you get.

Start and end times must be specified as strings in RFC 3339 format. For example:

The `date -Iseconds` command on Linux is useful for generating timestamps.

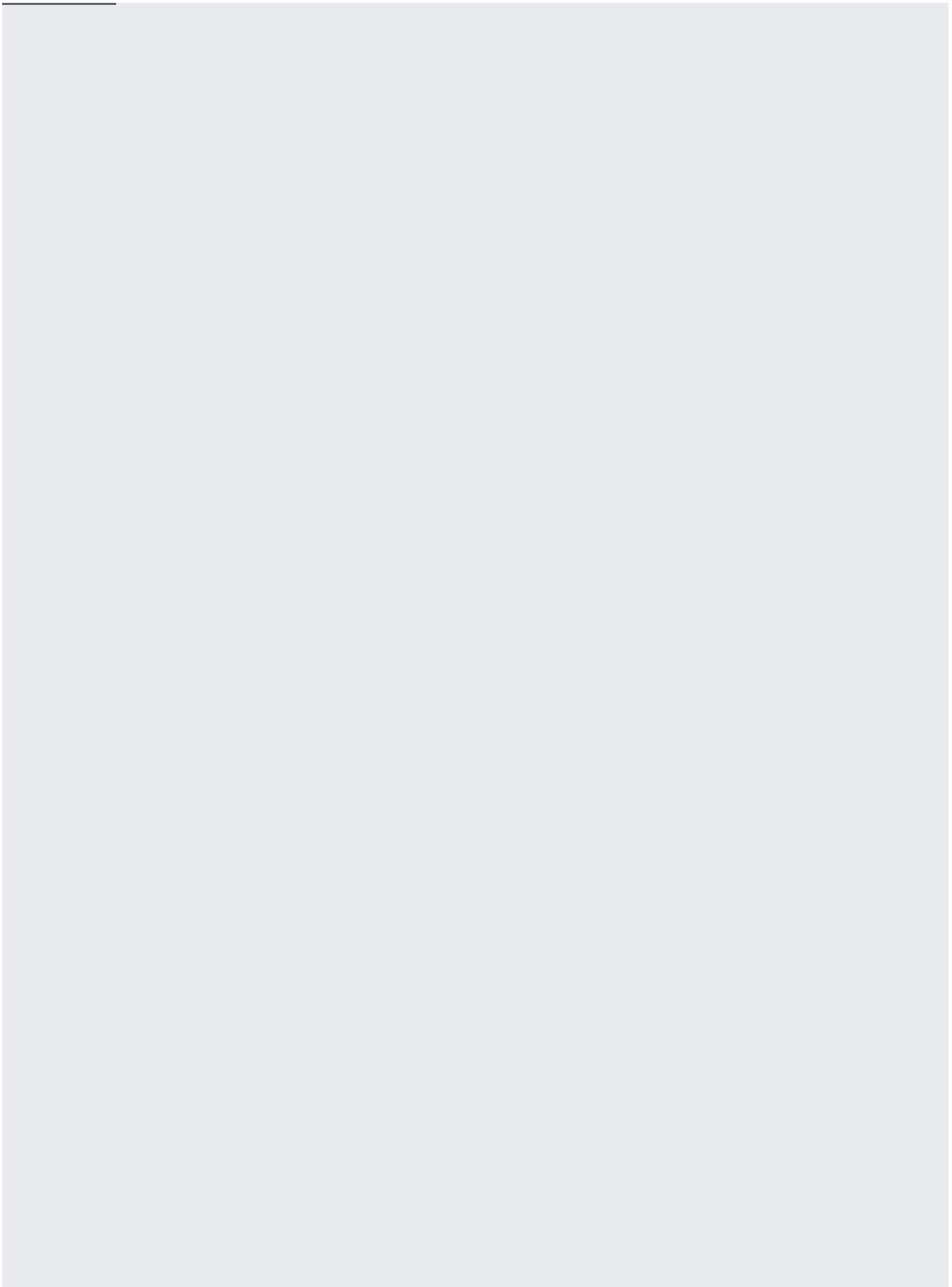
A time interval is specified by a start and end time, but the start time is not required by the API. If a start time is not specified, it defaults to the end time. This makes semantic sense only for GAUGE metrics, which measure points in time.

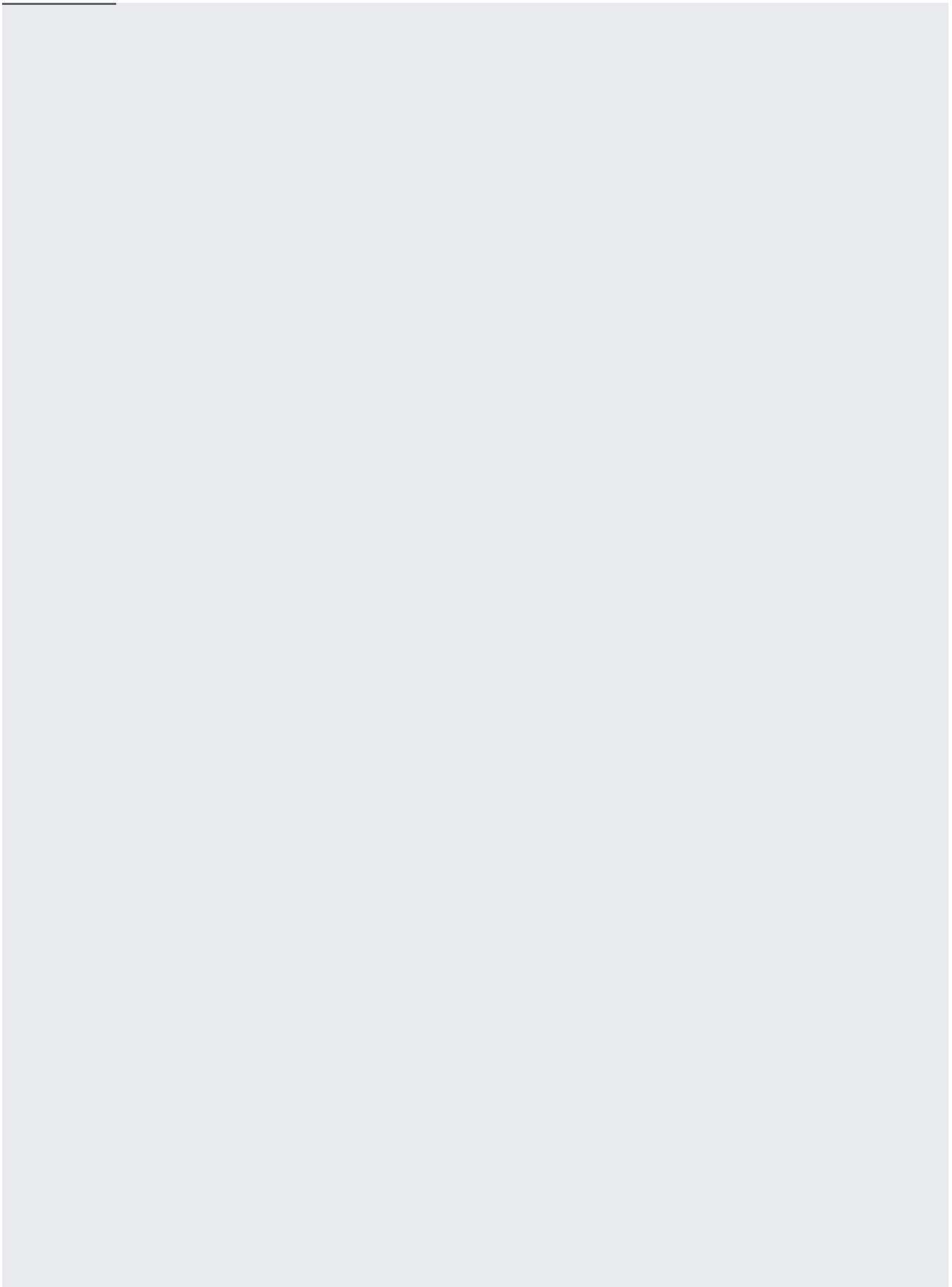
If your metric is CUMULATIVE or DELTA, then it measures an accumulation or change over time. For “over time” metrics, the start and end times of the interval must both be supplied, and the start time must be less than the end time.

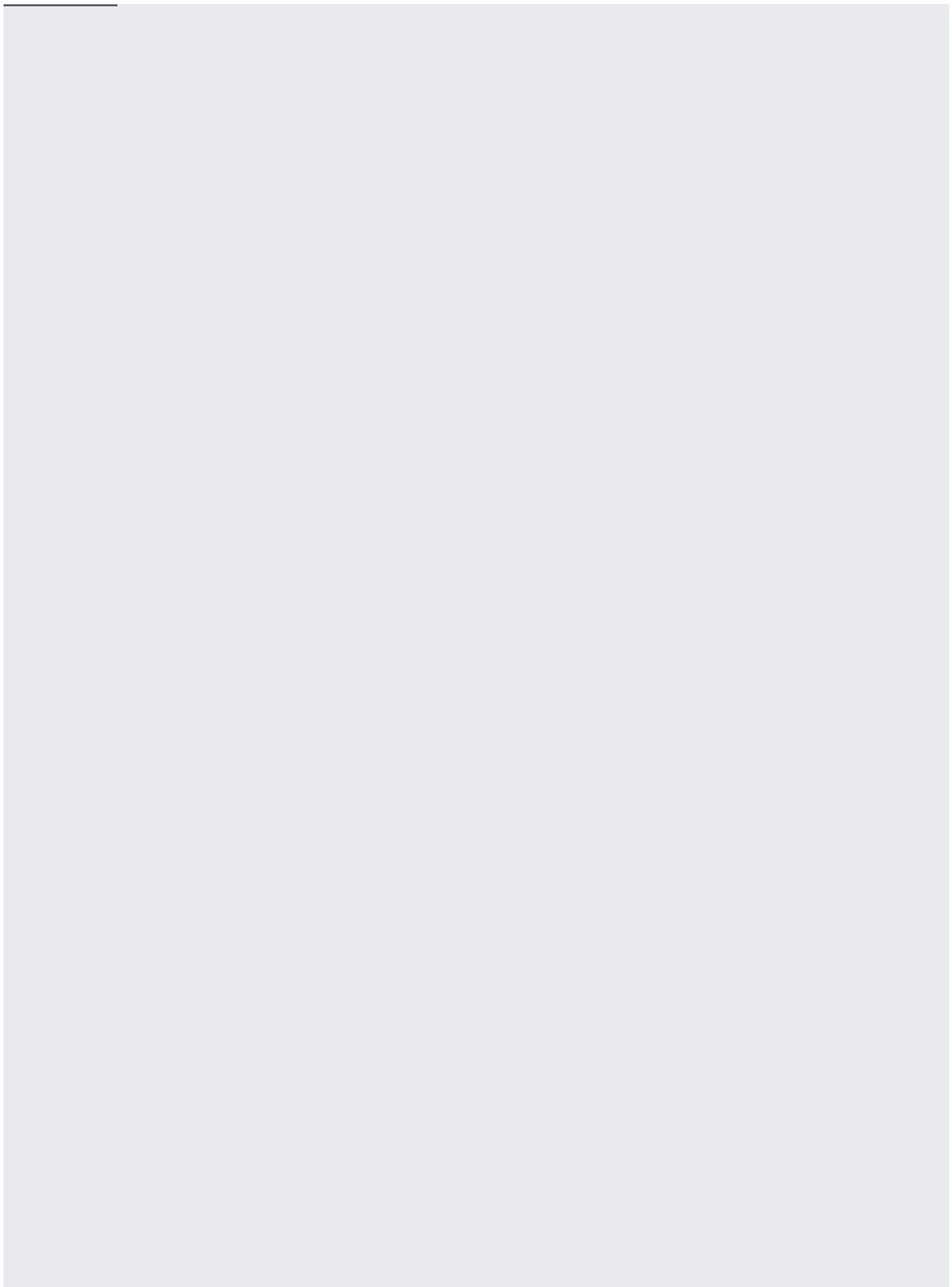
See [Kinds of metrics](/monitoring/api/v3/metrics-details#metric-kinds) (/monitoring/api/v3/metrics-details#metric-kinds) for more information.

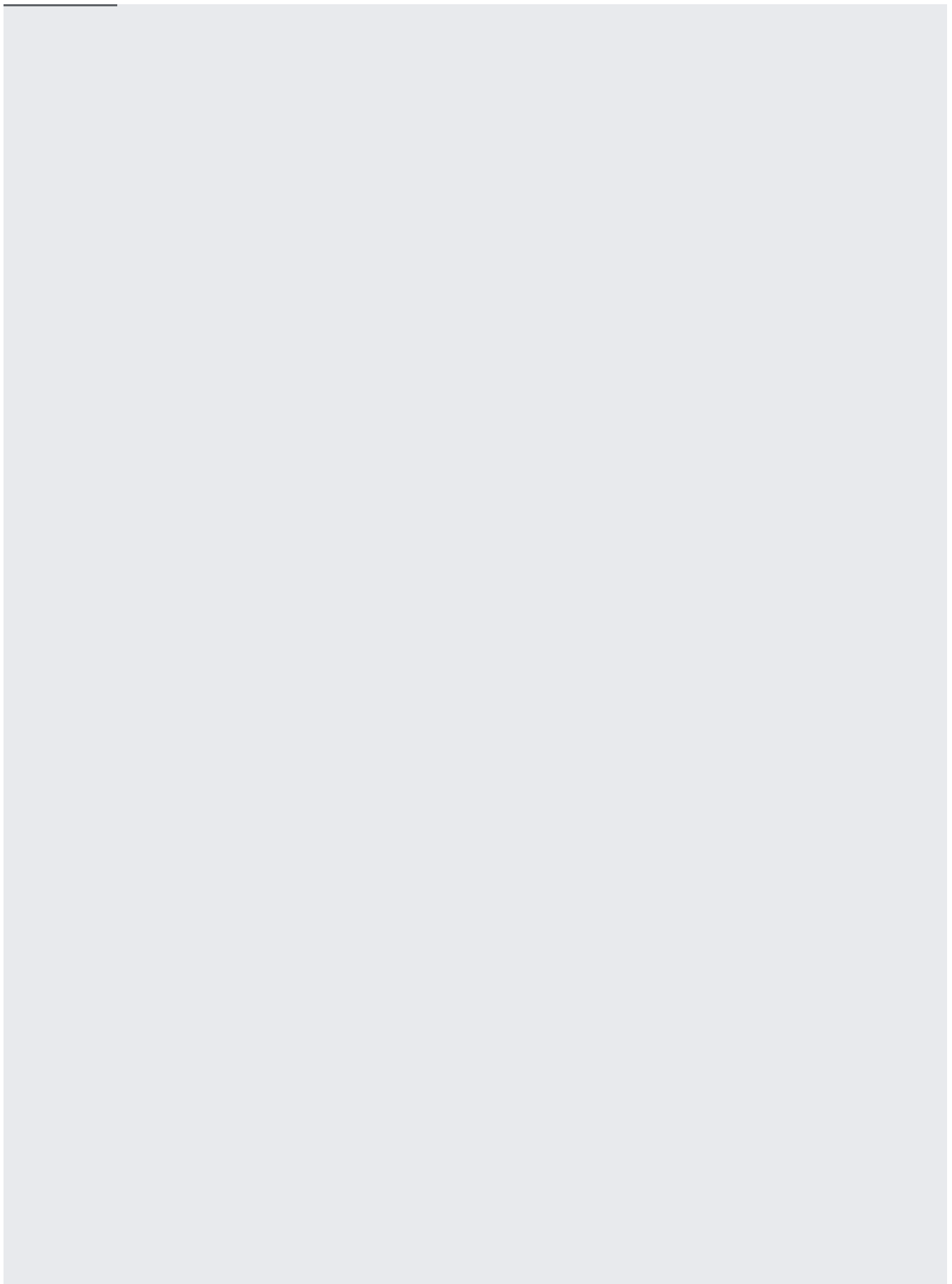
The `timeSeries.list` method can be used to return simple, raw data, or it can be used to return highly processed data. This section illustrates some basic uses.

This example shows how to list only the names and descriptions of the time series that match a filter, rather than returning all the available data:





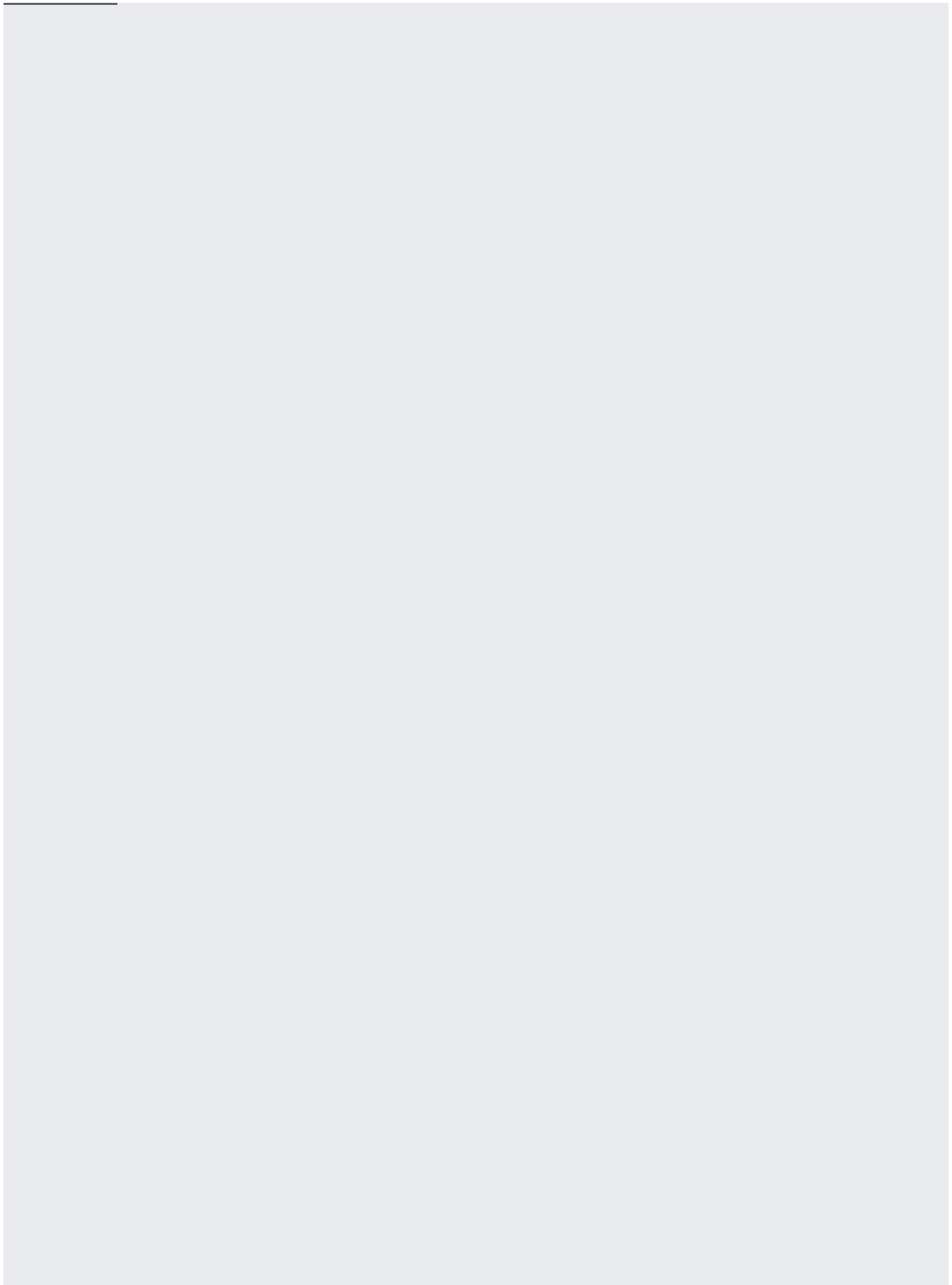


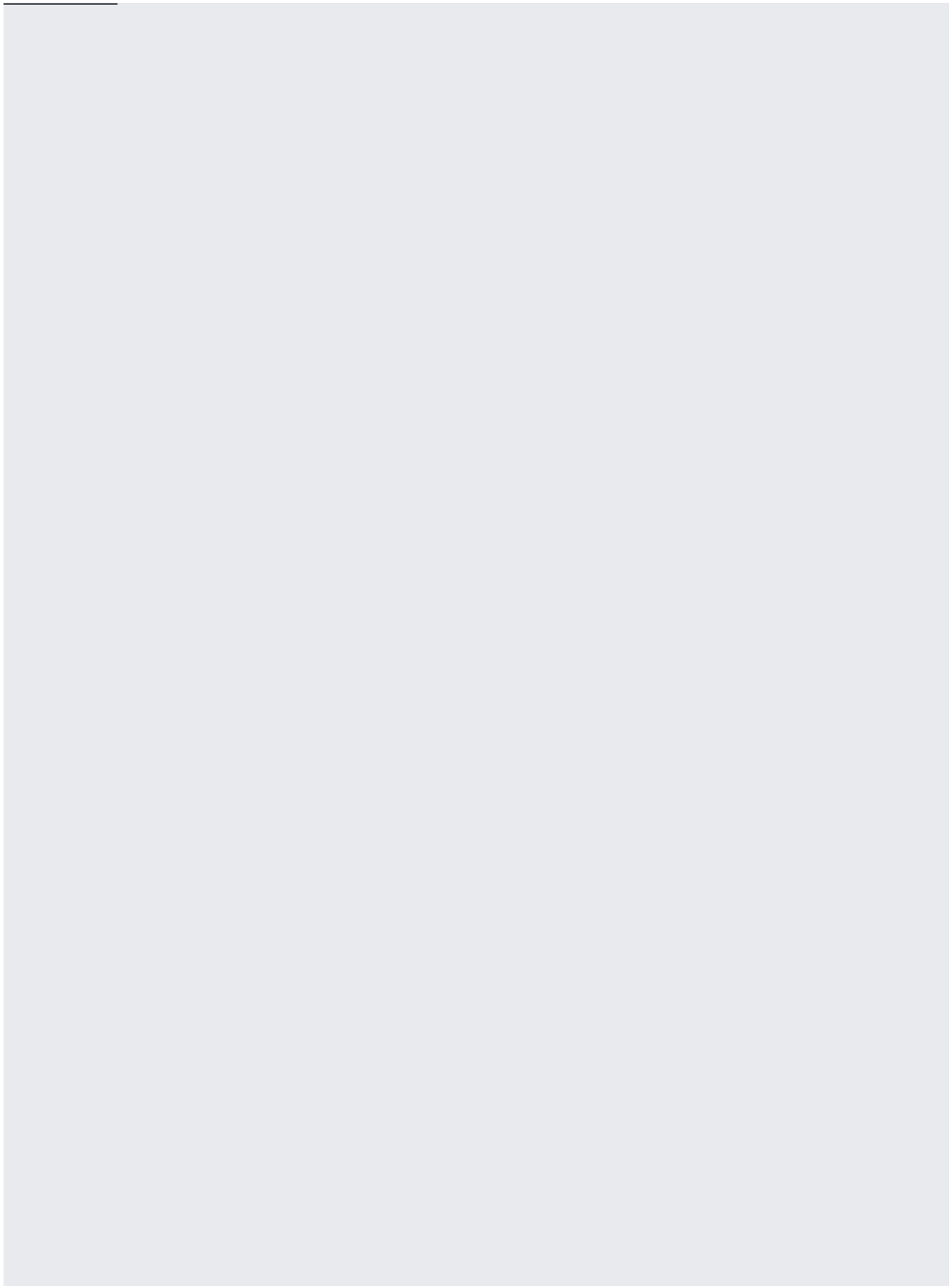


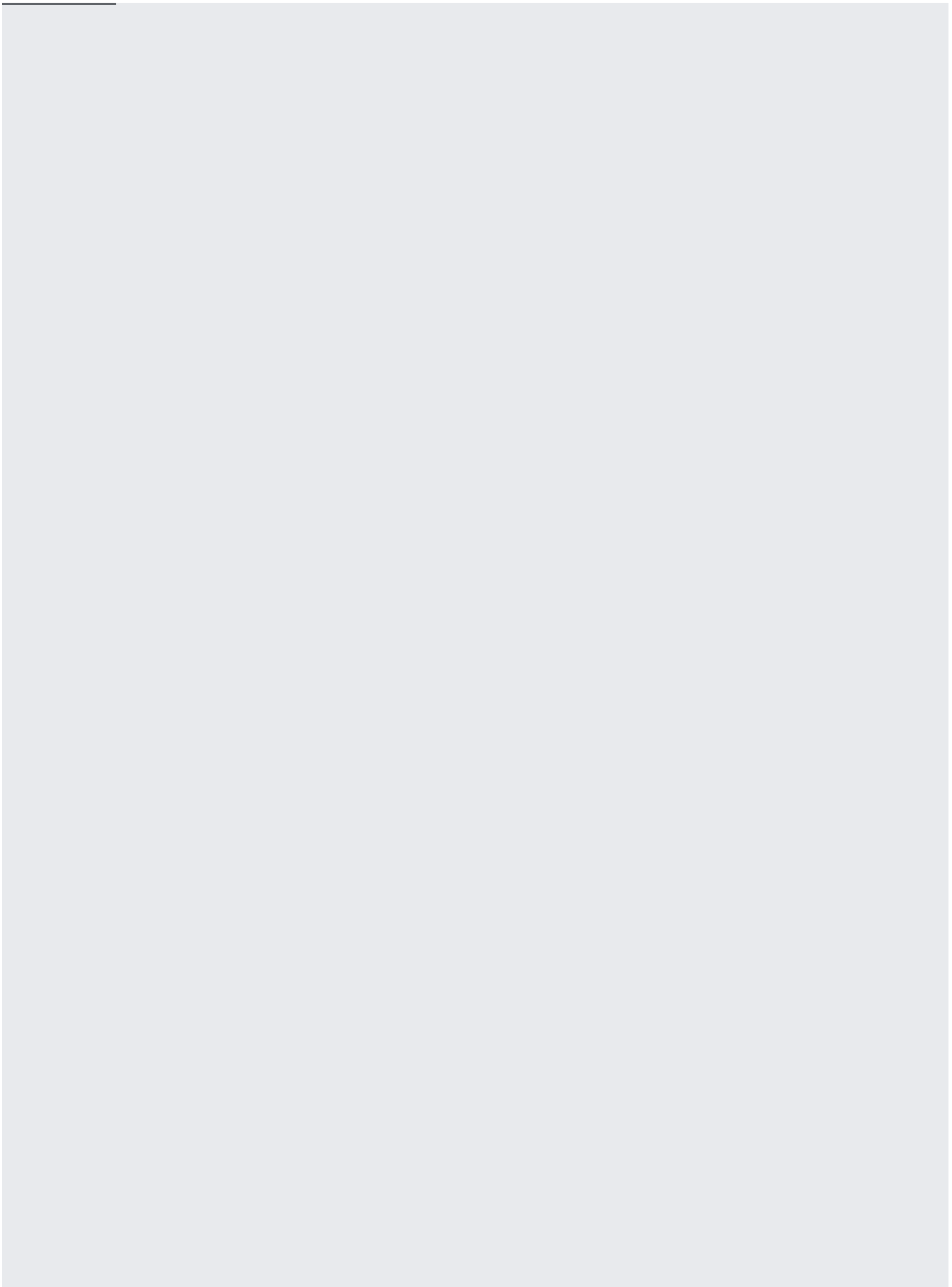
See [Troubleshooting API calls \(/monitoring/api/troubleshooting\)](/monitoring/api/troubleshooting) if you have difficulty.

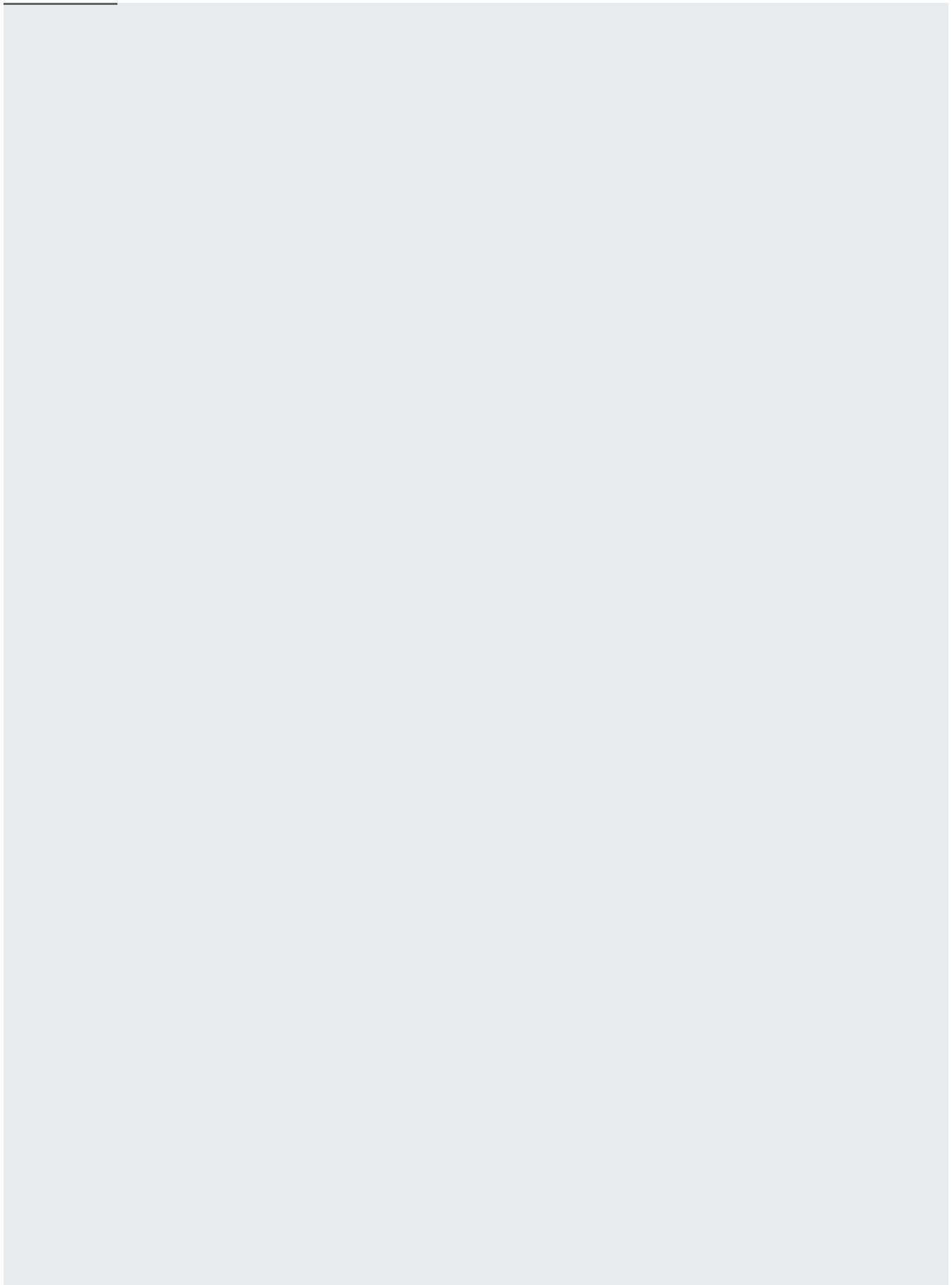
This example returns all the information available to the `timeSeries.list` request, including the metric data, from Compute Engine instances for the last 20 minutes.











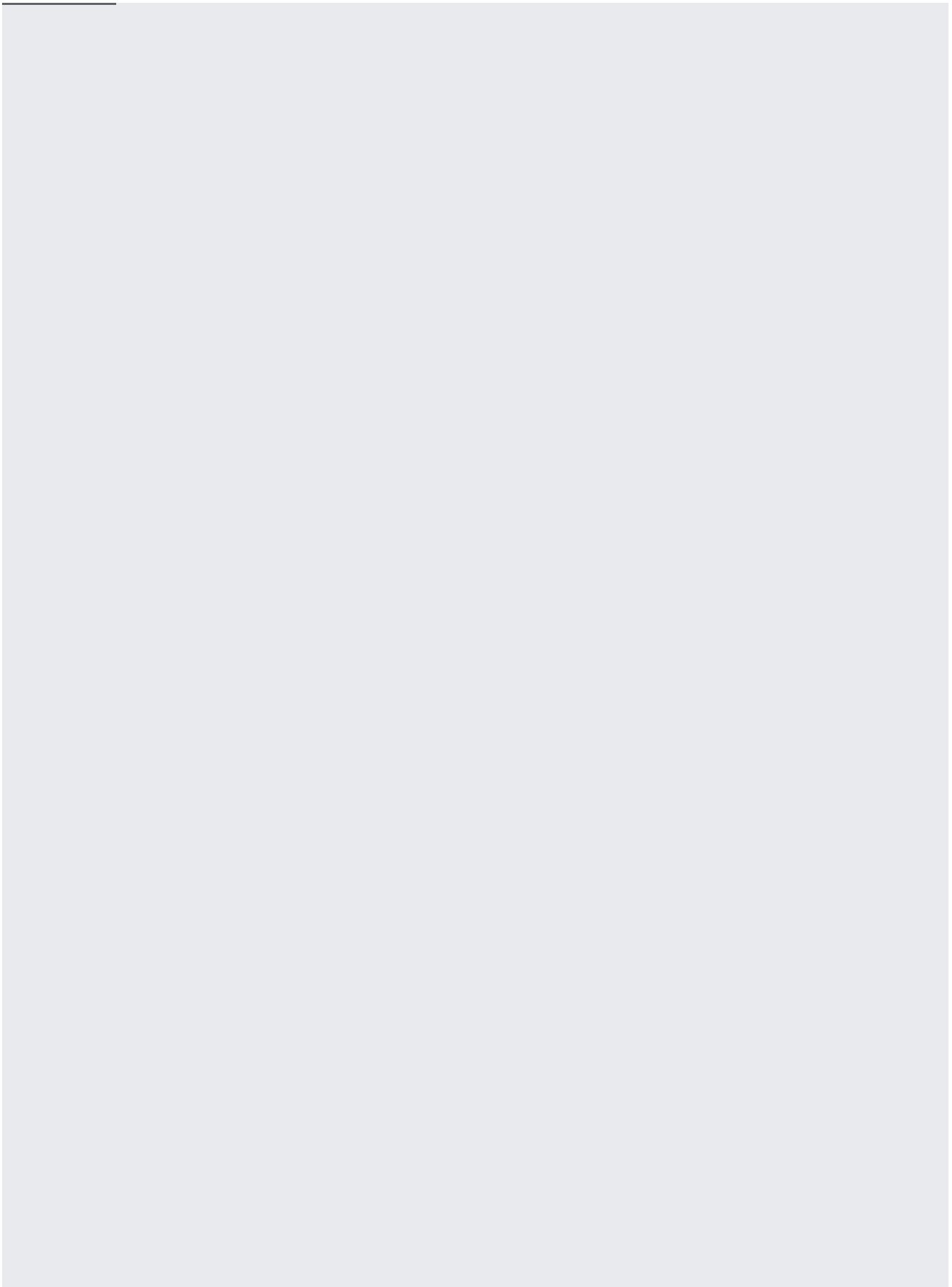
The returned data includes 20 data points in each time series over the 20-minute period, because Compute Engine metrics are collected every minute. For more information, see [Latency of metric data](/monitoring/api/v3/metrics-details#latency) (/monitoring/api/v3/metrics-details#latency). The API returns the data points in each time series in reverse time order; there is no override for this point ordering.

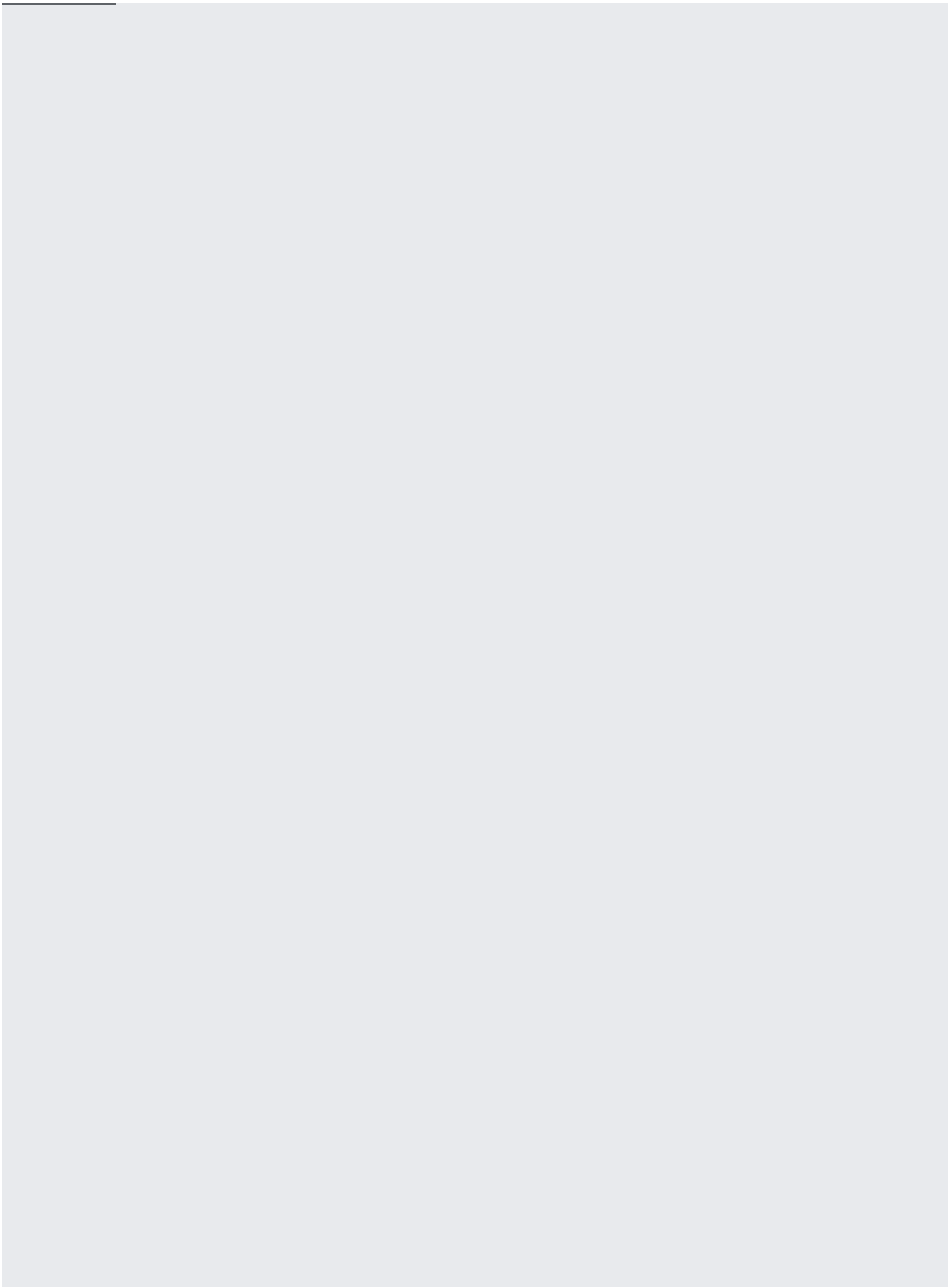
See [Troubleshooting API calls](/monitoring/api/troubleshooting) (/monitoring/api/troubleshooting) if you have difficulty.

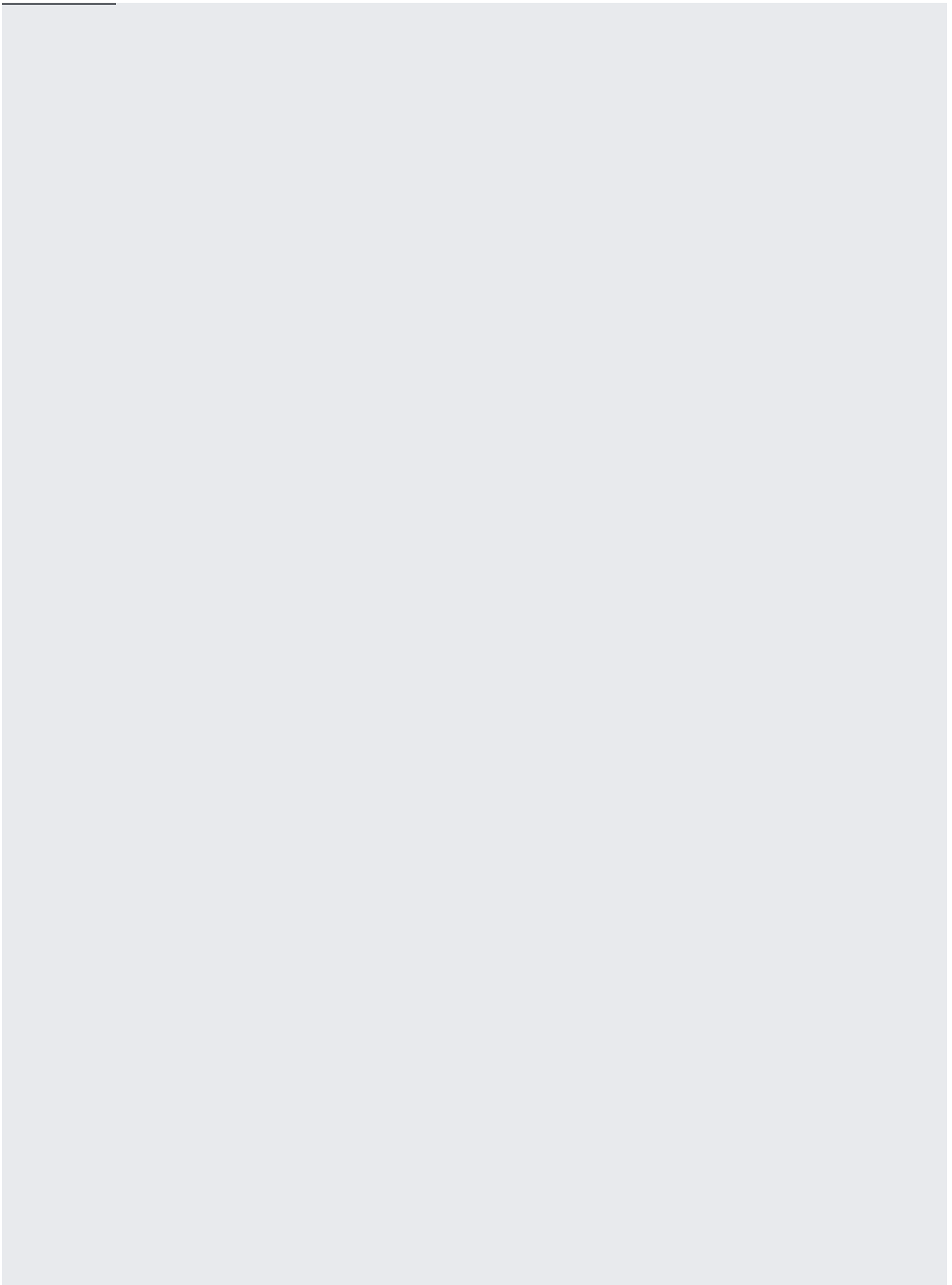
The [timeSeries.list](/monitoring/api/ref_v3/rest/v3/projects.timeSeries/list) (/monitoring/api/ref\_v3/rest/v3/projects.timeSeries/list) method can perform statistical aggregations and reductions on the returned time series data. The following sections demonstrate two examples; see the method's documentation for more options.

This example reduces the 20 individual utilization measurements in each time series to just two measurements: the mean utilization for the two 10-minute periods within the 20-minute interval. The data from each time series is first aligned into 10-minute (600-second) periods, and then the values in each 10-minute period are averaged.

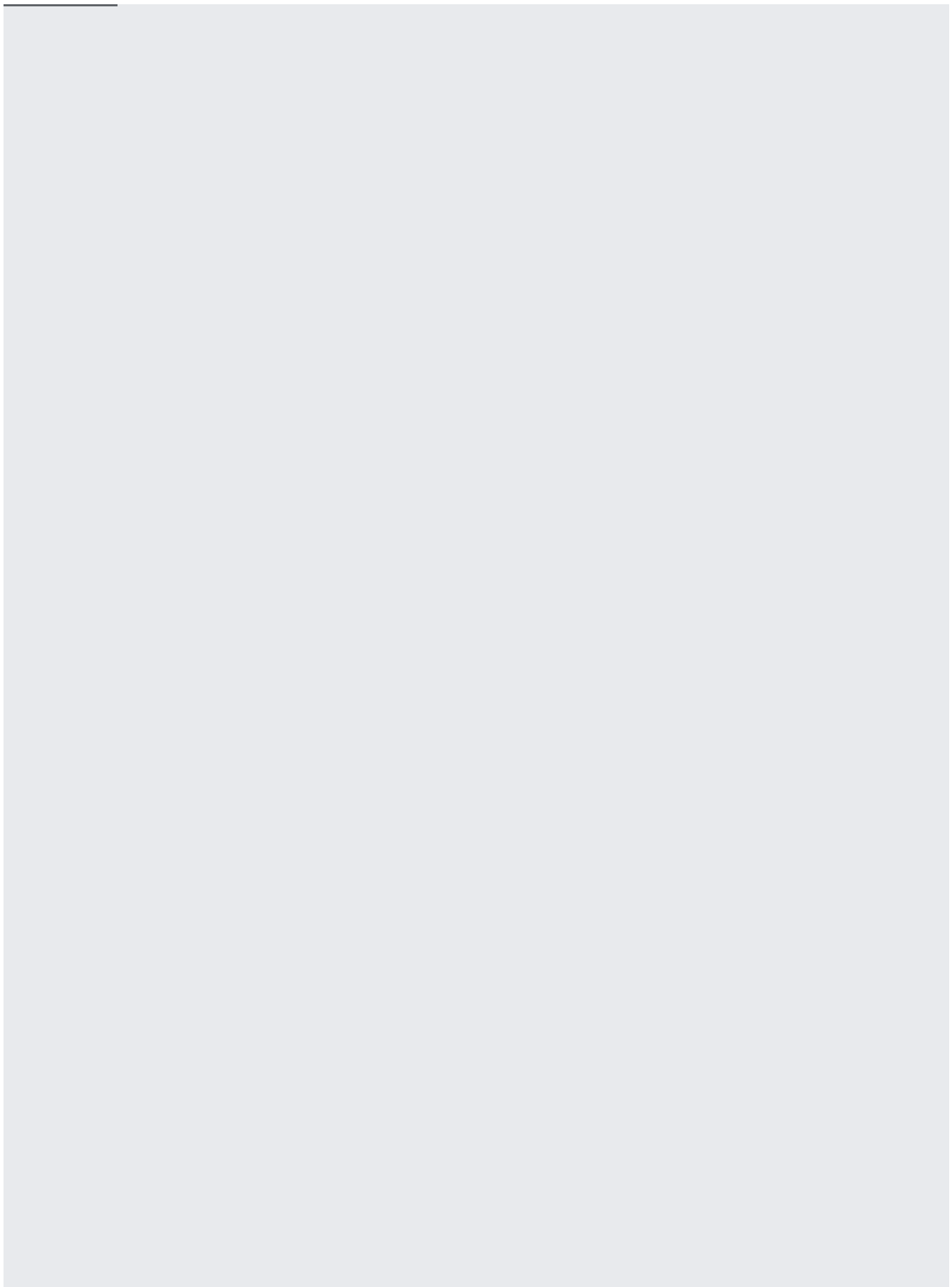
This example turns the twenty measurements per time series into two per time series. This operation has two advantages: it smooths out the data, and it aligns the data from all of the time series on exact 10-minute boundaries. The data can then be processed further.

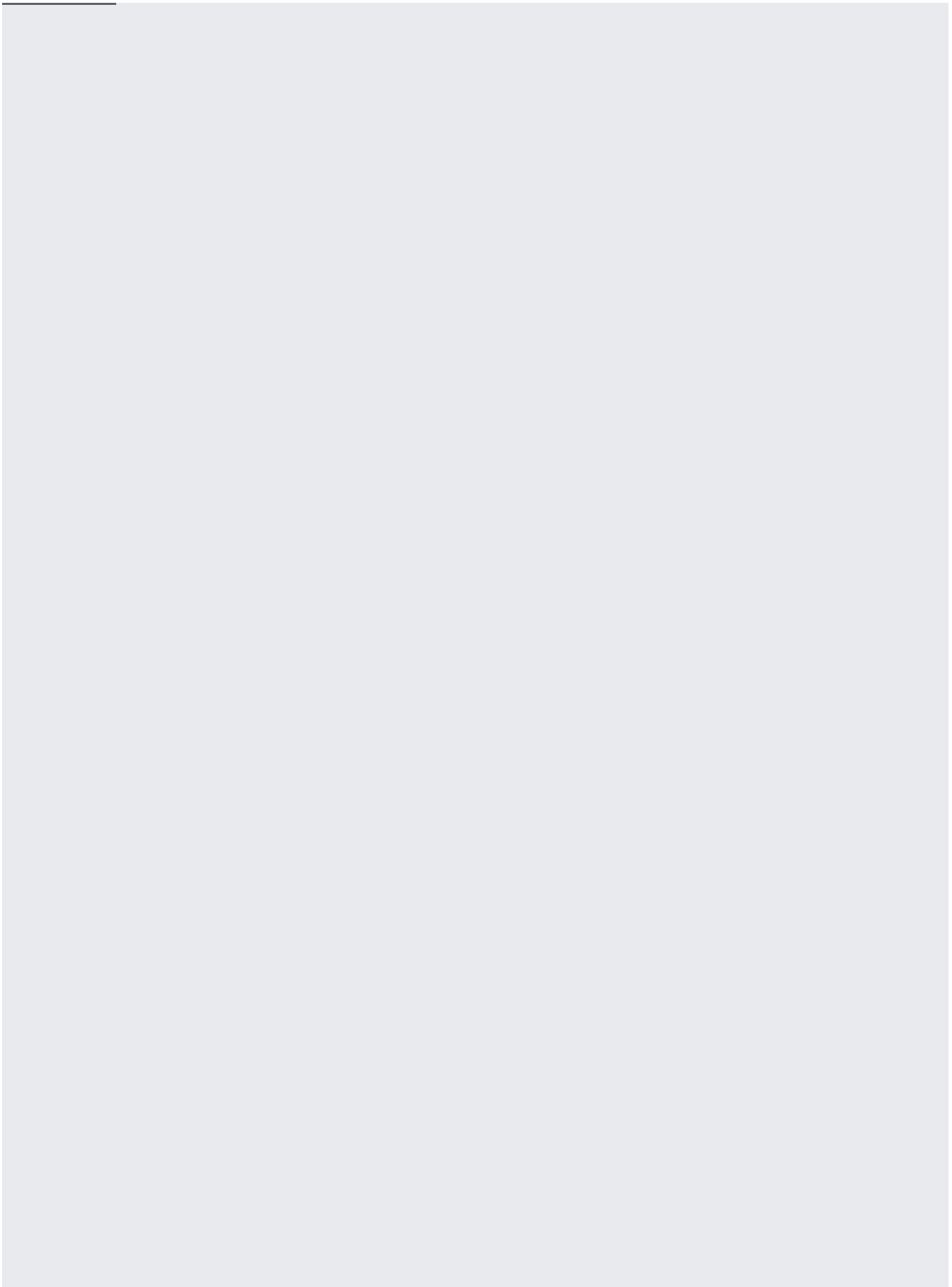


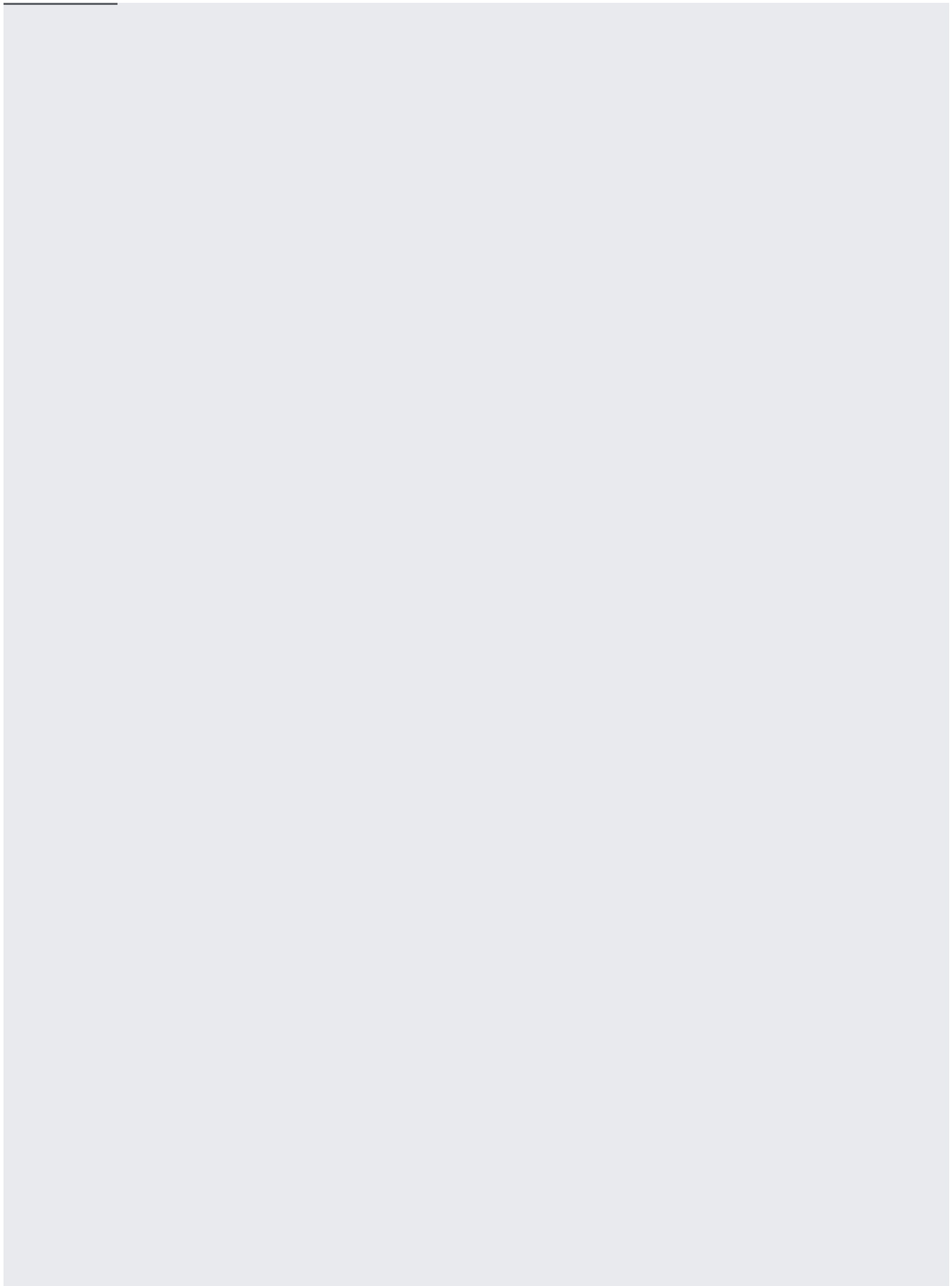






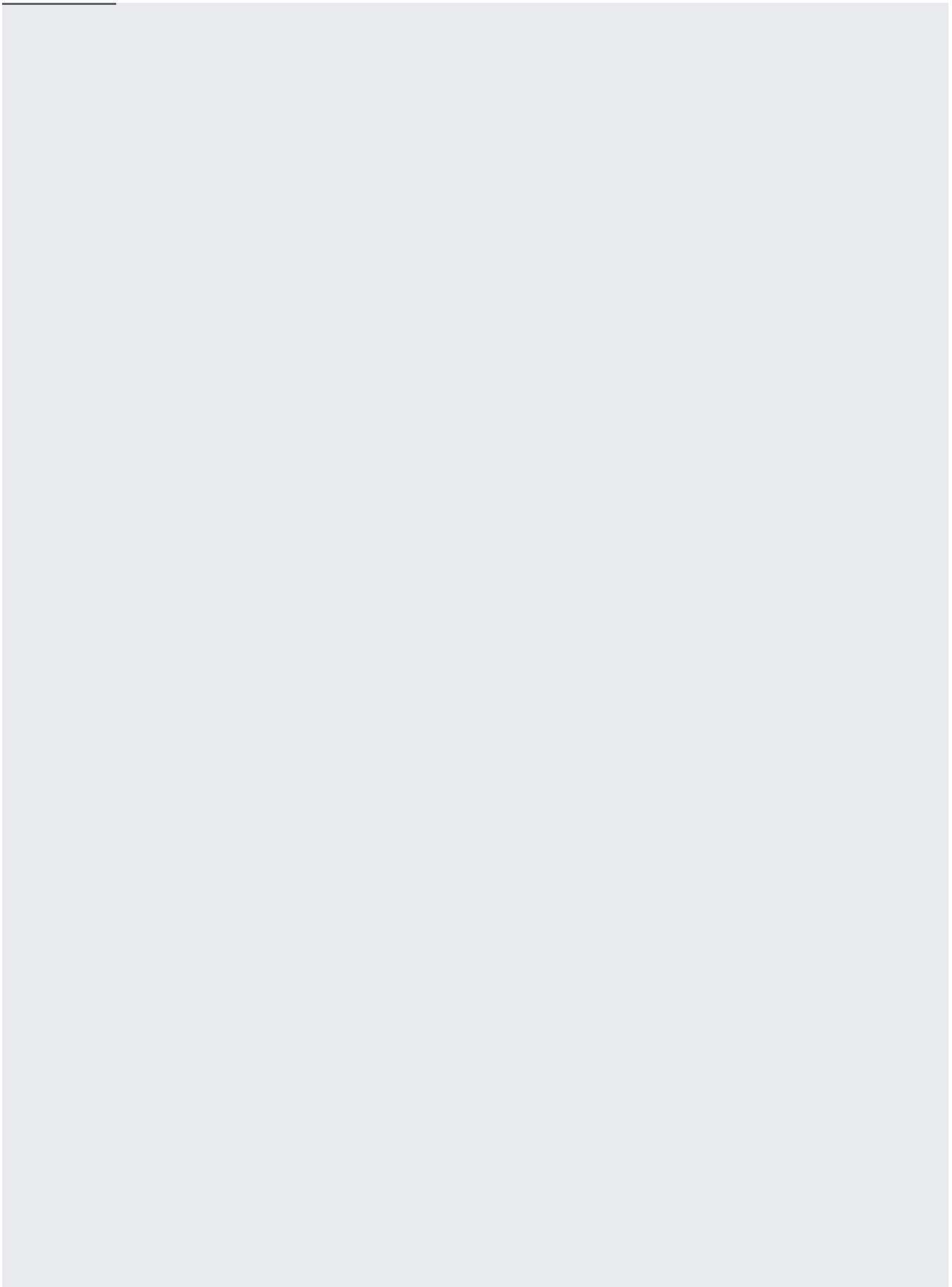


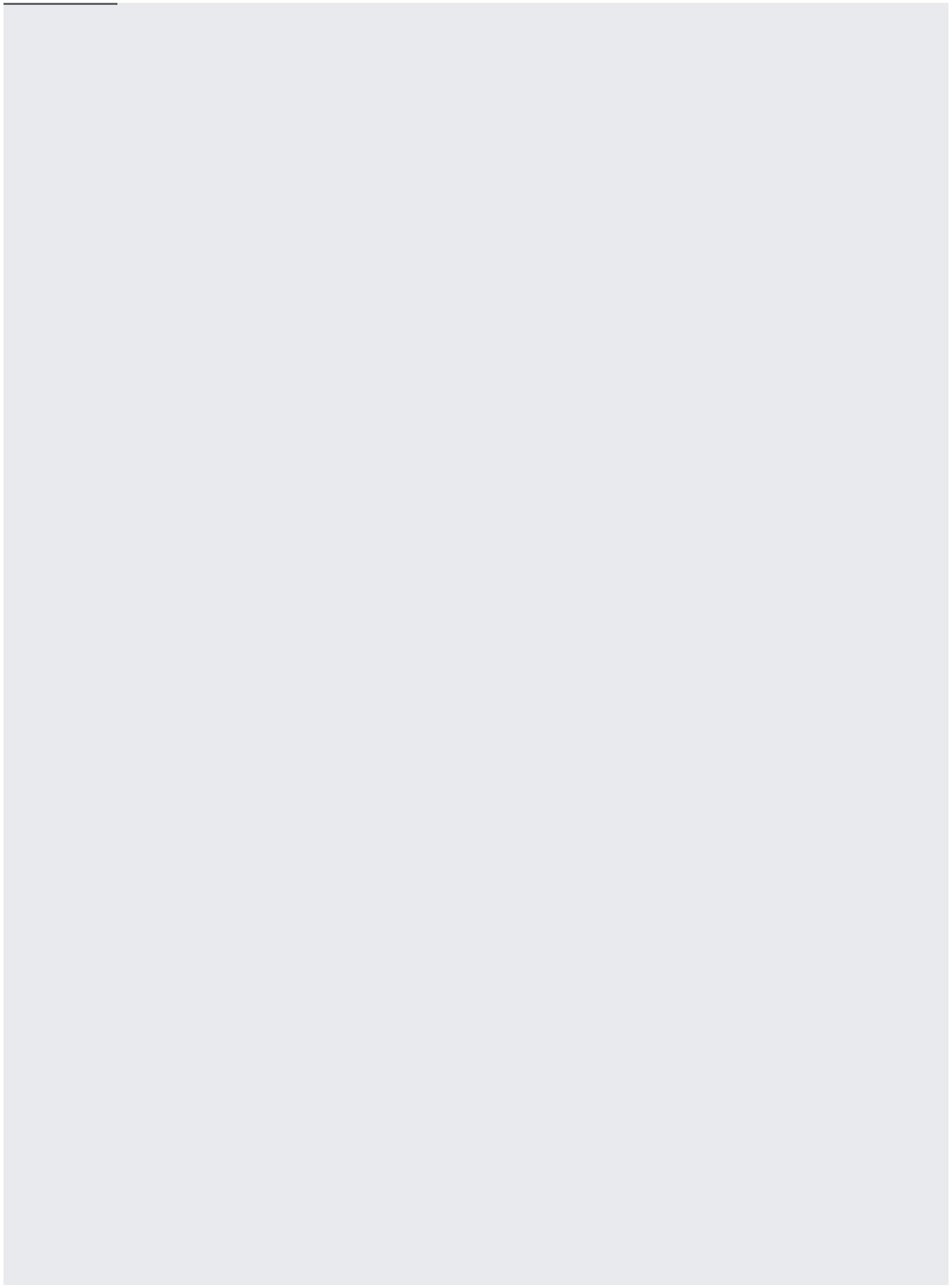


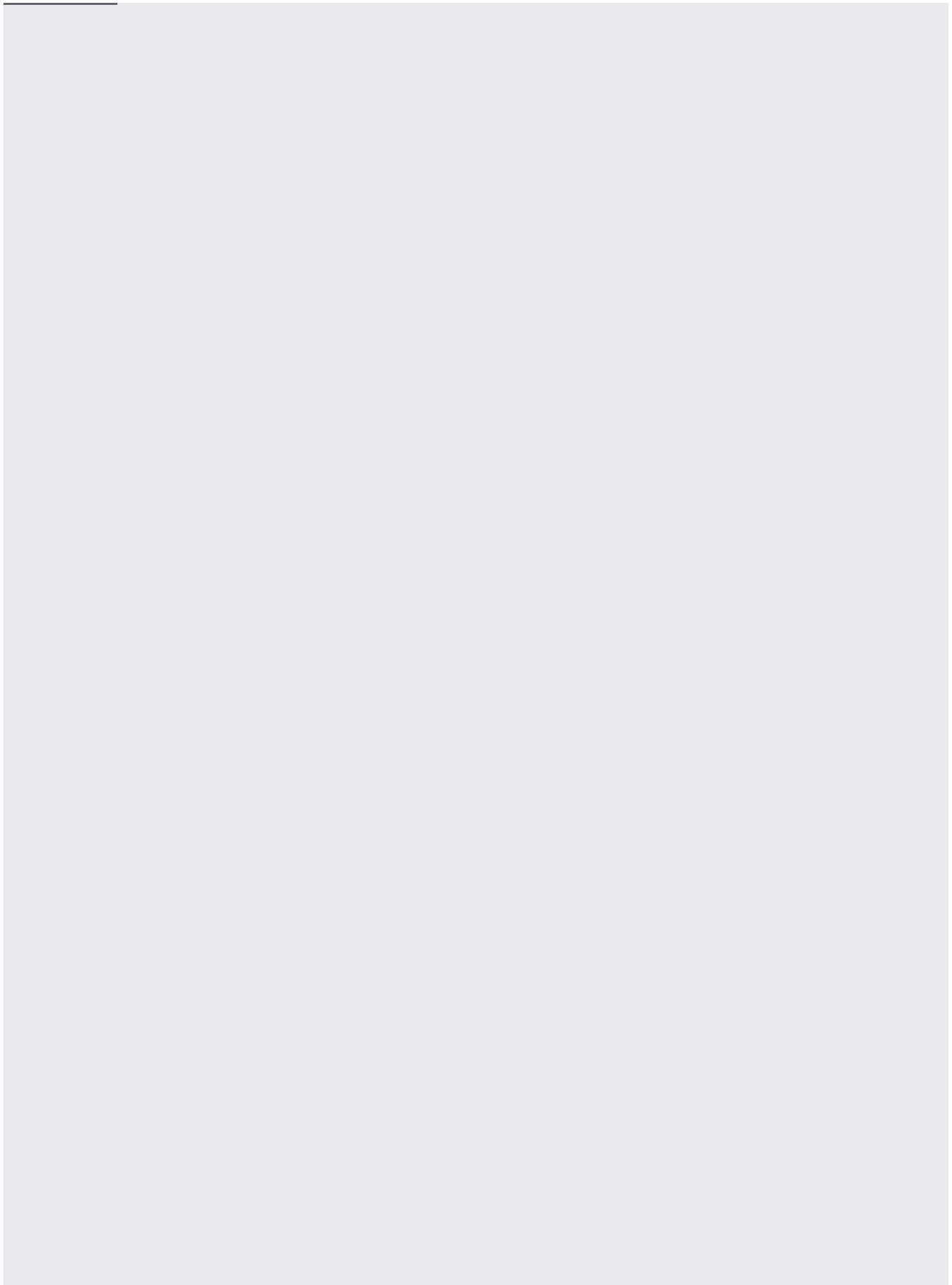


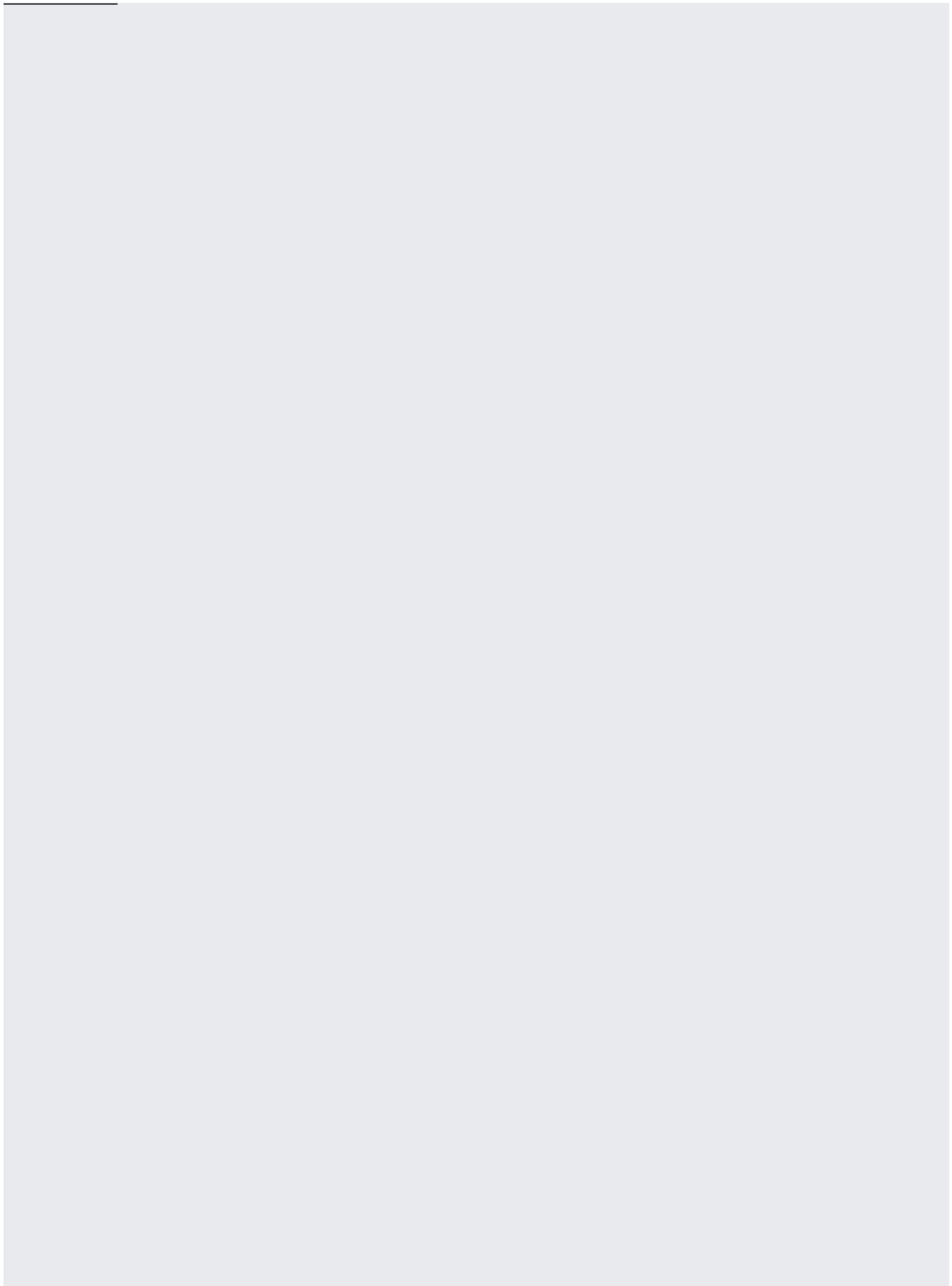
See [Troubleshooting API calls \(/monitoring/api/troubleshooting\)](/monitoring/api/troubleshooting) if you have difficulty.

This example extends the previous example by combining the aligned time series from the three VM instances into a single time series that measures the average utilization of all instances.

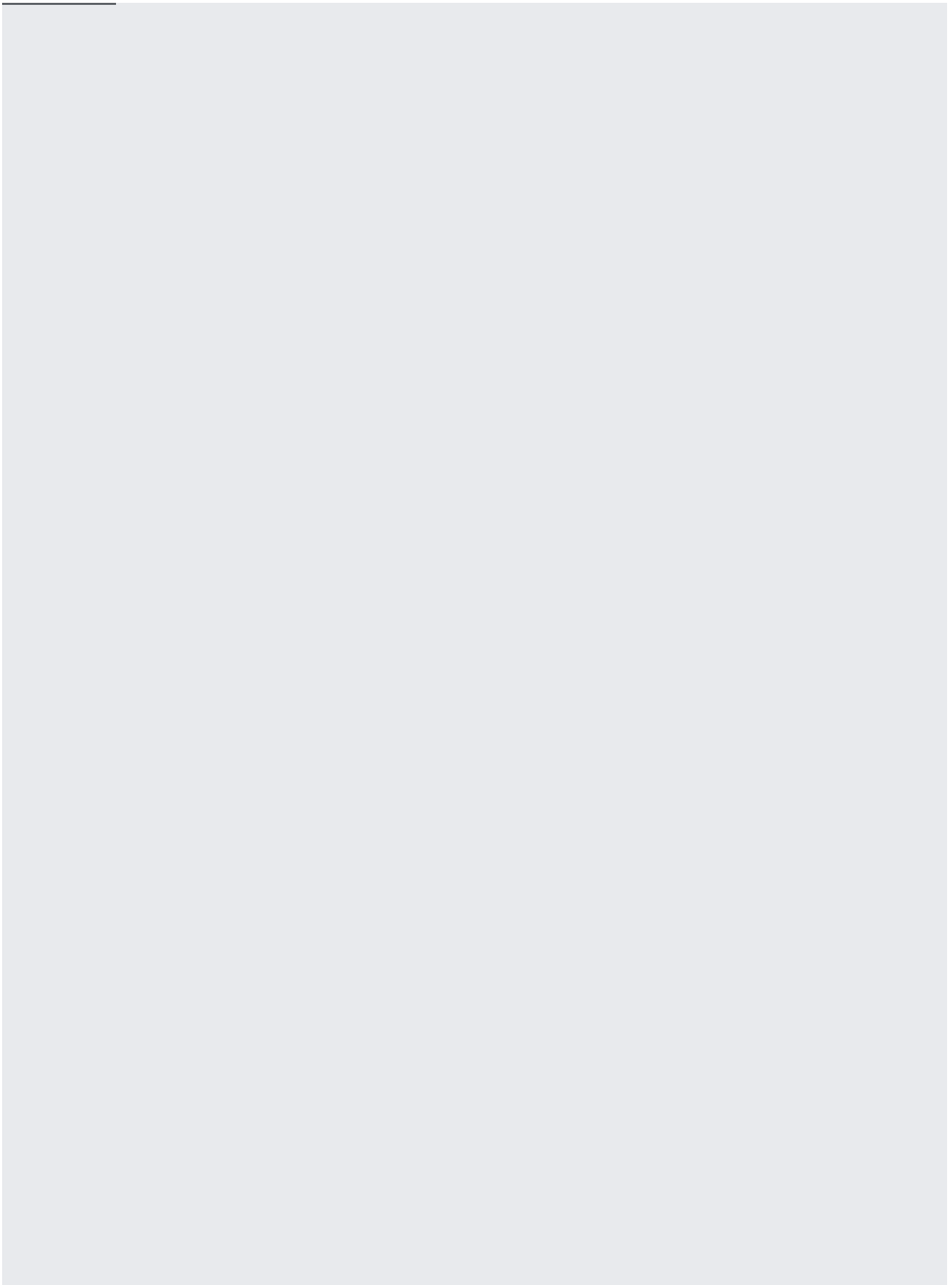


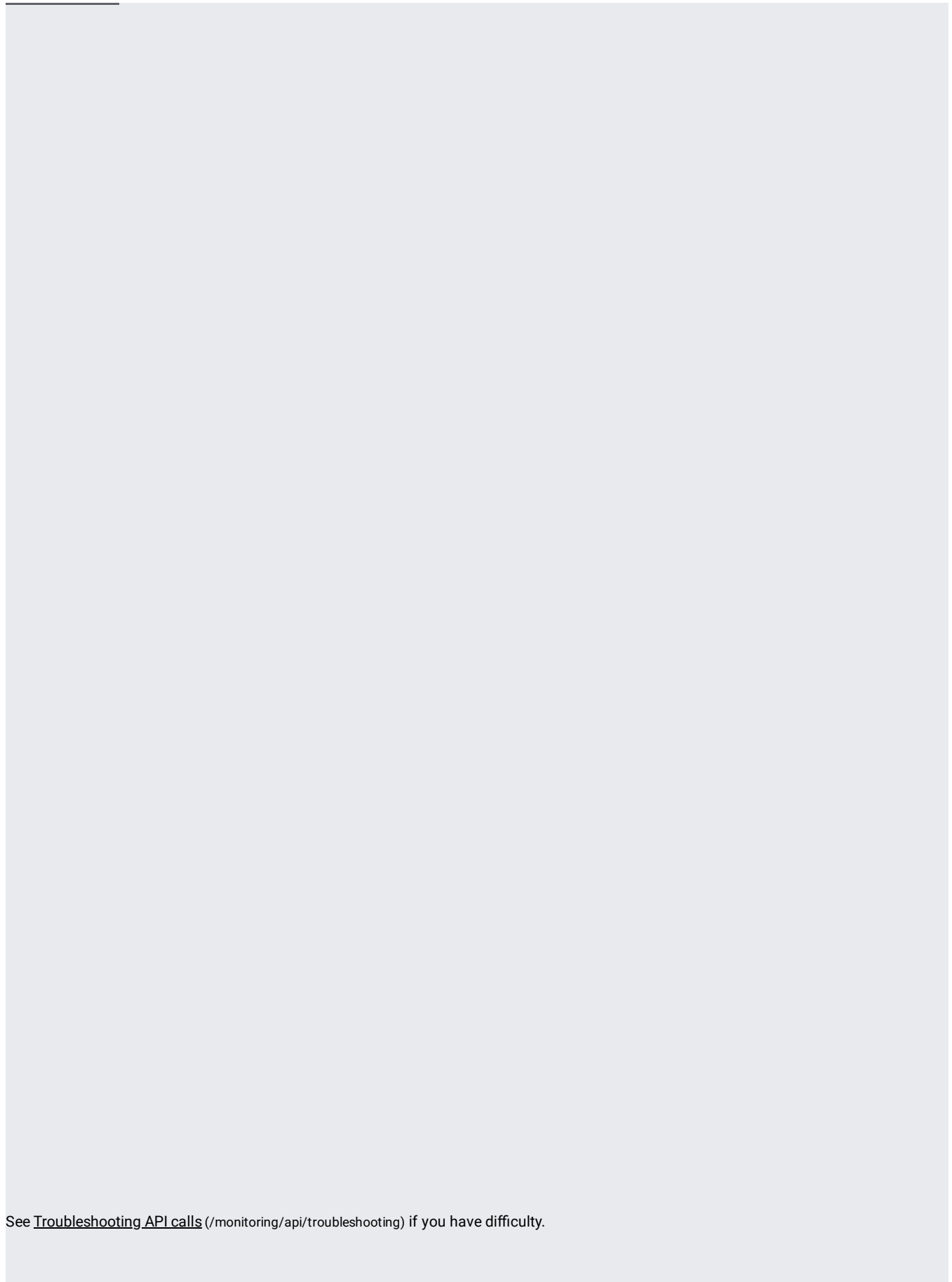












See [Troubleshooting API calls \(/monitoring/api/troubleshooting\)](/monitoring/api/troubleshooting) if you have difficulty.

