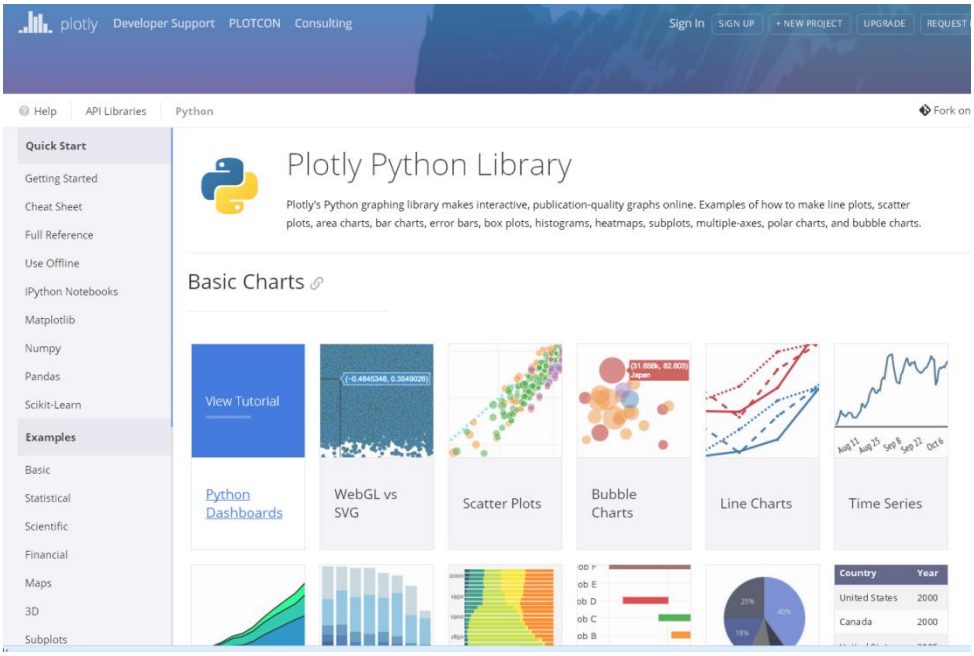
TopQuant.vip 极宽量化开源课件系列文库

《Plotly 绘图模块参考手册》

英文版中文目录

作者：TopQuant 极宽量化开源团队

2016,12,25 圣诞节



前言

尽管 plotly2016 年 6 月，才正式发布 python-api 文档，尽管 2017 年 1 月，plotly 1.0 才正式发布。

但 plotly，以其强度无比的功能，特别是强大的 web 交互式绘图功能，彻底替代了 matplotlib、seaborn、bokeh 等多种传统的 Python 每天模块，成为新一代的绘图模块当中的王中之王。

如今，很多人都听说了 plotly 的大名，很多人都渴望早日接触、早日使用 plotly 绘图包括。

不过，也有很少的一部分人，在默默 的工作，从 2016 年圣诞、2017 年元旦、2017 年春节。

这些很少的很少的一部分网友，默默地利用自己的节假日时间，默默地引进、翻译 plolty 的各种文档。

这些人，就是 TopQuant 极宽量化开源组的 plotly 翻译小组。

这些人是中国 Quant 的 Top 团队。

plolty 的文档翻译，可能是极宽量化开源团队，面对的最大一次挑战，时间方面，恰遇新年，文档规模，更是超过了前面的各种课件。

为此，极宽开源组只能采用折中的办法，分期进行翻译。

如今，第一期文档已经完成，中英双语，超过 240 页的 16 开版面，完全是一部图书的尺度。

何海群

TopQuant 极宽量化开源团队发起人

更多介绍请参见：

Plotly 互动型 python 绘图神器 - plotly 与数据可视化 - http://ziwang.com/forum.php?mod=viewthread&tid=41&extra=page%3D1

TopQuant-极宽量化社区（原 zw 量化），国内第一家专业 Python 量化社区

附录：任务分配

plotly 翻译任务分几期进行，这次为第一期，仅仅进行最简单的翻译任务。只翻译 getting_start, basic, 与 Financial Charts （plotl offline 部分后来由 youngle sunny 1535327967 个人兴趣而临时加进去）这三部分内容。

任务分配如下：

组长：

youngle sunny 1535327967 basic 1,2 + 任务分配+ 校对，组长

成员：

余勤 441499022 校对 + 机动安排任务

leon, 华子（尹少华 32509167） geting_start 1/2

啦啦啦 505512828 geting_start 2/2

禛 948280670 basic 5,6

L. 1248515039 basic7,8

Rikimaru 11766429 basic 9,10

iris 704699640 13,14

信平 759949947 basic 3,4 + 校对

吴娜 2184934 basic20-2 + 校对 + 机动安排任务
周涛 510548099 11,12
zw 木子 719735825 basic 15,16
非洲兔 85011284 basic 17,18
十二月 378258849 basic 19,20-1
大朱 775941748 Financial Charts 1
我爱作文你信吗 571171954 Financial Charts 2

补充其他：

Youngle sunny 1535327967 plot offline，该部分内容不在本次翻译计划范围之内，属于个人兴趣之作。

plotly 绘图模块

- Pricing
- PLOTCON OAK
- API

- Sign In
- SIGN UP
- + NEW PROJECT
- UPGRADE
- REQUEST DEMO

- Feed
- Pricing
- Make a Chart
- API

- Sign In
- SIGN UP
- + NEW PROJECT
- UPGRADE
- REQUEST DEMO

Show Sidebar Hide Sidebar

- Help
- API Libraries
- Python

Fork on Github
Plot Objects

- Scatter
- Bar
- Box
- Heatmap
- Histogram
- Histogram 2d
- Area
- Pie
- Contour
- Histogram 2d Contour
- 3D
- Scatter

- Surface
- Mesh
- Maps
- Scatter Geo
- Choropleth
- WebGL
- Scatter

Layout

- Margins
- Width
- Height
- Font
- X Axis:
- Title
- Type
- Ticks
- Range
- Y Axis:
- Title
- Type
- Ticks
- Range
- 3D Scene:
- Camera
- Aspect Ratio
- 3D Axes:
- X
- Y
- Z
- Geo:
- Projection
- Resolution
- Scope
- Geo-Axes:
- Longitude
- Latitude
- Legend
- Annotations
- Shapes

Back to API Index

Python 绘图参考手册

`plotly` charts are described *declaratively* with objects in `plotly.graph_objs` and `dict`. Every aspect of a plotly chart (the colors, the grids, the data, and so on) has a corresponding key-value attribute in these objects. This page contains an extensive list of these attributes.

Plotly's graph description places attributes into two categories: traces (objects that describe a single series of data in a graph like `Scatter` or `Heatmap`) and `layout` attributes that apply to the rest of the chart, like the `title`, `xaxis`, or `annotations`).

Here is a simple example of a plotly chart inlined with links to each attribute's reference section.

```
import plotly
```

```
import plotly.plotly as py
import plotly.graph_objs as go
print plotly.__version__                    # At least 1.8.6 is required. Upgrade with: $ pip install plotly --upgrade
data = [
    go.Scatter(                             # all "scatter" attributes: https://plot.ly/python/reference/#scatter
        x=[1, 2, 3],                        # more about "x":  /python/reference/#scatter-x
        y=[3, 1, 6],                        # more about "y":  /python/reference/#scatter-y
        marker=dict(                        # marker is an dict, marker keys: /python/reference/#scatter-marker
            color="rgb(16, 32, 77)"         # more about marker's "color": /python/reference/#scatter-marker-color
        )
    ),
    go.Bar(                             # all "bar" chart attributes: /python/reference/#bar
        x=[1, 2, 3],                    # more about "x": /python/reference/#bar-x
        y=[3, 1, 6],                    # /python/reference/#bar-y
        name="bar chart example"        # /python/reference/#bar-name
    )
]
layout = go.Layout(                 # all "layout" attributes: /python/reference/#layout
    title="simple example",         # more about "layout's" "title": /python/reference/#layout-title
    xaxis=dict(                     # all "layout's" "xaxis" attributes: /python/reference/#layout-xaxis
        title="time"                # more about "layout's" "xaxis's" "title": /python/reference/#layout-xaxis-title
    ),
    annotations=[
        dict(                               # all "annotation" attributes: /python/reference/#layout-annotations
            text="simple annotation",       # /python/reference/#layout-annotations-text
            x=0,                            # /python/reference/#layout-annotations-x
            xref="paper",                   # /python/reference/#layout-annotations-xref
            y=0,                            # /python/reference/#layout-annotations-y
            yref="paper"                    # /python/reference/#layout-annotations-yref
        )
    ]
```

```
)
figure = go.Figure(data=data, layout=layout)
py.plot(figure, filename='api-docs/reference-graph')
```

散点图

`import plotly.graph_objs as go`
`go.Scatter`
A `Scatter` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- x0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (`number`)
  default: `1`
  Sets the x coordinate step. See `x0` for more info.
- y (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the y coordinates.
- y0 (`number or categorical coordinate string`)
  default: `0`

Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.

- dy (number)
  default: 1
  Sets the y coordinate step. See `y0` for more info.
- ids (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  A list of keys for object constancy of data points during animation
- text (string)
  default: ""
  Sets text elements associated with each (x,y) pair. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y) coordinates.
- mode (flaglist string)
  Any combination of "lines", "markers", "text" joined with a "+" OR "none".
  examples: "lines", "markers", "lines+markers", "lines+markers+text", "none"
  Determines the drawing mode for this scatter trace. If the provided `mode` includes "text" then the `text` elements appear at the coordinates. Otherwise, the `text` elements appear on hover. If there are less than 20 points, then the default is "lines+markers". Otherwise, "lines".
- hoveron (flaglist string)
  Any combination of "points", "fills" joined with a "+"
  examples: "points", "fills", "points+fills"
  Do the hover effects highlight individual points (markers or line points) or do they highlight filled regions? If the fill is "toself" or "tonext" and there are no markers or text, then the default is "fills", otherwise it is "points".
- line
  - color (color)
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - shape (enumerated: "linear" | "spline" | "hv" | "vh" | "hvh" | "vhv")
    default: "linear"
    Determines the line shape. With "spline" the lines are drawn using spline interpolation. The other available values correspond to step-wise line shapes.
  - smoothing (number between or equal to 0 and 1.3)
    default: 1
    Has an effect only if `shape` is set to "spline" Sets the amount of smoothing. "0" corresponds to no smoothing (equivalent to a "linear" shape).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
  - simplify (boolean)
    default: True
    Simplifies lines by removing nearly-collinear points. When transitioning lines, it may be desirable to disable this so that the number of points along the resulting SVG path is unaffected.
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- fill (enumerated: "none" | "tozeroy" | "tozerox" | "tonexty" | "tonextx" | "toself" | "tonext")
  default: "none"
  Sets the area to fill with a solid color. Use with `fillcolor` if not "none". "tozerox" and "tozeroy" fill to x=0 and y=0 respectively. "tonextx" and "tonexty" fill between the endpoints of this trace and the endpoints of the trace before it, connecting those endpoints with straight lines (to make a stacked area graph); if there is no trace before it, they behave like "tozerox" and "tozeroy". "toself" connects the endpoints of the trace (or each segment of the trace if it has gaps) into a closed shape. "tonext" fills the space between two traces if one completely encloses the other (eg consecutive contour lines), and behaves like "toself" if there is no trace before it. "tonext" should not be used if one trace does not enclose the other.
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- marker
  - symbol (enumerated: "0" | "circle" | "100" | "circle-open" | "200" | "circle-dot" | "300" | "circle-open-dot" | "1" | "square" | "101" | "square-open" | "201" | "square-dot" | "301" | "square-open-dot" | "2" | "diamond" | "102" | "diamond-open" | "202" | "diamond-dot" | "302" | "diamond-open-dot" | "3" |

"cross" | "103" | "cross-open" | "203" | "cross-dot" | "303" | "cross-open-dot" | "4" | "x" | "104" | "x-open" | "204" | "x-dot" | "304" | "x-open-dot" | "5" | "triangle-up" | "105" | "triangle-up-open" | "205" | "triangle-up-dot" | "305" | "triangle-up-open-dot" | "6" | "triangle-down" | "106" | "triangle-down-open" | "206" | "triangle-down-dot" | "306" | "triangle-down-open-dot" | "7" | "triangle-left" | "107" | "triangle-left-open" | "207" | "triangle-left-dot" | "307" | "triangle-left-open-dot" | "8" | "triangle-right" | "108" | "triangle-right-open" | "208" | "triangle-right-dot" | "308" | "triangle-right-open-dot" | "9" | "triangle-ne" | "109" | "triangle-ne-open" | "209" | "triangle-ne-dot" | "309" | "triangle-ne-open-dot" | "10" | "triangle-se" | "110" | "triangle-se-open" | "210" | "triangle-se-dot" | "310" | "triangle-se-open-dot" | "11" | "triangle-sw" | "111" | "triangle-sw-open" | "211" | "triangle-sw-dot" | "311" | "triangle-sw-open-dot" | "12" | "triangle-nw" | "112" | "triangle-nw-open" | "212" | "triangle-nw-dot" | "312" | "triangle-nw-open-dot" | "13" | "pentagon" | "113" | "pentagon-open" | "213" | "pentagon-dot" | "313" | "pentagon-open-dot" | "14" | "hexagon" | "114" | "hexagon-open" | "214" | "hexagon-dot" | "314" | "hexagon-open-dot" | "15" | "hexagon2" | "115" | "hexagon2-open" | "215" | "hexagon2-dot" | "315" | "hexagon2-open-dot" | "16" | "octagon" | "116" | "octagon-open" | "216" | "octagon-dot" | "316" | "octagon-open-dot" | "17" | "star" | "117" | "star-open" | "217" | "star-dot" | "317" | "star-open-dot" | "18" | "hexagram" | "118" | "hexagram-open" | "218" | "hexagram-dot" | "318" | "hexagram-open-dot" | "19" | "star-triangle-up" | "119" | "star-triangle-up-open" | "219" | "star-triangle-up-dot" | "319" | "star-triangle-up-open-dot" | "20" | "star-triangle-down" | "120" | "star-triangle-down-open" | "220" | "star-triangle-down-dot" | "320" | "star-triangle-down-open-dot" | "21" | "star-square" | "121" | "star-square-open" | "221" | "star-square-dot" | "321" | "star-square-open-dot" | "22" | "star-diamond" | "122" | "star-diamond-open" | "222" | "star-diamond-dot" | "322" | "star-diamond-open-dot" | "23" | "diamond-tall" | "123" | "diamond-tall-open" | "223" | "diamond-tall-dot" | "323" | "diamond-tall-open-dot" | "24" | "diamond-wide" | "124" | "diamond-wide-open" | "224" | "diamond-wide-dot" | "324" | "diamond-wide-open-dot" | "25" | "hourglass" | "125" | "hourglass-open" | "26" | "bowtie" | "126" | "bowtie-open" | "27" | "circle-cross" | "127" | "circle-cross-open" | "28" | "circle-x" | "128" | "circle-x-open" | "29" | "square-cross" | "129" | "square-cross-open" | "30" | "square-x" | "130" | "square-x-open" | "31" | "diamond-cross" | "131" | "diamond-cross-open" | "32" | "diamond-x" | "132" | "diamond-x-open" | "33" | "cross-thin" | "133" | "cross-thin-open" | "34" | "x-thin" | "134" | "x-thin-open" | "35" | "asterisk" | "135" | "asterisk-open" | "36" | "hash" | "136" | "hash-open" | "236" | "hash-dot" | "336" | "hash-open-dot" | "37" | "y-up" | "137" | "y-up-open" | "38" | "y-down" | "138" | "y-down-open" | "39" | "y-left" | "139" | "y-left-open" | "40" | "y-right" | "140" | "y-right-open" | "41" | "line-ew" | "141" | "line-ew-open" | "42" | "line-ns" | "142" | "line-ns-open" | "43" | "line-ne" | "143" | "line-ne-open" | "44" | "line-nw" | "144" | "line-nw-open")
default: "circle"
Sets the marker symbol type. Adding 100 is equivalent to appending "-open" to a symbol name. Adding 200 is equivalent to appending "-dot" to a symbol name. Adding 300 is equivalent to appending "-open-dot" or "dot-open" to a symbol name.

- opacity (number between or equal to 0 and 1)
  Sets the marker opacity.
- size (number greater than or equal to 0)
  default: 6
  Sets the marker size (in px).
- maxdisplayed (number greater than or equal to 0)
  default: 0
  Sets a maximum number of points to be drawn on the graph. "0" corresponds to no limit.
- sizeref (number)
  default: 1
  Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
- sizemin (number greater than or equal to 0)
  default: 0
  Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.
- sizemode (enumerated: "diameter" | "area")
  default: "diameter"

Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.

- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.
  - bordercolor (color)
    default: "#444"
    Sets the axis line color.
  - borderwidth (number greater than or equal to 0)
    default: 0
    Sets the width (in px) or the border enclosing this color bar.
  - bgcolor (color)
    default: "rgba(0,0,0,0)"
    Sets the color of padded area.
  - tickmode (enumerated: "auto" | "linear" | "array")
    Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
  - nticks (integer greater than or equal to 0)
    default: 0

Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".

- tick0 (`number or categorical coordinate string`)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (`number or categorical coordinate string`)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"

- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - line
- width (number greater than or equal to 0)
  Sets the width (in px) of the lines bounding the marker points.
- color (color)
  Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu,

Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis

- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
- cmin (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
- autocolorscale (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (`boolean`)
  Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- widthsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for width . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- color (`color`)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (`colorscale`)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
- cmin (`number`)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (`boolean`)
  default: `True`
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (`boolean`)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- symbolsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for symbol . A `source string` refers to a unique identifier that is assigned

to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- opacitysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for opacity . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's opacity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "middle center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.
- textfont
  Sets the text font.
  - family (string)
  - size (number greater than or equal to 1)
  - color (color)
  - familysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - sizesrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- r (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  For polar chart only.Sets the radial coordinates.
- t (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  For polar chart only.Sets the angular coordinates.
- error_y
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.

- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_x
- visible (boolean)
  Determines whether or not this set of error bars is visible.
- type (enumerated: "percent" | "constant" | "sqrt" | "data")
  Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
- symmetric (boolean)
  Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
- array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
- arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.

- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- idssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ids . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ids data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textpositionsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for textposition . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- rsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for r . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's r data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- tsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for t . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's t data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

scatter

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- x0
- dx
- y
- y0
- dy
- ids
- text
- mode
- hoveron
- line
  - color
  - width
  - shape
  - smoothing
  - dash
  - simplify
- connectgaps
- fill
- fillcolor
- marker
  - symbol
  - opacity
  - size
  - maxdisplayed
  - sizeref
  - sizemin
  - sizemode
  - showscale
  - colorbar
    - thicknessmode
    - thickness
    - lenmode
    - len
    - x
    - xanchor
    - xpad
    - y
    - yanchor
    - ypad
    - outlinecolor
    - outlinewidth
    - bordercolor

- borderwidth
- bgcolor
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc
- line
  - width
  - color
- colorscale
- cauto
- cmax
- cmin
- autocolorscale
- reversescale
- widthsrc
- colorsrc
- color
- colorscale
- cauto
- cmax
- cmin
- autocolorscale
- reversescale
- symbolsrc
- opacitysrc
- sizesrc
- colorsrc
- textposition
- textfont
  - family
  - size
  - color
- familysrc
- sizesrc
- colorsrc

- r
- t
- error_y
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_x
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc
- idssrc
- textsrc
- textpositionsrc
- rsrc
- tsrc

盒形图

```
import plotly.graph_objs as go
go.Box
```

A `Box` trace is a graph object with any of the named arguments or attributes listed below.

- visible (enumerated: True | False | "legendonly")
  default: True
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (boolean)
  default: True
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (string)
  default: ""
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the trace.
- name (string)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (flaglist string)
  Any combination of "x", "y", "z", "text", "name" joined with a "+" OR "all" or "none" or "skip". examples: "x", "y", "x+y", "x+y+z", "all"
  default: "all"
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (string)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (number between or equal to 0 and 10000)
    default: 500
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y sample data or coordinates. See overview for more info.
- x (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the x sample data or coordinates. See overview for more info.
- x0 (number or categorical coordinate string)
  Sets the x coordinate of the box. See overview for more info.
- y0 (number or categorical coordinate string)
  Sets the y coordinate of the box. See overview for more info.
- whiskerwidth (number between or equal to 0 and 1)
  default: 0.5
  Sets the width of the whiskers relative to the box' width. For example, with 1, the whiskers are as wide as the box(es).
- boxpoints (enumerated: "all" | "outliers" | "suspectedoutliers" | False)
  default: "outliers"
  If "outliers", only the sample points lying outside the whiskers are shown If "suspectedoutliers", the outlier points are shown and points either less than 4"Q1-3"Q3 or greater than 4"Q3-3"Q1 are highlighted (see `outliercolor`) If "all", all sample points are shown If "False", only the box(es) are shown with no sample points
- boxmean (enumerated: True | "sd" | False)
  If "True", the mean of the box(es)' underlying distribution is drawn as a dashed line inside the box(es). If "sd" the standard deviation is also drawn.
- jitter (number between or equal to 0 and 1)
  Sets the amount of jitter in the sample points drawn. If "0", the sample points align along the distribution axis. If "1", the sample points are drawn in a random jitter of width equal to the width of the box(es).
- pointpos (number between or equal to -2 and 2)
  Sets the position of the sample points in relation to the box(es). If "0", the sample points are places over the center of the box(es). Positive (negative) values correspond to positions to the right (left) for vertical boxes and above (below) for horizontal boxes
- orientation (enumerated: "v" | "h")
  Sets the orientation of the box(es). If "v" ("h"), the distribution is visualized along the vertical (horizontal).
- marker
  - outliercolor (color)
    default: "rgba(0, 0, 0, 0)"
    Sets the color of the outlier sample points.

- symbol (enumerated: "0" | "circle" | "100" | "circle-open" | "200" | "circle-dot" | "300" | "circle-open-dot" | "1" | "square" | "101" | "square-open" | "201" | "square-dot" | "301" | "square-open-dot" | "2" | "diamond" | "102" | "diamond-open" | "202" | "diamond-dot" | "302" | "diamond-open-dot" | "3" | "cross" | "103" | "cross-open" | "203" | "cross-dot" | "303" | "cross-open-dot" | "4" | "x" | "104" | "x-open" | "204" | "x-dot" | "304" | "x-open-dot" | "5" | "triangle-up" | "105" | "triangle-up-open" | "205" | "triangle-up-dot" | "305" | "triangle-up-open-dot" | "6" | "triangle-down" | "106" | "triangle-down-open" | "206" | "triangle-down-dot" | "306" | "triangle-down-open-dot" | "7" | "triangle-left" | "107" | "triangle-left-open" | "207" | "triangle-left-dot" | "307" | "triangle-left-open-dot" | "8" | "triangle-right" | "108" | "triangle-right-open" | "208" | "triangle-right-dot" | "308" | "triangle-right-open-dot" | "9" | "triangle-ne" | "109" | "triangle-ne-open" | "209" | "triangle-ne-dot" | "309" | "triangle-ne-open-dot" | "10" | "triangle-se" | "110" | "triangle-se-open" | "210" | "triangle-se-dot" | "310" | "triangle-se-open-dot" | "11" | "triangle-sw" | "111" | "triangle-sw-open" | "211" | "triangle-sw-dot" | "311" | "triangle-sw-open-dot" | "12" | "triangle-nw" | "112" | "triangle-nw-open" | "212" | "triangle-nw-dot" | "312" | "triangle-nw-open-dot" | "13" | "pentagon" | "113" | "pentagon-open" | "213" | "pentagon-dot" | "313" | "pentagon-open-dot" | "14" | "hexagon" | "114" | "hexagon-open" | "214" | "hexagon-dot" | "314" | "hexagon-open-dot" | "15" | "hexagon2" | "115" | "hexagon2-open" | "215" | "hexagon2-dot" | "315" | "hexagon2-open-dot" | "16" | "octagon" | "116" | "octagon-open" | "216" | "octagon-dot" | "316" | "octagon-open-dot" | "17" | "star" | "117" | "star-open" | "217" | "star-dot" | "317" | "star-open-dot" | "18" | "hexagram" | "118" | "hexagram-open" | "218" | "hexagram-dot" | "318" | "hexagram-open-dot" | "19" | "star-triangle-up" | "119" | "star-triangle-up-open" | "219" | "star-triangle-up-dot" | "319" | "star-triangle-up-open-dot" | "20" | "star-triangle-down" | "120" | "star-triangle-down-open" | "220" | "star-triangle-down-dot" | "320" | "star-triangle-down-open-dot" | "21" | "star-square" | "121" | "star-square-open" | "221" | "star-square-dot" | "321" | "star-square-open-dot" | "22" | "star-diamond" | "122" | "star-diamond-open" | "222" | "star-diamond-dot" | "322" | "star-diamond-open-dot" | "23" | "diamond-tall" | "123" | "diamond-tall-open" | "223" | "diamond-tall-dot" | "323" | "diamond-tall-open-dot" | "24" | "diamond-wide" | "124" | "diamond-wide-open" | "224" | "diamond-wide-dot" | "324" | "diamond-wide-open-dot" | "25" | "hourglass" | "125" | "hourglass-open" | "26" | "bowtie" | "126" | "bowtie-open" | "27" | "circle-cross" | "127" | "circle-cross-open" | "28" | "circle-x" | "128" | "circle-x-open" | "29" | "square-cross" | "129" | "square-cross-open" | "30" | "square-x" | "130" | "square-x-open" | "31" | "diamond-cross" | "131" | "diamond-cross-open" | "32" | "diamond-x" | "132" | "diamond-x-open" | "33" | "cross-thin" | "133" | "cross-thin-open" | "34" | "x-thin" | "134" | "x-thin-open" | "35" | "asterisk" | "135" | "asterisk-open" | "36" | "hash" | "136" | "hash-open" | "236" | "hash-dot" | "336" | "hash-open-dot" | "37" | "y-up" | "137" | "y-up-open" | "38" | "y-down" | "138" | "y-down-open" | "39" | "y-left" | "139" | "y-left-open" | "40" | "y-right" | "140" | "y-right-open" | "41" | "line-ew" | "141" | "line-ew-open" | "42" | "line-ns" | "142" | "line-ns-open" | "43" | "line-ne" | "143" | "line-ne-open" | "44" | "line-nw" | "144" | "line-nw-open")
  default: "circle"
  Sets the marker symbol type. Adding 100 is equivalent to appending "-open" to a symbol name. Adding 200 is equivalent to appending "-dot" to a symbol name. Adding 300 is equivalent to appending "-open-dot" or "dot-open" to a symbol name.
- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the marker opacity.
- size (number greater than or equal to 0)
  default: 6
  Sets the marker size (in px).
- color (color)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- line
  - color (color)
    default: "#444"
    Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.

- width (number greater than or equal to 0)
  default: 0
  Sets the width (in px) of the lines bounding the marker points.
- outliercolor (color)
  Sets the border line color of the outlier sample points. Defaults to marker.color
- outlierwidth (number greater than or equal to 0)
  default: 1
  Sets the border line width (in px) of the outlier sample points.
- line
  - color (color)
    Sets the color of line bounding the box(es).
  - width (number greater than or equal to 0)
    default: 2
    Sets the width (in px) of line bounding the box(es).
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

box

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- y
- x
- x0
- y0
- whiskerwidth
- boxpoints
- boxmean
- jitter
- pointpos
- orientation
- marker
  - outliercolor

- symbol
- opacity
- size
- color
- line
  - color
  - width
  - outliercolor
  - outlierwidth
- line
  - color
  - width
- fillcolor
- xaxis
- yaxis
- ysrc
- xsrc

柱形图

```
import plotly.graph_objs as go
go.Bar
```

A `Bar` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
  examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.

- x (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the x coordinates.
- x0 (number or categorical coordinate string)
  default: 0
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (number)
  default: 1
  Sets the x coordinate step. See `x0` for more info.
- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- y0 (number or categorical coordinate string)
  default: 0
  Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.
- dy (number)
  default: 1
  Sets the y coordinate step. See `y0` for more info.
- text (string)
  default: ""
  Sets text elements associated with each (x,y) pair. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y) coordinates.
- textposition (enumerated: "inside" | "outside" | "auto" | "none")
  default: "none"
  Specifies the location of the `text`. "inside" positions `text` inside, next to the bar end (rotated and scaled if needed). "outside" positions `text` outside, next to the bar end (scaled if needed). "auto" positions `text` inside or outside so that `text` size is maximized.
- textfont
  Sets the font used for `text`.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
  - familysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - sizesrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- insidetextfont
  Sets the font used for `text` lying inside the bar.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".

- size (number greater than or equal to 1)
- color (color)
- familysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- outsidetextfont
  Sets the font used for `text` lying outside the bar.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- familysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- orientation (enumerated: "v" | "h")
  Sets the orientation of the bars. With "v" ("h"), the value of the each bar spans along the vertical (horizontal).
- base (number or categorical coordinate string)
  Sets where the bar base is drawn (in position axis units). In "stack" or "relative" barmode, traces that set "base" will be excluded and drawn in "overlay" mode instead.
- offset (number)
  Shifts the position where the bar is drawn (in position axis units). In "group" barmode, traces that set "offset" will be excluded and drawn in "overlay" mode instead.
- width (number greater than or equal to 0)
  Sets the bar width (in position axis units).
- marker
- line
- width (number greater than or equal to 0)
  default: 0
  Sets the width (in px) of the lines bounding the marker points.
- color (color)
  Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be

an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis

- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
- cmin (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
- autocolorscale (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (`boolean`)
  Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- widthsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for width . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- color (`color`)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (`colorscale`)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
- cmin (`number`)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (`boolean`)
  default: `True`
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.

- reversescale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.
  - bordercolor (color)
    default: "#444"
    Sets the axis line color.
  - borderwidth (number greater than or equal to 0)
    default: 0
    Sets the width (in px) or the border enclosing this color bar.
  - bgcolor (color)
    default: "rgba(0,0,0,0)"
    Sets the color of padded area.
  - tickmode (enumerated: "auto" | "linear" | "array")
    Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
  - nticks (integer greater than or equal to 0)
    default: 0

Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".

- tick0 (`number or categorical coordinate string`)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (`number or categorical coordinate string`)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"

- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- r (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  For polar chart only.Sets the radial coordinates.
- t (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  For polar chart only.Sets the angular coordinates.
- error_y

- visible (boolean)
  Determines whether or not this set of error bars is visible.
- type (enumerated: "percent" | "constant" | "sqrt" | "data")
  Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
- symmetric (boolean)
  Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
- array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
- arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_x
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.

- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's

useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textpositionsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for textposition . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- basesrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for base . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's base data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- offsetsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for offset . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's offset data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- widthsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for width . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- rsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for r . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's r data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- tsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for t . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's t data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

bar

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- x0
- dx
- y
- y0
- dy
- text
- textposition
- textfont
  - family
  - size
  - color
  - familysrc
  - sizesrc
  - colorsrc
- insidetextfont

- family
- size
- color
- familysrc
- sizesrc
- colorsrc
- outsidetextfont
  - family
  - size
  - color
  - familysrc
  - sizesrc
  - colorsrc
- orientation
- base
- offset
- width
- marker
  - line
    - width
    - color
    - colorscale
    - cauto
    - cmax
    - cmin
    - autocolorscale
    - reversescale
    - widthsrc
    - colorsrc
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - showscale
  - colorbar
    - thicknessmode
    - thickness
    - lenmode
    - len
    - x
    - xanchor
    - xpad
    - y
    - yanchor
    - ypad
    - outlinecolor
    - outlinewidth
    - bordercolor
    - borderwidth
    - bgcolor
    - tickmode
    - nticks
    - tick0
    - dtick
    - tickvals
    - ticktext
    - ticks
    - ticklen
    - tickwidth
    - tickcolor
    - showticklabels

- tickfont
- family
- size
- color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
- family
- size
- color
- titleside
- tickvalssrc
- ticktextsrc
- colorsrc
- r
- t
- error_y
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_x
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc

- textsrc
- textpositionsrc
- basesrc
- offsetsrc
- widthsrc
- rsrc
- tsrc

热点图

```
import plotly.graph_objs as go
go.Heatmap
```

A `Heatmap` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
  examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the z data.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- x0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (`number`)
  default: `1`
  Sets the x coordinate step. See `x0` for more info.

- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- y0 (number or categorical coordinate string)
  default: 0
  Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.
- dy (number)
  default: 1
  Sets the y coordinate step. See `y0` for more info.
- text (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text elements associated with each z value.
- transpose (boolean)
  Transposes the z data.
- xtype (enumerated: "array" | "scaled")
  If "array", the heatmap's x coordinates are given by "x" (the default behavior when `x` is provided). If "scaled", the heatmap's x coordinates are given by "x0" and "dx" (the default behavior when `x` is not provided).
- ytype (enumerated: "array" | "scaled")
  If "array", the heatmap's y coordinates are given by "y" (the default behavior when `y` is provided) If "scaled", the heatmap's y coordinates are given by "y0" and "dy" (the default behavior when `y` is not provided)
- zsmooth (enumerated: "fast" | "best" | False)
  Picks a smoothing algorithm use to smooth `z` data.
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the `z` data are filled in.
- xgap (number greater than or equal to 0)
  default: 0
  Sets the horizontal gap (in pixels) between bricks.
- ygap (number greater than or equal to 0)
  default: 0
  Sets the vertical gap (in pixels) between bricks.
- zauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (number)
  Sets the lower bound of color domain.
- zmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1

Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.

- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.

- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- zsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful

if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

heatmap

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- z
- x
- x0
- dx
- y
- y0
- dy
- text
- transpose
- xtype
- ytype
- zsmooth
- connectgaps
- xgap
- ygap
- zauto
- zmin
- zmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0

- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc

- xaxis
- yaxis
- xcalendar
- ycalendar
- zsrc
- xsrc
- ysrc
- textsrc

直方图

```
import plotly.graph_objs as go
go.Histogram
```

A `Histogram` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.

- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the trace.
- name (string)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (flaglist string)
  Any combination of "x", "y", "z", "text", "name" joined with a "+" OR "all" or "none" or "skip".
  examples: "x", "y", "x+y", "x+y+z", "all"
  default: "all"
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (string)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (number between or equal to 0 and 10000)
    default: 500
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the sample data to be binned on the x axis.
- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the sample data to be binned on the y axis.
- text (string)
  default: ""
  Sets text elements associated with each (x,y) pair. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y) coordinates.
- orientation (enumerated: "v" | "h")
  Sets the orientation of the bars. With "v" ("h"), the value of the each bar spans along the vertical (horizontal).
- histfunc (enumerated: "count" | "sum" | "avg" | "min" | "max")
  default: "count"
  Specifies the binning function used for this histogram trace. If "count", the histogram values are computed by counting the number of values lying inside each bin. If "sum", "avg", "min", "max", the histogram values are computed using the sum, the average, the minimum or the maximum of the values lying inside each bin respectively.
- histnorm (enumerated: "" | "percent" | "probability" | "density" | "probability density")
  default: ""
  Specifies the type of normalization used for this histogram trace. If "", the span of each bar corresponds to the number of occurrences (i.e. the number of data points lying inside the bins). If "percent" / "probability", the span of each bar corresponds to the percentage / fraction of occurrences with respect to the total number of sample points (here, the sum of all bin HEIGHTS equals 100% / 1). If "density", the span of each bar corresponds to the number of occurrences in a bin divided by the size of the bin interval (here, the sum of all bin AREAS equals the total number of sample points). If "probability density", the area of each bar corresponds to the probability that an event will fall into the corresponding bin (here, the sum of all bin AREAS equals 1).
- cumulative
  - enabled (boolean)
    If True, display the cumulative distribution by summing the binned values. Use the `direction` and `centralbin` attributes to tune the accumulation method. Note: in this mode, the "density" `histnorm` settings behave the same as their equivalents without "density": "" and "density" both rise to the number of data points, and "probability" and "probability density" both rise to the number of sample points.
  - direction (enumerated: "increasing" | "decreasing")
    default: "increasing"
    Only applies if cumulative is enabled. If "increasing" (default) we sum all prior bins, so the result increases from left to right. If "decreasing" we sum later bins so the result decreases from left to right.
  - currentbin (enumerated: "include" | "exclude" | "half")
    default: "include"
    Only applies if cumulative is enabled. Sets whether the current bin is included, excluded, or has half of its value included in the current cumulative value. "include" is the default for compatibility with various other tools, however it introduces a half-bin bias to the results. "exclude" makes the opposite half-bin bias, and "half" removes it.
- autobinx (boolean)
  Determines whether or not the x axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in xbins.

- nbinsx (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- xbins
  - start (number or categorical coordinate string)
    Sets the starting value for the x axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the x axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each x axis bin.
- autobiny (boolean)
  Determines whether or not the y axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in ybins.
- nbinsy (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- ybins
  - start (number or categorical coordinate string)
    Sets the starting value for the y axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the y axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each y axis bin.
- marker
  - line
    - width (number greater than or equal to 0)
      default: 0
      Sets the width (in px) of the lines bounding the marker points.
    - color (color)
      Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
    - colorscale (colorscale)
      Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
    - cauto (boolean)
      default: True
      Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
    - cmax (number)
      Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
    - cmin (number)
      Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
    - autocolorscale (boolean)
      default: True
      Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
    - reversescale (boolean)
      Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
    - widthsrc (source string | plotly.grid objs.Column)
      Sets the source reference on plot.ly for width . A source string refers to a unique identifier that is assigned to

a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- colorsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - color (`color`)
    Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - colorscale (`colorscale`)
    Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
  - cauto (`boolean`)
    default: `True`
    Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
  - cmax (`number`)
    Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
  - cmin (`number`)
    Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
  - autocolorscale (`boolean`)
    default: `True`
    Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
  - reversescale (`boolean`)
    Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
  - showscale (`boolean`)
    Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
  - colorbar
- thicknessmode (`enumerated: "fraction" | "pixels"`)
  default: `"pixels"`
  Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
- thickness (`number greater than or equal to 0`)
  default: `30`
  Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
- lenmode (`enumerated: "fraction" | "pixels"`)
  default: `"fraction"`
  Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
- len (`number greater than or equal to 0`)
  default: `1`
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (`number between or equal to -2 and 3`)
  default: `1.02`
  Sets the x position of the color bar (in plot fraction).
- xanchor (`enumerated: "left" | "center" | "right"`)
  default: `"left"`
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.

- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""

Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.

- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated
- exponentformat (`enumerated: "none" | "e" | "E" | "power" | "SI" | "B"`)
  default: `"B"`
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (`string`)
  default: `"Click to enter colorscale title"`
  Sets the title of the color bar.

- titlefont
  Sets this color bar's title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- error_y
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
  - value (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
  - valueminus (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
  - traceref (integer greater than or equal to 0)
    default: 0
  - tracerefminus (integer greater than or equal to 0)
    default: 0
  - copy_ystyle (boolean)
  - copy_zstyle (boolean)
  - color (color)
    Sets the stoke color of the error bars.

- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_x
- visible (boolean)
  Determines whether or not this set of error bars is visible.
- type (enumerated: "percent" | "constant" | "sqrt" | "data")
  Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
- symmetric (boolean)
  Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
- array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
- arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

histogram

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- y
- text
- orientation
- histfunc
- histnorm
- cumulative
  - enabled
  - direction
  - currentbin
- autobinx
- nbinsx
- xbins
  - start
  - end

- size
- autobiny
- nbinsy
- vbins
- start
- end
- size
- marker
- line
  - width
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - widthsrc
  - colorsrc
- color
- colorscale
- cauto
- cmax
- cmin
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix
  - showtickprefix
  - ticksuffix
  - showticksuffix

- separatethousands
- exponentformat
- showexponent
- title
- titlefont
- family
- size
- color
- titleside
- tickvalssrc
- ticktextsrc
- colorsrc
- error_y
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_x
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc
- textsrc

2d 直方图

```
import plotly.graph_objs as go
go.Histogram2D
```

A `Histogram2D` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the sample data to be binned on the x axis.
- y (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the sample data to be binned on the y axis.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the aggregation data.
- marker
  - color (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
    Sets the aggregation data.
  - colorsrc (`source string | plotly.grid objs.Column`)
    Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- histnorm (`enumerated: "" | "percent" | "probability" | "density" | "probability density"`)
  default: `""`
  Specifies the type of normalization used for this histogram trace. If "", the span of each bar corresponds to the number of occurrences (i.e. the number of data points lying inside the bins). If "percent" / "probability", the span of each bar corresponds to the percentage / fraction of occurrences with respect to the total number of sample points (here, the sum of all bin HEIGHTS equals 100% / 1). If "density", the span of each bar corresponds to the number of occurrences in a bin divided by the size of the bin interval (here, the sum of all bin AREAS equals the total number of sample points). If "probability density", the area of each bar corresponds to the probability that an event will fall into the corresponding bin (here, the sum of all bin AREAS equals 1).
- histfunc (`enumerated: "count" | "sum" | "avg" | "min" | "max"`)
  default: `"count"`
  Specifies the binning function used for this histogram trace. If "count", the histogram values are computed by counting the number of values lying inside each bin. If "sum", "avg", "min", "max", the histogram values are

computed using the sum, the average, the minimum or the maximum of the values lying inside each bin respectively.

- autobinx (boolean)
  Determines whether or not the x axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in xbins.
- nbinsx (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- xbins
  - start (number or categorical coordinate string)
    Sets the starting value for the x axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the x axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each x axis bin.
- autobiny (boolean)
  Determines whether or not the y axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in ybins.
- nbinsy (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- ybins
  - start (number or categorical coordinate string)
    Sets the starting value for the y axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the y axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each y axis bin.
- xgap (number greater than or equal to 0)
  default: 0
  Sets the horizontal gap (in pixels) between bricks.
- ygap (number greater than or equal to 0)
  default: 0
  Sets the vertical gap (in pixels) between bricks.
- zsmooth (enumerated: "fast" | "best" | False)
  Picks a smoothing algorithm use to smooth `z` data.
- zauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (number)
  Sets the lower bound of color domain.
- zmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.

- thickness (number greater than or equal to 0)
  default: 30
  Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
- lenmode (enumerated: "fraction" | "pixels")
  default: "fraction"
  Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
- len (number greater than or equal to 0)
  default: 1
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick

to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (`string`)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (`number greater than or equal to 1`)
  - color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.

- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful

if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- ysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for y . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for z . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

histogram2d

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- y
- z
- marker
  - color
  - colorsrc
- histnorm
- histfunc
- autobinx
- nbinsx
- xbins
  - start
  - end
  - size
- autobiny
- nbinsy
- vbins
  - start
  - end
  - size
- xgap
- ygap
- zsmooth
- zauto
- zmin
- zmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len

- x
- xanchor
- xpad
- y
- yanchor
- ypad
- outlinecolor
- outlinewidth
- bordercolor
- borderwidth
- bgcolor
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc

- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc
- zsrc

2d 等值线直方图

```
import plotly.graph_objs as go
go.Histogram2Dcontour
```

A `Histogram2Dcontour` trace is a graph object with any of the named arguments or attributes listed below.

- visible (enumerated: True | False | "legendonly")
  default: True
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (boolean)
  default: True
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (string)
  default: ""
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the trace.
- name (string)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (flaglist string)
  Any combination of "x", "y", "z", "text", "name" joined with a "+" OR "all" or "none" or "skip". examples: "x", "y", "x+y", "x+y+z", "all"
  default: "all"
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (string)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (number between or equal to 0 and 10000)
    default: 500
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the sample data to be binned on the x axis.
- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the sample data to be binned on the y axis.
- z (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the aggregation data.
- marker
  - color (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the aggregation data.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- histnorm (enumerated: "" | "percent" | "probability" | "density" | "probability density")
  default: ""
  Specifies the type of normalization used for this histogram trace. If "", the span of each bar corresponds to the number of occurrences (i.e. the number of data points lying inside the bins). If "percent" / "probability", the span of each bar corresponds to the percentage / fraction of occurrences with respect to the total number of sample points (here, the sum of all bin HEIGHTS equals 100% / 1). If "density", the span of each bar corresponds to the number of occurrences in a bin divided by the size of the bin interval (here, the sum of all bin AREAS equals the total number of sample points). If "probability density", the area of each bar corresponds to the probability that an event will fall into the corresponding bin (here, the sum of all bin AREAS equals 1).
- histfunc (enumerated: "count" | "sum" | "avg" | "min" | "max")
  default: "count"
  Specifies the binning function used for this histogram trace. If "count", the histogram values are computed by counting the number of values lying inside each bin. If "sum", "avg", "min", "max", the histogram values are computed using the sum, the average, the minimum or the maximum of the values lying inside each bin respectively.
- autobinx (boolean)
  Determines whether or not the x axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in xbins.

- nbinsx (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- xbins
  - start (number or categorical coordinate string)
    Sets the starting value for the x axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the x axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each x axis bin.
- autobiny (boolean)
  Determines whether or not the y axis bin attributes are picked by an algorithm. Note that this should be set to False if you want to manually set the number of bins using the attributes in ybins.
- nbinsy (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of desired bins. This value will be used in an algorithm that will decide the optimal bin size such that the histogram best visualizes the distribution of the data.
- ybins
  - start (number or categorical coordinate string)
    Sets the starting value for the y axis bins.
  - end (number or categorical coordinate string)
    Sets the end value for the y axis bins.
  - size (number or categorical coordinate string)
    Sets the step in-between value each y axis bin.
- autocontour (boolean)
  default: True
  Determines whether or not the contour level attributes are picked by an algorithm. If "True", the number of contour levels can be set in `ncontours`. If "False", set the contour level attributes in `contours`.
- ncontours (integer greater than or equal to 1)
  default: 15
  Sets the maximum number of contour levels. The actual number of contours will be chosen automatically to be less than or equal to the value of `ncontours`. Has an effect only if `autocontour` is "True" or if `contours.size` is missing.
- contours
  - start (number)
    Sets the starting contour level value. Must be less than `contours.end`
  - end (number)
    Sets the end contour level value. Must be more than `contours.start`
  - size (number greater than or equal to 0)
    Sets the step between each contour level. Must be positive.
  - coloring (enumerated: "fill" | "heatmap" | "lines" | "none")
    default: "fill"
    Determines the coloring method showing the contour values. If "fill", coloring is done evenly between each contour level If "heatmap", a heatmap gradient coloring is applied between each contour level. If "lines", coloring is done on the contour lines. If "none", no coloring is applied on this trace.
  - showlines (boolean)
    default: True
    Determines whether or not the contour lines are drawn. Has only an effect if `contours.coloring` is set to "fill".
- line
  - color (color)
    Sets the color of the contour level. Has no if `contours.coloring` is set to "lines".
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
  - smoothing (number between or equal to 0 and 1.3)
    default: 1
    Sets the amount of smoothing for the contour lines, where "0" corresponds to no smoothing.

- zauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (number)
  Sets the lower bound of color domain.
- zmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  default: True
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.

- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and

supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".

- size (number greater than or equal to 1)
- color (color)

- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.

- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)

- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

histogram2dcontour

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- y
- z
- marker
  - color
  - colorsrc
- histnorm
- histfunc
- autobinx
- nbinsx

- xbins
  - start
  - end
  - size
- autobiny
- nbinsy
- vbins
  - start
  - end
  - size
- autocontour
- ncontours
- contours
  - start
  - end
  - size
  - coloring
  - showlines
- line
  - color
  - width
  - dash
  - smoothing
- zauto
- zmin
- zmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix

- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc
- zsrc

饼图扇形图

```
import plotly.graph_objs as go
go.Pie
```

A `Pie` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"label"`, `"text"`, `"value"`, `"percent"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
  examples: `"label"`, `"text"`, `"label+text"`, `"label+text+value"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.

- maxpoints (number between or equal to 0 and 10000)
  default: 500
  Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- labels (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the sector labels.
- label0 (number)
  default: 0
  Alternate to `labels`. Builds a numeric set of labels. Use with `dlabel` where `label0` is the starting label and `dlabel` the step.
- dlabel (number)
  default: 1
  Sets the label step. See `label0` for more info.
- marker
  - colors (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the color of each sector of this pie chart. If not specified, the default trace color set is used to pick the sector colors.
  - line
    - color (color)
      default: "#444"
      Sets the color of the line enclosing each sector.
    - width (number greater than or equal to 0)
      default: 0
      Sets the width (in px) of the line enclosing each sector.
    - colorsrc (source string | plotly.grid objs.Column)
      Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
    - widthsrc (source string | plotly.grid objs.Column)
      Sets the source reference on plot.ly for width . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorssrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for colors . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's colors data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- text (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets text elements associated with each sector.
- scalegroup (string)
  default: ""
  If there are multiple pies that should be sized according to their totals, link them by providing a non-empty group id here shared by every trace in the same group.
- textinfo (flaglist string)
  Any combination of "label", "text", "value", "percent" joined with a "+" OR "none".
  examples: "label", "text", "label+text", "label+text+value", "none"
  Determines which trace information appear on the graph.
- textposition (enumerated: "inside" | "outside" | "auto" | "none")
  default: "auto"
  Specifies the location of the `textinfo`.
- textfont
  Sets the font used for `textinfo`.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)

- color (color)
- insidetextfont
  Sets the font used for `textinfo` lying inside the pie.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- outsidetextfont
  Sets the font used for `textinfo` lying outside the pie.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- domain
  - x (list)
    default: [0, 1]
    Sets the horizontal domain of this pie trace (in plot fraction).
    Each dict has one or more of the keys listed below.
  - y (list)
    default: [0, 1]
    Sets the vertical domain of this pie trace (in plot fraction).
    Each dict has one or more of the keys listed below.
- hole (number between or equal to 0 and 1)
  default: 0
  Sets the fraction of the radius to cut out of the pie. Use this to make a donut chart.
- sort (boolean)
  default: True
  Determines whether or not the sectors of reordered from largest to smallest.
- direction (enumerated: "clockwise" | "counterclockwise")
  default: "counterclockwise"
  Specifies the direction at which succeeding sectors follow one another.
- rotation (number between or equal to -360 and 360)
  default: 0
  Instead of the first slice starting at 12 o'clock, rotate to some other angle.
- pull (number between or equal to 0 and 1)
  default: 0
  Sets the fraction of larger radius to pull the sectors out from the center. This can be a constant to pull all slices apart from each other equally or an array to highlight one or more slices.
- labelssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for labels . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's labels data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- valuessrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for values . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's values data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's

useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textpositionsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for textposition . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- pullsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for pull . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's pull data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

pie

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- labels
- label0
- dlabel
- marker
  - colors
  - line
    - color
    - width
    - colorsrc
    - widthsrc
  - colorssrc
- text
- scalegroup
- textinfo
- textposition
- textfont
  - family
  - size
  - color
- insidetextfont
  - family
  - size
  - color
- outsidetextfont
  - family
  - size
  - color
- domain
  - x
    Each dict has one or more of the keys listed below.
  - y
    Each dict has one or more of the keys listed below.
- hole
- sort
- direction
- rotation

- pull
- labelssrc
- valuessrc
- textsrc
- textpositionsrc
- pullsrc

轮廓图

```
import plotly.graph_objs as go
go.Contour
```

A `Contour` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the z data.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- x0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (`number`)
  default: `1`
  Sets the x coordinate step. See `x0` for more info.

- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- y0 (number or categorical coordinate string)
  default: 0
  Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.
- dy (number)
  default: 1
  Sets the y coordinate step. See `y0` for more info.
- text (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text elements associated with each z value.
- transpose (boolean)
  Transposes the z data.
- xtype (enumerated: "array" | "scaled")
  If "array", the heatmap's x coordinates are given by "x" (the default behavior when `x` is provided). If "scaled", the heatmap's x coordinates are given by "x0" and "dx" (the default behavior when `x` is not provided).
- ytype (enumerated: "array" | "scaled")
  If "array", the heatmap's y coordinates are given by "y" (the default behavior when `y` is provided) If "scaled", the heatmap's y coordinates are given by "y0" and "dy" (the default behavior when `y` is not provided)
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the `z` data are filled in.
- autocontour (boolean)
  default: True
  Determines whether or not the contour level attributes are picked by an algorithm. If "True", the number of contour levels can be set in `ncontours`. If "False", set the contour level attributes in `contours`.
- ncontours (integer greater than or equal to 1)
  default: 15
  Sets the maximum number of contour levels. The actual number of contours will be chosen automatically to be less than or equal to the value of `ncontours`. Has an effect only if `autocontour` is "True" or if `contours.size` is missing.
- contours
  - start (number)
    Sets the starting contour level value. Must be less than `contours.end`
  - end (number)
    Sets the end contour level value. Must be more than `contours.start`
  - size (number greater than or equal to 0)
    Sets the step between each contour level. Must be positive.
  - coloring (enumerated: "fill" | "heatmap" | "lines" | "none")
    default: "fill"
    Determines the coloring method showing the contour values. If "fill", coloring is done evenly between each contour level If "heatmap", a heatmap gradient coloring is applied between each contour level. If "lines", coloring is done on the contour lines. If "none", no coloring is applied on this trace.
  - showlines (boolean)
    default: True
    Determines whether or not the contour lines are drawn. Has only an effect if `contours.coloring` is set to "fill".
- line
  - color (color)
    Sets the color of the contour level. Has no if `contours.coloring` is set to "lines".
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
  - smoothing (number between or equal to 0 and 1.3)
    default: 1
    Sets the amount of smoothing for the contour lines, where "0" corresponds to no smoothing.
- zauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (number)
  Sets the lower bound of color domain.

- zmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.
  - bordercolor (color)
    default: "#444"
    Sets the axis line color.
  - borderwidth (number greater than or equal to 0)
    default: 0
    Sets the width (in px) or the border enclosing this color bar.

- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)

- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- zsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

contour

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- z
- x
- x0
- dx
- y
- y0

- dy
- text
- transpose
- xtype
- ytype
- connectgaps
- autocontour
- ncontours
- contours
  - start
  - end
  - size
  - coloring
  - showlines
- line
  - color
  - width
  - dash
  - smoothing
- zauto
- zmin
- zmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix
  - showtickprefix
  - ticksuffix
  - showticksuffix
  - separatethousands

- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- zsrc
- xsrc
- ysrc
- textsrc

散射图

```
import plotly.graph_objs as go
go.Scatterternary
```

A `Scatterternary` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"a"`, `"b"`, `"c"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"a"`, `"b"`, `"a+b"`, `"a+b+c"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.

- a (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the quantity of component `a` in each data point. If `a`, `b`, and `c` are all provided, they need not be normalized, only the relative values matter. If only two arrays are provided they must be normalized to match `ternary<i>.sum`.
- b (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the quantity of component `a` in each data point. If `a`, `b`, and `c` are all provided, they need not be normalized, only the relative values matter. If only two arrays are provided they must be normalized to match `ternary<i>.sum`.
- c (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the quantity of component `a` in each data point. If `a`, `b`, and `c` are all provided, they need not be normalized, only the relative values matter. If only two arrays are provided they must be normalized to match `ternary<i>.sum`.
- sum (number greater than or equal to 0)
  default: 0
  The number each triplet should sum to, if only two of `a`, `b`, and `c` are provided. This overrides `ternary<i>.sum` to normalize this specific trace, but does not affect the values displayed on the axes. 0 (or missing) means to use ternary<i>.sum
- mode (flaglist string)
  Any combination of "lines", "markers", "text" joined with a "+" OR "none".
  examples: "lines", "markers", "lines+markers", "lines+markers+text", "none"
  default: "markers"
  Determines the drawing mode for this scatter trace. If the provided `mode` includes "text" then the `text` elements appear at the coordinates. Otherwise, the `text` elements appear on hover. If there are less than 20 points, then the default is "lines+markers". Otherwise, "lines".
- text (string)
  default: ""
  Sets text elements associated with each (a,b,c) point. If a single string, the same string appears over all the data points. If an array of strings, the items are mapped in order to the the data points in (a,b,c).
- line
  - color (color)
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
  - shape (enumerated: "linear" | "spline")
    default: "linear"
    Determines the line shape. With "spline" the lines are drawn using spline interpolation. The other available values correspond to step-wise line shapes.
  - smoothing (number between or equal to 0 and 1.3)
    default: 1
    Has an effect only if `shape` is set to "spline" Sets the amount of smoothing. "0" corresponds to no smoothing (equivalent to a "linear" shape).
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- fill (enumerated: "none" | "toself" | "tonext")
  default: "none"
  Sets the area to fill with a solid color. Use with `fillcolor` if not "none". scatterternary has a subset of the options available to scatter. "toself" connects the endpoints of the trace (or each segment of the trace if it has gaps) into a closed shape. "tonext" fills the space between two traces if one completely encloses the other (eg consecutive contour lines), and behaves like "toself" if there is no trace before it. "tonext" should not be used if one trace does not enclose the other.
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- marker
  - symbol (enumerated: "0" | "circle" | "100" | "circle-open" | "200" | "circle-dot" | "300" | "circle-open-dot" | "1" | "square" | "101" | "square-open" | "201" | "square-dot" | "301" | "square-open-dot" | "2" | "diamond" | "102" | "diamond-open" | "202" | "diamond-dot" | "302" | "diamond-open-dot" | "3" | "cross" | "103" | "cross-open" | "203" | "cross-dot" | "303" | "cross-open-

dot" | "4" | "x" | "104" | "x-open" | "204" | "x-dot" | "304" | "x-open-dot" | "5" | "triangle-up" | "105" | "triangle-up-open" | "205" | "triangle-up-dot" | "305" | "triangle-up-open-dot" | "6" | "triangle-down" | "106" | "triangle-down-open" | "206" | "triangle-down-dot" | "306" | "triangle-down-open-dot" | "7" | "triangle-left" | "107" | "triangle-left-open" | "207" | "triangle-left-dot" | "307" | "triangle-left-open-dot" | "8" | "triangle-right" | "108" | "triangle-right-open" | "208" | "triangle-right-dot" | "308" | "triangle-right-open-dot" | "9" | "triangle-ne" | "109" | "triangle-ne-open" | "209" | "triangle-ne-dot" | "309" | "triangle-ne-open-dot" | "10" | "triangle-se" | "110" | "triangle-se-open" | "210" | "triangle-se-dot" | "310" | "triangle-se-open-dot" | "11" | "triangle-sw" | "111" | "triangle-sw-open" | "211" | "triangle-sw-dot" | "311" | "triangle-sw-open-dot" | "12" | "triangle-nw" | "112" | "triangle-nw-open" | "212" | "triangle-nw-dot" | "312" | "triangle-nw-open-dot" | "13" | "pentagon" | "113" | "pentagon-open" | "213" | "pentagon-dot" | "313" | "pentagon-open-dot" | "14" | "hexagon" | "114" | "hexagon-open" | "214" | "hexagon-dot" | "314" | "hexagon-open-dot" | "15" | "hexagon2" | "115" | "hexagon2-open" | "215" | "hexagon2-dot" | "315" | "hexagon2-open-dot" | "16" | "octagon" | "116" | "octagon-open" | "216" | "octagon-dot" | "316" | "octagon-open-dot" | "17" | "star" | "117" | "star-open" | "217" | "star-dot" | "317" | "star-open-dot" | "18" | "hexagram" | "118" | "hexagram-open" | "218" | "hexagram-dot" | "318" | "hexagram-open-dot" | "19" | "star-triangle-up" | "119" | "star-triangle-up-open" | "219" | "star-triangle-up-dot" | "319" | "star-triangle-up-open-dot" | "20" | "star-triangle-down" | "120" | "star-triangle-down-open" | "220" | "star-triangle-down-dot" | "320" | "star-triangle-down-open-dot" | "21" | "star-square" | "121" | "star-square-open" | "221" | "star-square-dot" | "321" | "star-square-open-dot" | "22" | "star-diamond" | "122" | "star-diamond-open" | "222" | "star-diamond-dot" | "322" | "star-diamond-open-dot" | "23" | "diamond-tall" | "123" | "diamond-tall-open" | "223" | "diamond-tall-dot" | "323" | "diamond-tall-open-dot" | "24" | "diamond-wide" | "124" | "diamond-wide-open" | "224" | "diamond-wide-dot" | "324" | "diamond-wide-open-dot" | "25" | "hourglass" | "125" | "hourglass-open" | "26" | "bowtie" | "126" | "bowtie-open" | "27" | "circle-cross" | "127" | "circle-cross-open" | "28" | "circle-x" | "128" | "circle-x-open" | "29" | "square-cross" | "129" | "square-cross-open" | "30" | "square-x" | "130" | "square-x-open" | "31" | "diamond-cross" | "131" | "diamond-cross-open" | "32" | "diamond-x" | "132" | "diamond-x-open" | "33" | "cross-thin" | "133" | "cross-thin-open" | "34" | "x-thin" | "134" | "x-thin-open" | "35" | "asterisk" | "135" | "asterisk-open" | "36" | "hash" | "136" | "hash-open" | "236" | "hash-dot" | "336" | "hash-open-dot" | "37" | "y-up" | "137" | "y-up-open" | "38" | "y-down" | "138" | "y-down-open" | "39" | "y-left" | "139" | "y-left-open" | "40" | "y-right" | "140" | "y-right-open" | "41" | "line-ew" | "141" | "line-ew-open" | "42" | "line-ns" | "142" | "line-ns-open" | "43" | "line-ne" | "143" | "line-ne-open" | "44" | "line-nw" | "144" | "line-nw-open")
default: "circle"
Sets the marker symbol type. Adding 100 is equivalent to appending "-open" to a symbol name. Adding 200 is equivalent to appending "-dot" to a symbol name. Adding 300 is equivalent to appending "-open-dot" or "dot-open" to a symbol name.

- opacity (number between or equal to 0 and 1)
  Sets the marker opacity.
- maxdisplayed (number greater than or equal to 0)
  default: 0
  Sets a maximum number of points to be drawn on the graph. "0" corresponds to no limit.
- size (number greater than or equal to 0)
  default: 6
  Sets the marker size (in px).
- sizeref (number)
  default: 1
  Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
- sizemin (number greater than or equal to 0)
  default: 0
  Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.
- sizemode (enumerated: "diameter" | "area")
  default: "diameter"

Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.

- line
  - width (`number greater than or equal to 0`)
    Sets the width (in px) of the lines bounding the marker points.
  - color (`color`)
    Sets the color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - colorscale (`colorscale`)
    Sets the colorscale and only has an effect if `.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `.cmin` and `.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
  - cauto (`boolean`)
    default: `True`
    Has an effect only if `.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
  - cmax (`number`)
    Has an effect only if `.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `.color` array index, and if set, `.cmin` must be set as well.
  - cmin (`number`)
    Has an effect only if `.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `.color` array index, and if set, `.cmax` must be set as well.
  - autocolorscale (`boolean`)
    default: `True`
    Has an effect only if `.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
  - reversescale (`boolean`)
    Has an effect only if `.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
  - widthsrc (`source string | plotly.grid objs.Column`)
    Sets the source reference on plot.ly for width . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (`source string | plotly.grid objs.Column`)
    Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- color (`color`)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (`colorscale`)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.

- cmin (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.
  - bordercolor (color)
    default: "#444"
    Sets the axis line color.
  - borderwidth (number greater than or equal to 0)
    default: 0
    Sets the width (in px) or the border enclosing this color bar.

- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)

- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- symbolsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for symbol . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- opacitysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for opacity . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's opacity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textfont
  Sets the text font.
  - family (string)
  - size (number greater than or equal to 1)
  - color (color)
  - familysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - sizesrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - colorsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "middle center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.
- hoveron (flaglist string)
  Any combination of "points", "fills" joined with a "+"
  examples: "points", "fills", "points+fills"
  Do the hover effects highlight individual points (markers or line points) or do they highlight filled regions? If the fill is "toself" or "tonext" and there are no markers or text, then the default is "fills", otherwise it is "points".
- subplot (subplotid)
  default: ternary
  Sets a reference between this trace's data coordinates and a ternary subplot. If "ternary" (the default value), the data refer to `layout.ternary`. If "ternary2", the data refer to `layout.ternary2`, and so on.
- asrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for a . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's a data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- bsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for b . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's b data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- csrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for c . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's c data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textpositionsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for textposition . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- a
- b
- c
- sum
- mode
- text
- line
  - color
  - width
  - dash
  - shape
  - smoothing
- connectgaps
- fill
- fillcolor
- marker
  - symbol
  - opacity
  - maxdisplayed
  - size
  - sizeref
  - sizemin
  - sizemode
  - line
    - width
    - color
    - colorscale
    - cauto
    - cmax
    - cmin
    - autocolorscale
    - reversescale
    - widthsrc
    - colorsrc
  - color
  - colorscale
  - cauto

- cmax
- cmin
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix
  - showtickprefix
  - ticksuffix
  - showticksuffix
  - separatethousands
  - exponentformat
  - showexponent
  - title
  - titlefont
    - family
    - size
    - color
  - titleside
  - tickvalssrc
  - ticktextsrc
- symbolsrc
- opacitysrc
- sizesrc
- colorsrc
- textfont
  - family
  - size
  - color
- familysrc
- sizesrc
- colorsrc

- textposition
- hoveron
- subplot
- asrc
- bsrc
- csrc
- textsrc
- textpositionsrc

3d 散点图

```
import plotly.graph_objs as go
go.Scatter3D
```

A `Scatter3D` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- y (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the y coordinates.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the z coordinates.
- text (`string`)
  default: `""`
  Sets text elements associated with each (x,y,z) triplet. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y,z) coordinates.

- mode (flaglist string)
  Any combination of "lines", "markers", "text" joined with a "+" OR "none".
  examples: "lines", "markers", "lines+markers", "lines+markers+text", "none"
  default: "lines+markers"
  Determines the drawing mode for this scatter trace. If the provided `mode` includes "text" then the `text` elements appear at the coordinates. Otherwise, the `text` elements appear on hover. If there are less than 20 points, then the default is "lines+markers". Otherwise, "lines".
- surfaceaxis (enumerated: "-1" | "0" | "1" | "2")
  default: "-1"
  If "-1", the scatter points are not fill with a surface If "0", "1", "2", the scatter points are filled with a Delaunay surface about the x, y, z respectively.
- surfacecolor (color)
  Sets the surface fill color.
- projection
  - x
    - show (boolean)
      Sets whether or not projections are shown along the x axis.
    - opacity (number between or equal to 0 and 1)
      default: 1
      Sets the projection color.
    - scale (number between or equal to 0 and 10)
      default: 0.6666666666666666
      Sets the scale factor determining the size of the projection marker points.
  - y
    - show (boolean)
      Sets whether or not projections are shown along the y axis.
    - opacity (number between or equal to 0 and 1)
      default: 1
      Sets the projection color.
    - scale (number between or equal to 0 and 10)
      default: 0.6666666666666666
      Sets the scale factor determining the size of the projection marker points.
  - z
    - show (boolean)
      Sets whether or not projections are shown along the z axis.
    - opacity (number between or equal to 0 and 1)
      default: 1
      Sets the projection color.
    - scale (number between or equal to 0 and 10)
      default: 0.6666666666666666
      Sets the scale factor determining the size of the projection marker points.
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- line
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
  - showscale (boolean)
    Has an effect only if `line.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
  - color (color)
    Sets the line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - colorscale (colorscale)
    Sets the colorscale and only has an effect if `line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `line.cmin` and `line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis

- cauto (boolean)
  default: True
  Has an effect only if `line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (number)
  Has an effect only if `line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `line.color` array index, and if set, `line.cmin` must be set as well.
- cmin (number)
  Has an effect only if `line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `line.color` array index, and if set, `line.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- marker
- symbol (enumerated: "circle" | "circle-open" | "square" | "square-open" | "diamond" | "diamond-open" | "cross" | "x")
  default: "circle"
  Sets the marker symbol type.
- size (number greater than or equal to 0)
  default: 8
  Sets the marker size (in px).
- sizeref (number)
  default: 1
  Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
- sizemin (number greater than or equal to 0)
  default: 0
  Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.
- sizemode (enumerated: "diameter" | "area")
  default: "diameter"
  Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.
- opacity (number between or equal to 0 and 1)
  Sets the marker opacity. Note that the marker opacity for scatter3d traces must be a scalar value for performance reasons. To set a blending opacity value (i.e. which is not transparent), set "marker.color" to an rgba color and use its alpha channel.
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar

  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.

- len (number greater than or equal to 0)
  default: 1
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n`

must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated
- exponentformat (`enumerated: "none" | "e" | "E" | "power" | "SI" | "B"`)
  default: `"B"`

Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.

- showexponent (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (`string`)
  default: `"Click to enter colorscale title"`
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- titleside (`enumerated: "right" | "top" | "bottom"`)
  default: `"top"`
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for tickvals . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for ticktext . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - line
- width (`number greater than or equal to 0`)
  Sets the width (in px) of the lines bounding the marker points.
- color (`color`)
  Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (`colorscale`)
  Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
- cmin (`number`)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
- autocolorscale (`boolean`)
  default: `True`
  Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default

palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.

- reversescale (boolean)
  Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- color (color)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
- cmin (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- symbolsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for symbol . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "top center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.
- textfont
  Sets the text font.
  - family (string)
  - size (number greater than or equal to 1)

- color (color)
- familysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_x
- visible (boolean)
  Determines whether or not this set of error bars is visible.
- type (enumerated: "percent" | "constant" | "sqrt" | "data")
  Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
- symmetric (boolean)
  Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
- array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
- arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_y
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
  - value (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
  - valueminus (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
  - traceref (integer greater than or equal to 0)
    default: 0
  - tracerefminus (integer greater than or equal to 0)
    default: 0
  - copy_ystyle (boolean)
  - copy_zstyle (boolean)
  - color (color)
    Sets the stoke color of the error bars.
  - thickness (number greater than or equal to 0)
    default: 2
    Sets the thickness (in px) of the error bars.
  - width (number greater than or equal to 0)
    Sets the width (in px) of the cross-bar at both ends of the error bars.
  - arraysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - arrayminussrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- error_z
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)

Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.

- value (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
- valueminus (number greater than or equal to 0)
  default: 10
  Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
- traceref (integer greater than or equal to 0)
  default: 0
- tracerefminus (integer greater than or equal to 0)
  default: 0
- copy_ystyle (boolean)
- copy_zstyle (boolean)
- color (color)
  Sets the stoke color of the error bars.
- thickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the error bars.
- width (number greater than or equal to 0)
  Sets the width (in px) of the cross-bar at both ends of the error bars.
- arraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- arrayminussrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- scene (subplotid)
  default: scene
  Sets a reference between this trace's 3D coordinate system and a 3D scene. If "scene" (the default value), the (x,y,z) coordinates refer to `layout.scene`. If "scene2", the (x,y,z) coordinates refer to `layout.scene2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- zcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `z` date data.
- xsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- zsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for z . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textpositionsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for textposition . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

scatter3d

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- y
- z
- text
- mode
- surfaceaxis
- surfacecolor
- projection
  - x
    - show
    - opacity
    - scale
  - y
    - show
    - opacity
    - scale
  - z
    - show
    - opacity
    - scale
- connectgaps
- line
  - width
  - dash
  - showscale
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - colorsrc

- marker
  - symbol
  - size
  - sizeref
  - sizemin
  - sizemode
  - opacity
  - showscale
  - colorbar
    - thicknessmode
    - thickness
    - lenmode
    - len
    - x
    - xanchor
    - xpad
    - y
    - yanchor
    - ypad
    - outlinecolor
    - outlinewidth
    - bordercolor
    - borderwidth
    - bgcolor
    - tickmode
    - nticks
    - tick0
    - dtick
    - tickvals
    - ticktext
    - ticks
    - ticklen
    - tickwidth
    - tickcolor
    - showticklabels
    - tickfont
      - family
      - size
      - color
    - tickangle
    - tickformat
    - tickprefix
    - showtickprefix
    - ticksuffix
    - showticksuffix
    - separatethousands
    - exponentformat
    - showexponent
    - title
    - titlefont
      - family
      - size
      - color
    - titleside
    - tickvalssrc
    - ticktextsrc
  - line
    - width
    - color
    - colorscale
    - cauto
    - cmax
    - cmin
    - autocolorscale

- reversescale
- colorsrc
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - symbolsrc
  - sizesrc
  - colorsrc
- textposition
- textfont
  - family
  - size
  - color
  - familysrc
  - sizesrc
  - colorsrc
- error_x
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_y
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_z
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle

- color
- thickness
- width
- arraysrc
- arrayminussrc
- scene
- xcalendar
- ycalendar
- zcalendar
- xsrc
- ysrc
- zsrc
- textsrc
- textpositionsrc

表面图

```
import plotly.graph_objs as go
go.Surface
```

A `Surface` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the surface.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the z coordinates.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.

- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- text (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text elements associated with each z value.
- surfacecolor (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the surface color values, used for setting a color scale independent of `z`.
- cauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- cmin (number)
  Sets the lower bound of color domain.
- cmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.

- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.

- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.

- tickvalssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for tickvals . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for ticktext . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- contours
  - x
    - show (`boolean`)
      Determines whether or not contour lines about the x dimension are drawn.
    - project
    - x (`boolean`)
      Determines whether or not these contour lines are projected on the x plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - y (`boolean`)
      Determines whether or not these contour lines are projected on the y plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - z (`boolean`)
      Determines whether or not these contour lines are projected on the z plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - color (`color`)
      default: `"#444"`
      Sets the color of the contour lines.
    - usecolormap (`boolean`)
      An alternate to "color". Determines whether or not the contour lines are colored using the trace "colorscale".
    - width (`number between or equal to 1 and 16`)
      default: `2`
      Sets the width of the contour lines.
    - highlight (`boolean`)
      default: `True`
      Determines whether or not contour lines about the x dimension are highlighted on hover.
    - highlightcolor (`color`)
      default: `"#444"`
      Sets the color of the highlighted contour lines.
    - highlightwidth (`number between or equal to 1 and 16`)
      default: `2`
      Sets the width of the highlighted contour lines.
  - y
    - show (`boolean`)
      Determines whether or not contour lines about the y dimension are drawn.
    - project
    - x (`boolean`)
      Determines whether or not these contour lines are projected on the x plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - y (`boolean`)
      Determines whether or not these contour lines are projected on the y plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - z (`boolean`)
      Determines whether or not these contour lines are projected on the z plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
    - color (`color`)
      default: `"#444"`
      Sets the color of the contour lines.

- usecolormap (boolean)
  An alternate to "color". Determines whether or not the contour lines are colored using the trace "colorscale".
- width (number between or equal to 1 and 16)
  default: 2
  Sets the width of the contour lines.
- highlight (boolean)
  default: True
  Determines whether or not contour lines about the y dimension are highlighted on hover.
- highlightcolor (color)
  default: "#444"
  Sets the color of the highlighted contour lines.
- highlightwidth (number between or equal to 1 and 16)
  default: 2
  Sets the width of the highlighted contour lines.
  - z
- show (boolean)
  Determines whether or not contour lines about the z dimension are drawn.
- project
- x (boolean)
  Determines whether or not these contour lines are projected on the x plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
- y (boolean)
  Determines whether or not these contour lines are projected on the y plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
- z (boolean)
  Determines whether or not these contour lines are projected on the z plane. If `highlight` is set to "True" (the default), the projected lines are shown on hover. If `show` is set to "True", the projected lines are shown in permanence.
- color (color)
  default: "#444"
  Sets the color of the contour lines.
- usecolormap (boolean)
  An alternate to "color". Determines whether or not the contour lines are colored using the trace "colorscale".
- width (number between or equal to 1 and 16)
  default: 2
  Sets the width of the contour lines.
- highlight (boolean)
  default: True
  Determines whether or not contour lines about the z dimension are highlighted on hover.
- highlightcolor (color)
  default: "#444"
  Sets the color of the highlighted contour lines.
- highlightwidth (number between or equal to 1 and 16)
  default: 2
  Sets the width of the highlighted contour lines.

- hidesurface (boolean)
  Determines whether or not a surface is drawn. For example, set `hidesurface` to "False" `contours.x.show` to "True" and `contours.y.show` to "True" to draw a wire frame plot.
- lightposition
  - x (number between or equal to -100000 and 100000)
    default: 10
    Numeric vector, representing the X coordinate for each vertex.
  - y (number between or equal to -100000 and 100000)
    default: 10000
    Numeric vector, representing the Y coordinate for each vertex.
  - z (number between or equal to -100000 and 100000)
    default: 0
    Numeric vector, representing the Z coordinate for each vertex.
- lighting

- ambient (number between or equal to 0 and 1)
  default: 0.8
  Ambient light increases overall color visibility but can wash out the image.
- diffuse (number between or equal to 0 and 1)
  default: 0.8
  Represents the extent that incident rays are reflected in a range of angles.
- specular (number between or equal to 0 and 2)
  default: 0.05
  Represents the level that incident rays are reflected in a single direction, causing shine.
- roughness (number between or equal to 0 and 1)
  default: 0.5
  Alters specular reflection; the rougher the surface, the wider and less contrasty the shine.
- fresnel (number between or equal to 0 and 5)
  default: 0.2
  Represents the reflectance as a dependency of the viewing angle; e.g. paper is reflective when viewing it from the edge of the paper (almost 90 degrees), causing shine.

- scene (subplotid)
  default: scene
  Sets a reference between this trace's 3D coordinate system and a 3D scene. If "scene" (the default value), the (x,y,z) coordinates refer to `layout.scene`. If "scene2", the (x,y,z) coordinates refer to `layout.scene2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- zcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `z` date data.
- zsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- surfacecolorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for surfacecolor . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's surfacecolor data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

surface

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- z
- x
- y
- text
- surfacecolor
- cauto
- cmin
- cmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix
  - showtickprefix
  - ticksuffix
  - showticksuffix
  - separatethousands
  - exponentformat
  - showexponent

- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc
- contours
  - x
    - show
    - project
    - x
    - y
    - z
    - color
    - usecolormap
    - width
    - highlight
    - highlightcolor
    - highlightwidth
  - y
    - show
    - project
    - x
    - y
    - z
    - color
    - usecolormap
    - width
    - highlight
    - highlightcolor
    - highlightwidth
  - z
    - show
    - project
    - x
    - y
    - z
    - color
    - usecolormap
    - width
    - highlight
    - highlightcolor
    - highlightwidth
- hidesurface
- lightposition
  - x
  - y
  - z
- lighting
  - ambient
  - diffuse
  - specular
  - roughness
  - fresnel
- scene
- xcalendar
- ycalendar
- zcalendar
- zsrc
- xsrc
- ysrc

- textsrc
- surfacecolorsrc

3d 网格

`import plotly.graph_objs as go`
`go.Mesh3D`
A `Mesh3D` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the surface.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the X coordinates of the vertices. The nth element of vectors `x`, `y` and `z` jointly represent the X, Y and Z coordinates of the nth vertex.
- y (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the Y coordinates of the vertices. The nth element of vectors `x`, `y` and `z` jointly represent the X, Y and Z coordinates of the nth vertex.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the Z coordinates of the vertices. The nth element of vectors `x`, `y` and `z` jointly represent the X, Y and Z coordinates of the nth vertex.
- i (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  A vector of vertex indices, i.e. integer values between 0 and the length of the vertex vectors, representing the "first" vertex of a triangle. For example, `{i[m], j[m], k[m]}` together represent face m (triangle m) in the mesh, where `i[m] = n` points to the triplet `{x[n], y[n], z[n]}` in the vertex arrays. Therefore, each element in `i` represents a point in space, which is the first vertex of a triangle.
- j (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  A vector of vertex indices, i.e. integer values between 0 and the length of the vertex vectors, representing the

"second" vertex of a triangle. For example, `{i[m], j[m], k[m]}` together represent face m (triangle m) in the mesh, where `j[m] = n` points to the triplet `{x[n], y[n], z[n]}` in the vertex arrays. Therefore, each element in `j` represents a point in space, which is the second vertex of a triangle.

- k (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  A vector of vertex indices, i.e. integer values between 0 and the length of the vertex vectors, representing the "third" vertex of a triangle. For example, `{i[m], j[m], k[m]}` together represent face m (triangle m) in the mesh, where `k[m] = n` points to the triplet `{x[n], y[n], z[n]}` in the vertex arrays. Therefore, each element in `k` represents a point in space, which is the third vertex of a triangle.
- delaunayaxis (`enumerated: "x" | "y" | "z"`)
  default: `"z"`
  Sets the Delaunay axis, which is the axis that is perpendicular to the surface of the Delaunay triangulation. It has an effect if `i`, `j`, `k` are not provided and `alphahull` is set to indicate Delaunay triangulation.
- alphahull (`number`)
  default: `-1`
  Determines how the mesh surface triangles are derived from the set of vertices (points) represented by the `x`, `y` and `z` arrays, if the `i`, `j`, `k` arrays are not supplied. For general use of `mesh3d` it is preferred that `i`, `j`, `k` are supplied. If "-1", Delaunay triangulation is used, which is mainly suitable if the mesh is a single, more or less layer surface that is perpendicular to `delaunayaxis`. In case the `delaunayaxis` intersects the mesh surface at more than one point it will result triangles that are very long in the dimension of `delaunayaxis`. If ">0", the alpha-shape algorithm is used. In this case, the positive `alphahull` value signals the use of the alpha-shape algorithm, _and_ its value acts as the parameter for the mesh fitting. If "0", the convex-hull algorithm is used. It is suitable for convex bodies or if the intention is to enclose the `x`, `y` and `z` point set into a convex hull.
- intensity (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the vertex intensity values, used for plotting fields on meshes
- color (`color`)
  Sets the color of the whole mesh
- vertexcolor (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the color of each vertex Overrides "color".
- facecolor (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the color of each face Overrides "color" and "vertexcolor".
- flatshading (`boolean`)
  Determines whether or not normal smoothing is applied to the meshes, creating meshes with an angular, low-poly look via flat reflections.
- contour
  - show (`boolean`)
    Sets whether or not dynamic contours are shown on hover
  - color (`color`)
    default: `"#444"`
    Sets the color of the contour lines.
  - width (`number between or equal to 1 and 16`)
    default: `2`
    Sets the width of the contour lines.
- colorscale (`colorscale`)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- reversescale (`boolean`)
  Reverses the colorscale.
- showscale (`boolean`)
  default: `True`
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (`enumerated: "fraction" | "pixels"`)
    default: `"pixels"`
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (`number greater than or equal to 0`)
    default: `30`
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.

- lenmode (enumerated: "fraction" | "pixels")
  default: "fraction"
  Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
- len (number greater than or equal to 0)
  default: 1
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small

digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (`string`)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (`number greater than or equal to 1`)
  - color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated

- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- lightposition
- x (number between or equal to -100000 and 100000)
  default: 100000
  Numeric vector, representing the X coordinate for each vertex.
- y (number between or equal to -100000 and 100000)
  default: 100000
  Numeric vector, representing the Y coordinate for each vertex.
- z (number between or equal to -100000 and 100000)
  default: 0
  Numeric vector, representing the Z coordinate for each vertex.

- lighting
- vertexnormalsepsilon (number between or equal to 0 and 1)
  default: 1e-12
  Epsilon for vertex normals calculation avoids math issues arising from degenerate geometry.
- facenormalsepsilon (number between or equal to 0 and 1)
  default: 1e-06
  Epsilon for face normals calculation avoids math issues arising from degenerate geometry.
- ambient (number between or equal to 0 and 1)
  default: 0.8
  Ambient light increases overall color visibility but can wash out the image.
- diffuse (number between or equal to 0 and 1)
  default: 0.8
  Represents the extent that incident rays are reflected in a range of angles.
- specular (number between or equal to 0 and 2)
  default: 0.05
  Represents the level that incident rays are reflected in a single direction, causing shine.

- roughness (number between or equal to 0 and 1)
  default: 0.5
  Alters specular reflection; the rougher the surface, the wider and less contrasty the shine.
- fresnel (number between or equal to 0 and 5)
  default: 0.2
  Represents the reflectance as a dependency of the viewing angle; e.g. paper is reflective when viewing it from the edge of the paper (almost 90 degrees), causing shine.

- scene (subplotid)
  default: scene
  Sets a reference between this trace's 3D coordinate system and a 3D scene. If "scene" (the default value), the (x,y,z) coordinates refer to `layout.scene`. If "scene2", the (x,y,z) coordinates refer to `layout.scene2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- ycalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `y` date data.
- zcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `z` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for y . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- isrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for i . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's i data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- jsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for j . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's j data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ksrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for k . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's k data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- intensitysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for intensity . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's intensity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- vertexcolorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for vertexcolor . A source string refers to a unique identifier that is

assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's vertexcolor data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- facecolorsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for facecolor . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's facecolor data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

mesh3d

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- y
- z
- i
- j
- k
- delaunayaxis
- alphahull
- intensity
- color
- vertexcolor
- facecolor
- flatshading
- contour
  - show
  - color
  - width
- colorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick

- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc

- lightposition
  - x
  - y
  - z
- lighting
  - vertexnormalsepsilon
  - facenormalsepsilon
  - ambient
  - diffuse
  - specular
  - roughness
  - fresnel
- scene
- xcalendar
- ycalendar
- zcalendar
- xsrc
- ysrc
- zsrc
- isrc
- jsrc
- ksrc
- intensitysrc
- vertexcolorsrc
- facecolorsrc

地理散点图

```
import plotly.graph_objs as go
go.Scattergeo
```

A `Scattergeo` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"lon"`, `"lat"`, `"location"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
  examples: `"lon"`, `"lat"`, `"lon+lat"`, `"lon+lat+location"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- lon (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the longitude coordinates (in degrees East).
- lat (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the latitude coordinates (in degrees North).
- locations (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the coordinates via location IDs or names. Coordinates correspond to the centroid of each location given. See `locationmode` for more info.
- locationmode (`enumerated: "ISO-3" | "USA-states" | "country names"`)
  default: `"ISO-3"`
  Determines the set of locations used to match entries in `locations` to regions on the map.
- mode (`flaglist string`)
  Any combination of `"lines"`, `"markers"`, `"text"` joined with a `"+"` OR `"none"`.
  examples: `"lines"`, `"markers"`, `"lines+markers"`, `"lines+markers+text"`, `"none"`
  default: `"markers"`
  Determines the drawing mode for this scatter trace. If the provided `mode` includes "text" then the `text` elements appear at the coordinates. Otherwise, the `text` elements appear on hover. If there are less than 20 points, then the default is "lines+markers". Otherwise, "lines".
- text (`string`)
  default: `""`
  Sets text elements associated with each (lon,lat) pair or item in `locations`. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (lon,lat) or `locations` coordinates.
- textfont
  Sets the text font.
  - family (`string`)
  - size (`number greater than or equal to 1`)
  - color (`color`)

- familysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for family . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's family data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "middle center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.
- line
  - color (color)
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- marker
  - symbol (enumerated: "0" | "circle" | "100" | "circle-open" | "200" | "circle-dot" | "300" | "circle-open-dot" | "1" | "square" | "101" | "square-open" | "201" | "square-dot" | "301" | "square-open-dot" | "2" | "diamond" | "102" | "diamond-open" | "202" | "diamond-dot" | "302" | "diamond-open-dot" | "3" | "cross" | "103" | "cross-open" | "203" | "cross-dot" | "303" | "cross-open-dot" | "4" | "x" | "104" | "x-open" | "204" | "x-dot" | "304" | "x-open-dot" | "5" | "triangle-up" | "105" | "triangle-up-open" | "205" | "triangle-up-dot" | "305" | "triangle-up-open-dot" | "6" | "triangle-down" | "106" | "triangle-down-open" | "206" | "triangle-down-dot" | "306" | "triangle-down-open-dot" | "7" | "triangle-left" | "107" | "triangle-left-open" | "207" | "triangle-left-dot" | "307" | "triangle-left-open-dot" | "8" | "triangle-right" | "108" | "triangle-right-open" | "208" | "triangle-right-dot" | "308" | "triangle-right-open-dot" | "9" | "triangle-ne" | "109" | "triangle-ne-open" | "209" | "triangle-ne-dot" | "309" | "triangle-ne-open-dot" | "10" | "triangle-se" | "110" | "triangle-se-open" | "210" | "triangle-se-dot" | "310" | "triangle-se-open-dot" | "11" | "triangle-sw" | "111" | "triangle-sw-open" | "211" | "triangle-sw-dot" | "311" | "triangle-sw-open-dot" | "12" | "triangle-nw" | "112" | "triangle-nw-open" | "212" | "triangle-nw-dot" | "312" | "triangle-nw-open-dot" | "13" | "pentagon" | "113" | "pentagon-open" | "213" | "pentagon-dot" | "313" | "pentagon-open-dot" | "14" | "hexagon" | "114" | "hexagon-open" | "214" | "hexagon-dot" | "314" | "hexagon-open-dot" | "15" | "hexagon2" | "115" | "hexagon2-open" | "215" | "hexagon2-dot" | "315" | "hexagon2-open-dot" | "16" | "octagon" | "116" | "octagon-open" | "216" | "octagon-dot" | "316" | "octagon-open-dot" | "17" | "star" | "117" | "star-open" | "217" | "star-dot" | "317" | "star-open-dot" | "18" | "hexagram" | "118" | "hexagram-open" | "218" | "hexagram-dot" | "318" | "hexagram-open-dot" | "19" | "star-triangle-up" | "119" | "star-triangle-up-open" | "219" | "star-triangle-up-dot" | "319" | "star-triangle-up-open-dot" | "20" | "star-triangle-down" | "120" | "star-triangle-down-open" | "220" | "star-triangle-down-dot" | "320" | "star-triangle-down-open-dot" | "21" | "star-square" | "121" | "star-square-open" | "221" | "star-square-dot" | "321" | "star-square-open-dot" | "22" | "star-diamond" | "122" | "star-diamond-open" | "222" | "star-diamond-dot" | "322" | "star-diamond-open-dot" | "23" | "diamond-tall" | "123" | "diamond-tall-open" | "223" | "diamond-

tall-dot" | "323" | "diamond-tall-open-dot" | "24" | "diamond-wide" | "124" | "diamond-wide-open" | "224" | "diamond-wide-dot" | "324" | "diamond-wide-open-dot" | "25" | "hourglass" | "125" | "hourglass-open" | "26" | "bowtie" | "126" | "bowtie-open" | "27" | "circle-cross" | "127" | "circle-cross-open" | "28" | "circle-x" | "128" | "circle-x-open" | "29" | "square-cross" | "129" | "square-cross-open" | "30" | "square-x" | "130" | "square-x-open" | "31" | "diamond-cross" | "131" | "diamond-cross-open" | "32" | "diamond-x" | "132" | "diamond-x-open" | "33" | "cross-thin" | "133" | "cross-thin-open" | "34" | "x-thin" | "134" | "x-thin-open" | "35" | "asterisk" | "135" | "asterisk-open" | "36" | "hash" | "136" | "hash-open" | "236" | "hash-dot" | "336" | "hash-open-dot" | "37" | "y-up" | "137" | "y-up-open" | "38" | "y-down" | "138" | "y-down-open" | "39" | "y-left" | "139" | "y-left-open" | "40" | "y-right" | "140" | "y-right-open" | "41" | "line-ew" | "141" | "line-ew-open" | "42" | "line-ns" | "142" | "line-ns-open" | "43" | "line-ne" | "143" | "line-ne-open" | "44" | "line-nw" | "144" | "line-nw-open")
default: "circle"
Sets the marker symbol type. Adding 100 is equivalent to appending "-open" to a symbol name. Adding 200 is equivalent to appending "-dot" to a symbol name. Adding 300 is equivalent to appending "-open-dot" or "dot-open" to a symbol name.

- opacity (number between or equal to 0 and 1)
  Sets the marker opacity.
- size (number greater than or equal to 0)
  default: 6
  Sets the marker size (in px).
- sizeref (number)
  default: 1
  Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
- sizemin (number greater than or equal to 0)
  default: 0
  Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.
- sizemode (enumerated: "diameter" | "area")
  default: "diameter"
  Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.

- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.

- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.

- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - line
- width (number greater than or equal to 0)
  Sets the width (in px) of the lines bounding the marker points.
- color (color)
  Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (boolean)
  default: True
  Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (number)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
- cmin (number)
  Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- widthsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for width . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- color (color)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
- cmin (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- symbolsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for symbol . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- opacitysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for opacity . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's opacity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- fill (enumerated: "none" | "toself")
  default: "none"
  Sets the area to fill with a solid color. Use with `fillcolor` if not "none". "toself" connects the endpoints of the trace (or each segment of the trace if it has gaps) into a closed shape.
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- geo (subplotid)
  default: geo

Sets a reference between this trace's geospatial coordinates and a geographic map. If "geo" (the default value), the geospatial coordinates refer to `layout.geo`. If "geo2", the geospatial coordinates refer to `layout.geo2`, and so on.

- lonsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for lon . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's lon data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- latsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for lat . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's lat data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- locationssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for locations . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's locations data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textpositionsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for textposition . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's textposition data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

scattergeo

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- lon
- lat
- locations
- locationmode
- mode
- text
- textfont
  - family
  - size
  - color
  - familysrc
  - sizesrc
  - colorsrc
- textposition
- line
  - color
  - width
  - dash
- connectgaps
- marker
  - symbol

- opacity
- size
- sizeref
- sizemin
- sizemode
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - tickfont
    - family
    - size
    - color
  - tickangle
  - tickformat
  - tickprefix
  - showtickprefix
  - ticksuffix
  - showticksuffix
  - separatethousands
  - exponentformat
  - showexponent
  - title
  - titlefont
    - family
    - size
    - color
  - titleside
  - tickvalssrc
  - ticktextsrc
- line
  - width
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - widthsrc

- colorsrc
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - symbolsrc
  - opacitysrc
  - sizesrc
  - colorsrc
- fill
- fillcolor
- geo
- lonsrc
- latsrc
- locationssrc
- textsrc
- textpositionsrc

等值线图

```
import plotly.graph_objs as go
go.Choropleth
```

A `Choropleth` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"location"`, `"z"`, `"text"`, `"name"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
  examples: `"location"`, `"z"`, `"location+z"`, `"location+z+text"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.

- maxpoints (number between or equal to 0 and 10000)
  default: 500
  Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- locations (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the coordinates via location IDs or names. See `locationmode` for more info.
- locationmode (enumerated: "ISO-3" | "USA-states" | "country names")
  default: "ISO-3"
  Determines the set of locations used to match entries in `locations` to regions on the map.
- z (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the color values.
- text (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text elements associated with each location.
- marker
  - line
    - color (color)
      Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
    - width (number greater than or equal to 0)
      default: 1
      Sets the width (in px) of the lines bounding the marker points.
    - colorsrc (source string | plotly.grid objs.Column)
      Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
    - widthsrc (source string | plotly.grid objs.Column)
      Sets the source reference on plot.ly for width . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zauto (boolean)
  default: True
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (number)
  Sets the lower bound of color domain.
- zmax (number)
  Sets the upper bound of color domain.
- colorscale (colorscale)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (boolean)
  default: True
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (boolean)
  Reverses the colorscale.
- showscale (boolean)
  default: True
  Determines whether or not a colorbar is displayed for this trace.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"

Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.

- len (number greater than or equal to 0)
  default: 1
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day,

set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  default: `""`
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (`string`)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (`number greater than or equal to 1`)
  - color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated
- exponentformat (`enumerated: "none" | "e" | "E" | "power" | "SI" | "B"`)
  default: `"B"`

Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.

- showexponent (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (`string`)
  default: `"Click to enter colorscale title"`
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (`string`)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (`number greater than or equal to 1`)
  - color (`color`)
- titleside (`enumerated: "right" | "top" | "bottom"`)
  default: `"top"`
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for tickvals . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for ticktext . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- geo (`subplotid`)
  default: `geo`
  Sets a reference between this trace's geospatial coordinates and a geographic map. If "geo" (the default value), the geospatial coordinates refer to `layout.geo`. If "geo2", the geospatial coordinates refer to `layout.geo2`, and so on.
- locationssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for locations . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's locations data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for z . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

choropleth

- visible
- showlegend
- legendgroup
- opacity
- name

- hoverinfo
- stream
  - token
  - maxpoints
- locations
- locationmode
- z
- text
- marker
  - line
    - color
    - width
    - colorsrc
    - widthsrc
  - zauto
  - zmin
  - zmax
  - colorscale
  - autocolorscale
  - reversescale
  - showscale
  - colorbar
    - thicknessmode
    - thickness
    - lenmode
    - len
    - x
    - xanchor
    - xpad
    - y
    - yanchor
    - ypad
    - outlinecolor
    - outlinewidth
    - bordercolor
    - borderwidth
    - bgcolor
    - tickmode
    - nticks
    - tick0
    - dtick
    - tickvals
    - ticktext
    - ticks
    - ticklen
    - tickwidth
    - tickcolor
    - showticklabels
    - tickfont
      - family
      - size
      - color
    - tickangle
    - tickformat
    - tickprefix
    - showtickprefix
    - ticksuffix
    - showticksuffix
    - separatethousands
    - exponentformat
    - showexponent
    - title
    - titlefont
      - family

- size
- color
  - titleside
  - tickvalssrc
  - ticktextsrc
- geo
- locationssrc
- zsrc
- textsrc

---

gl 散点图

```
import plotly.graph_objs as go
go.Scattergl
```

A `Scattergl` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- x0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (`number`)
  default: `1`
  Sets the x coordinate step. See `x0` for more info.

- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- y0 (number or categorical coordinate string)
  default: 0
  Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.
- dy (number)
  default: 1
  Sets the y coordinate step. See `y0` for more info.
- text (string)
  default: ""
  Sets text elements associated with each (x,y) pair to appear on hover. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y) coordinates.
- mode (flaglist string)
  Any combination of "lines", "markers" joined with a "+" OR "none".
  examples: "lines", "markers", "lines+markers", "none"
  Determines the drawing mode for this scatter trace.
- line
  - color (color)
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (enumerated: "solid" | "dot" | "dash" | "longdash" | "dashdot" | "longdashdot")
    default: "solid"
    Sets the style of the lines.
- marker
  - color (color)
    Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - colorscale (colorscale)
    Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
  - cauto (boolean)
    default: True
    Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
  - cmax (number)
    Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
  - cmin (number)
    Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
  - autocolorscale (boolean)
    default: True
    Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
  - reversescale (boolean)
    Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
  - symbol (enumerated: "circle" | "circle-open" | "square" | "square-open" | "diamond" | "diamond-open" | "cross" | "x")
    default: "circle"
    Sets the marker symbol type.

- size (number greater than or equal to 0)
  default: 6
  Sets the marker size (in px).
- sizeref (number)
  default: 1
  Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
- sizemin (number greater than or equal to 0)
  default: 0
  Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.
- sizemode (enumerated: "diameter" | "area")
  default: "diameter"
  Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.
- opacity (number between or equal to 0 and 1)
  Sets the marker opacity.
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.
  - y (number between or equal to -2 and 3)
    default: 0.5
    Sets the y position of the color bar (in plot fraction).
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "middle"
    Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
  - ypad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the y direction.
  - outlinecolor (color)
    default: "#444"
    Sets the axis line color.
  - outlinewidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the axis line.
  - bordercolor (color)
    default: "#444"
    Sets the axis line color.

- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono",

"Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".

- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned

to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- line
  - color (`color`)
    Sets the marker.line color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - colorscale (`colorscale`)
    Sets the colorscale and only has an effect if `marker.line.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.line.cmin` and `marker.line.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
  - cauto (`boolean`)
    default: `True`
    Has an effect only if `marker.line.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
  - cmax (`number`)
    Has an effect only if `marker.line.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmin` must be set as well.
  - cmin (`number`)
    Has an effect only if `marker.line.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.line.color` array index, and if set, `marker.line.cmax` must be set as well.
  - autocolorscale (`boolean`)
    default: `True`
    Has an effect only if `marker.line.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.line.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
  - reversescale (`boolean`)
    Has an effect only if `marker.line.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
  - width (`number greater than or equal to 0`)
    Sets the width (in px) of the lines bounding the marker points.
  - colorsrc (`source string | plotly.grid objs.Column`)
    Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - widthsrc (`source string | plotly.grid objs.Column`)
    Sets the source reference on plot.ly for width . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's width data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for color . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- symbolsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for symbol . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for size . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- opacitysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for opacity . A `source string` refers to a unique identifier that is assigned

to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's opacity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- fill (enumerated: "none" | "tozeroy" | "tozerox")
  default: "none"
  Sets the area to fill with a solid color. Use with `fillcolor` if not "none". "tozerox" and "tozeroy" fill to x=0 and y=0 respectively. "tonextx" and "tonexty" fill between the endpoints of this trace and the endpoints of the trace before it, connecting those endpoints with straight lines (to make a stacked area graph); if there is no trace before it, they behave like "tozerox" and "tozeroy". "toself" connects the endpoints of the trace (or each segment of the trace if it has gaps) into a closed shape. "tonext" fills the space between two traces if one completely encloses the other (eg consecutive contour lines), and behaves like "toself" if there is no trace before it. "tonext" should not be used if one trace does not enclose the other.
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- error_y
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
  - value (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
  - valueminus (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
  - traceref (integer greater than or equal to 0)
    default: 0
  - tracerefminus (integer greater than or equal to 0)
    default: 0
  - copy_ystyle (boolean)
  - copy_zstyle (boolean)
  - color (color)
    Sets the stoke color of the error bars.
  - thickness (number greater than or equal to 0)
    default: 2
    Sets the thickness (in px) of the error bars.
  - width (number greater than or equal to 0)
    Sets the width (in px) of the cross-bar at both ends of the error bars.
  - arraysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - arrayminussrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's

arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- error_x
  - visible (boolean)
    Determines whether or not this set of error bars is visible.
  - type (enumerated: "percent" | "constant" | "sqrt" | "data")
    Determines the rule used to generate the error bars. If "constant`, the bar lengths are of a constant value. Set this constant in `value`. If "percent", the bar lengths correspond to a percentage of underlying data. Set this percentage in `value`. If "sqrt", the bar lengths correspond to the sqaure of the underlying data. If "array", the bar lengths are set with data set `array`.
  - symmetric (boolean)
    Determines whether or not the error bars have the same length in both direction (top/bottom for vertical bars, left/right for horizontal bars.
  - array (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar. Values are plotted relative to the underlying data.
  - arrayminus (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the data corresponding the length of each error bar in the bottom (left) direction for vertical (horizontal) bars Values are plotted relative to the underlying data.
  - value (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars.
  - valueminus (number greater than or equal to 0)
    default: 10
    Sets the value of either the percentage (if `type` is set to "percent") or the constant (if `type` is set to "constant") corresponding to the lengths of the error bars in the bottom (left) direction for vertical (horizontal) bars
  - traceref (integer greater than or equal to 0)
    default: 0
  - tracerefminus (integer greater than or equal to 0)
    default: 0
  - copy_ystyle (boolean)
  - copy_zstyle (boolean)
  - color (color)
    Sets the stoke color of the error bars.
  - thickness (number greater than or equal to 0)
    default: 2
    Sets the thickness (in px) of the error bars.
  - width (number greater than or equal to 0)
    Sets the width (in px) of the cross-bar at both ends of the error bars.
  - arraysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for array . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's array data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - arrayminussrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for arrayminus . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's arrayminus data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.

- ycalendar (`enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura"`)
  default: `"gregorian"`
  Sets the calendar system to use with `y` date data.
- xsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for x . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for y . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

scattergl

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- x
- x0
- dx
- y
- y0
- dy
- text
- mode
- line
  - color
  - width
  - dash
- marker
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - symbol
  - size
  - sizeref
  - sizemin
  - sizemode
  - opacity
  - showscale
  - colorbar
    - thicknessmode

- thickness
- lenmode
- len
- x
- xanchor
- xpad
- y
- yanchor
- ypad
- outlinecolor
- outlinewidth
- bordercolor
- borderwidth
- bgcolor
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
- family
- size
- color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
- family
- size
- color
- titleside
- tickvalssrc
- ticktextsrc
- line
- color
- colorscale
- cauto
- cmax
- cmin
- autocolorscale
- reversescale
- width
- colorsrc
- widthsrc
- colorsrc
- symbolsrc
- sizesrc
- opacitysrc
- connectgaps
- fill

- fillcolor
- error_y
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- error_x
  - visible
  - type
  - symmetric
  - array
  - arrayminus
  - value
  - valueminus
  - traceref
  - tracerefminus
  - copy_ystyle
  - copy_zstyle
  - color
  - thickness
  - width
  - arraysrc
  - arrayminussrc
- xaxis
- yaxis
- xcalendar
- ycalendar
- xsrc
- ysrc
- textsrc

点云图

```
import plotly.graph_objs as go
go.Pointcloud
```

A `Pointcloud` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.

- legendgroup (string)
  default: ""
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the trace.
- name (string)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (flaglist string)
  Any combination of "x", "y", "z", "text", "name" joined with a "+" OR "all" or "none" or "skip".
  examples: "x", "y", "x+y", "x+y+z", "all"
  default: "all"
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (string)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (number between or equal to 0 and 10000)
    default: 500
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the x coordinates.
- y (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the y coordinates.
- xy (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Faster alternative to specifying `x` and `y` separately. If supplied, it must be a typed `Float32Array` array that represents points such that `xy[i * 2] = x[i]` and `xy[i * 2 + 1] = y[i]`
- indices (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  A sequential value, 0..n, supply it to avoid creating this array inside plotting. If specified, it must be a typed `Int32Array` array. Its length must be equal to or greater than the number of points. For the best performance and memory use, create one large `indices` typed array that is guaranteed to be at least as long as the largest number of points during use, and reuse it on each `Plotly.restyle()` call.
- xbounds (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Specify `xbounds` in the shape of `[xMin, xMax] to avoid looping through the `xy` typed array. Use it in conjunction with `xy` and `ybounds` for the performance benefits.
- ybounds (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Specify `ybounds` in the shape of `[yMin, yMax] to avoid looping through the `xy` typed array. Use it in conjunction with `xy` and `xbounds` for the performance benefits.
- text (string)
  default: ""
  Sets text elements associated with each (x,y) pair to appear on hover. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (x,y) coordinates.
- marker
  - color (color)
    Sets the marker fill color. It accepts a specific color.If the color is not fully opaque and there are hundreds of thousandsof points, it may cause slower zooming and panning.
  - opacity (number between or equal to 0 and 1)
    default: 1
    Sets the marker opacity. The default value is `1` (fully opaque). If the markers are not fully opaque and there are hundreds of thousands of points, it may cause slower zooming and panning. Opacity fades the color even if `blend` is left on `False` even if there is no translucency effect in that case.
  - blend (boolean)
    Determines if colors are blended together for a translucency effect in case `opacity` is specified as a value less then `1`. Setting `blend` to `True` reduces zoom/pan speed if used with large numbers of points.
  - sizemin (number between or equal to 0.1 and 2)
    default: 0.5
    Sets the minimum size (in px) of the rendered marker points, effective when the `pointcloud` shows a million or more points.
  - sizemax (number greater than or equal to 0.1)
    default: 20

Sets the maximum size (in px) of the rendered marker points. Effective when the `pointcloud` shows only few points.

- border
  - color (`color`)
    Sets the stroke color. It accepts a specific color. If the color is not fully opaque and there are hundreds of thousands of points, it may cause slower zooming and panning.
  - arearatio (`number between or equal to 0 and 1`)
    default: `0`
    Specifies what fraction of the marker area is covered with the border.
- xaxis (`subplotid`)
  default: `x`
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (`subplotid`)
  default: `y`
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for x . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for y . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for xy . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's xy data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- indicessrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for indices . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's indices data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xboundssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for xbounds . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's xbounds data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- yboundssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for ybounds . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ybounds data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

pointcloud

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo

- stream
  - token
  - maxpoints
- x
- y
- xy
- indices
- xbounds
- ybounds
- text
- marker
  - color
  - opacity
  - blend
  - sizemin
  - sizemax
  - border
    - color
    - arearatio
- xaxis
- yaxis
- xsrc
- ysrc
- xysrc
- indicessrc
- xboundssrc
- yboundssrc
- textsrc

---

gl 热点图

```
import plotly.graph_objs as go
go.Heatmapgl
```

A `Heatmapgl` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.

examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
default: `"all"`
Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.

- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- z (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the z data.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates.
- x0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `x`. Builds a linear space of x coordinates. Use with `dx` where `x0` is the starting coordinate and `dx` the step.
- dx (`number`)
  default: `1`
  Sets the x coordinate step. See `x0` for more info.
- y (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the y coordinates.
- y0 (`number or categorical coordinate string`)
  default: `0`
  Alternate to `y`. Builds a linear space of y coordinates. Use with `dy` where `y0` is the starting coordinate and `dy` the step.
- dy (`number`)
  default: `1`
  Sets the y coordinate step. See `y0` for more info.
- text (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text elements associated with each z value.
- transpose (`boolean`)
  Transposes the z data.
- xtype (`enumerated: "array" | "scaled"`)
  If "array", the heatmap's x coordinates are given by "x" (the default behavior when `x` is provided). If "scaled", the heatmap's x coordinates are given by "x0" and "dx" (the default behavior when `x` is not provided).
- ytype (`enumerated: "array" | "scaled"`)
  If "array", the heatmap's y coordinates are given by "y" (the default behavior when `y` is provided) If "scaled", the heatmap's y coordinates are given by "y0" and "dy" (the default behavior when `y` is not provided)
- zauto (`boolean`)
  default: `True`
  Determines the whether or not the color domain is computed with respect to the input data.
- zmin (`number`)
  Sets the lower bound of color domain.
- zmax (`number`)
  Sets the upper bound of color domain.
- colorscale (`colorscale`)
  Sets the colorscale. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in z space, use zmin and zmax
- autocolorscale (`boolean`)
  Determines whether or not the colorscale is picked using the sign of the input z values.
- reversescale (`boolean`)
  Reverses the colorscale.
- showscale (`boolean`)
  default: `True`
  Determines whether or not a colorbar is displayed for this trace.
- colorbar

- thicknessmode (enumerated: "fraction" | "pixels")
  default: "pixels"
  Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
- thickness (number greater than or equal to 0)
  default: 30
  Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
- lenmode (enumerated: "fraction" | "pixels")
  default: "fraction"
  Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
- len (number greater than or equal to 0)
  default: 1
  Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position of the color bar (in plot fraction).
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
- xpad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the x direction.
- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is

"category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.

- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"

If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.

- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- zsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for z . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's z data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- xsrc (source string | plotly.grid obis.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful

if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- ysrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for y . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's y data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

heatmapgl

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- z
- x
- x0
- dx
- y
- y0
- dy
- text
- transpose
- xtype
- ytype
- zauto
- zmin
- zmax
- colorscale
- autocolorscale
- reversescale
- showscale
- colorbar
  - thicknessmode
  - thickness
  - lenmode
  - len
  - x
  - xanchor
  - xpad
  - y
  - yanchor
  - ypad
  - outlinecolor
  - outlinewidth
  - bordercolor
  - borderwidth
  - bgcolor
  - tickmode
  - nticks

- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc

- xaxis
- yaxis
- zsrc
- xsrc
- ysrc
- textsrc

---

mapbox 散点图

```
import plotly.graph_objs as go
go.Scattermapbox
```

A `Scattermapbox` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.

- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the trace.
- name (string)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (flaglist string)
  Any combination of "lon", "lat", "text", "name", "name" joined with a "+" OR "all" or "none" or "skip".
  examples: "lon", "lat", "lon+lat", "lon+lat+text", "all"
  default: "all"
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (string)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (number between or equal to 0 and 10000)
    default: 500
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- lon (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the longitude coordinates (in degrees East).
- lat (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the latitude coordinates (in degrees North).
- mode (flaglist string)
  Any combination of "lines", "markers", "text" joined with a "+" OR "none".
  examples: "lines", "markers", "lines+markers", "lines+markers+text", "none"
  default: "markers"
  Determines the drawing mode for this scatter trace. If the provided `mode` includes "text" then the `text` elements appear at the coordinates. Otherwise, the `text` elements appear on hover.
- text (string)
  default: ""
  Sets text elements associated with each (lon,lat) pair If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to the this trace's (lon,lat) coordinates.
- line
  - color (color)
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
- connectgaps (boolean)
  Determines whether or not gaps (i.e. {nan} or missing values) in the provided data arrays are connected.
- marker
  - symbol (string)
    default: "circle"
    Sets the marker symbol. Full list: https://www.mapbox.com/maki-icons/ Note that the array `marker.color` and `marker.size` are only available for "circle" symbols.
  - opacity (number between or equal to 0 and 1)
    Sets the marker opacity.
  - size (number greater than or equal to 0)
    default: 6
    Sets the marker size (in px).
  - sizeref (number)
    default: 1
    Has an effect only if `marker.size` is set to a numerical array. Sets the scale factor used to determine the rendered size of marker points. Use with `sizemin` and `sizemode`.
  - sizemin (number greater than or equal to 0)
    default: 0
    Has an effect only if `marker.size` is set to a numerical array. Sets the minimum size (in px) of the rendered marker points.

- sizemode (enumerated: "diameter" | "area")
  default: "diameter"
  Has an effect only if `marker.size` is set to a numerical array. Sets the rule for which the data in `size` is converted to pixels.
- color (color)
  Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
- colorscale (colorscale)
  Sets the colorscale and only has an effect if `marker.color` is set to a numerical array. The colorscale must be an array containing arrays mapping a normalized value to an rgb, rgba, hex, hsl, hsv, or named color string. At minimum, a mapping for the lowest (0) and highest (1) values are required. For example, `[[0, 'rgb(0,0,255)', [1, 'rgb(255,0,0)']]`. To control the bounds of the colorscale in color space, use `marker.cmin` and `marker.cmax`. Alternatively, `colorscale` may be a palette name string of the following list: Greys, YlGnBu, Greens, YlOrRd, Bluered, RdBu, Reds, Blues, Picnic, Rainbow, Portland, Jet, Hot, Blackbody, Earth, Electric, Viridis
- cauto (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array and `cmin`, `cmax` are set by the user. In this case, it controls whether the range of colors in `colorscale` is mapped to the range of values in the `color` array (`cauto: True`), or the `cmin`/`cmax` values (`cauto: False`). Defaults to `False` when `cmin`, `cmax` are set by the user.
- cmax (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the upper bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmin` must be set as well.
- cmin (number)
  Has an effect only if `marker.color` is set to a numerical array. Sets the lower bound of the color domain. Value should be associated to the `marker.color` array index, and if set, `marker.cmax` must be set as well.
- autocolorscale (boolean)
  default: True
  Has an effect only if `marker.color` is set to a numerical array. Determines whether the colorscale is a default palette (`autocolorscale: True`) or the palette determined by `marker.colorscale`. In case `colorscale` is unspecified or `autocolorscale` is True, the default palette will be chosen according to whether numbers in the `color` array are all positive, all negative or mixed.
- reversescale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Reverses the color mapping if True (`cmin` will correspond to the last color in the array and `cmax` will correspond to the first color).
- showscale (boolean)
  Has an effect only if `marker.color` is set to a numerical array. Determines whether or not a colorbar is displayed.
- colorbar
  - thicknessmode (enumerated: "fraction" | "pixels")
    default: "pixels"
    Determines whether this color bar's thickness (i.e. the measure in the constant color direction) is set in units of plot "fraction" or in "pixels". Use `thickness` to set the value.
  - thickness (number greater than or equal to 0)
    default: 30
    Sets the thickness of the color bar This measure excludes the size of the padding, ticks and labels.
  - lenmode (enumerated: "fraction" | "pixels")
    default: "fraction"
    Determines whether this color bar's length (i.e. the measure in the color variation direction) is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
  - len (number greater than or equal to 0)
    default: 1
    Sets the length of the color bar This measure excludes the padding of both ends. That is, the color bar length is this length minus the padding on both ends.
  - x (number between or equal to -2 and 3)
    default: 1.02
    Sets the x position of the color bar (in plot fraction).
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets this color bar's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the color bar.
  - xpad (number greater than or equal to 0)
    default: 10
    Sets the amount of padding (in px) along the x direction.

- y (number between or equal to -2 and 3)
  default: 0.5
  Sets the y position of the color bar (in plot fraction).
- yanchor (enumerated: "top" | "middle" | "bottom")
  default: "middle"
  Sets this color bar's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the color bar.
- ypad (number greater than or equal to 0)
  default: 10
  Sets the amount of padding (in px) along the y direction.
- outlinecolor (color)
  default: "#444"
  Sets the axis line color.
- outlinewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- bordercolor (color)
  default: "#444"
  Sets the axis line color.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) or the border enclosing this color bar.
- bgcolor (color)
  default: "rgba(0,0,0,0)"
  Sets the color of padded area.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  default: ""
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.

- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- title (string)
  default: "Click to enter colorscale title"
  Sets the title of the color bar.
- titlefont
  Sets this color bar's title font.

- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- titleside (enumerated: "right" | "top" | "bottom")
  default: "top"
  Determines the location of the colorbar title with respect to the color bar.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- symbolsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for symbol . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- fill (enumerated: "none" | "toself")
  default: "none"
  Sets the area to fill with a solid color. Use with `fillcolor` if not "none". "toself" connects the endpoints of the trace (or each segment of the trace if it has gaps) into a closed shape.
- fillcolor (color)
  Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
- textfont
  Sets the icon text font. Has an effect only when `type` is set to "symbol".
  - family (string)
    default: "Open Sans Regular, Arial Unicode MS Regular"
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "middle center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.

- subplot (subplotid)
  default: mapbox
  Sets a reference between this trace's data coordinates and a mapbox subplot. If "mapbox" (the default value), the data refer to `layout.mapbox`. If "mapbox2", the data refer to `layout.mapbox2`, and so on.
- lonsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for lon . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's lon data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- latsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for lat . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's lat data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for text . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

scattermapbox

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- lon
- lat
- mode
- text
- line
  - color
  - width
  - dash
- connectgaps
- marker
  - symbol
  - opacity
  - size
  - sizeref
  - sizemin
  - sizemode
  - color
  - colorscale
  - cauto
  - cmax
  - cmin
  - autocolorscale
  - reversescale
  - showscale
  - colorbar
    - thicknessmode
    - thickness
    - lenmode
    - len
    - x

- xanchor
- xpad
- y
- yanchor
- ypad
- outlinecolor
- outlinewidth
- bordercolor
- borderwidth
- bgcolor
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickformat
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- separatethousands
- exponentformat
- showexponent
- title
- titlefont
  - family
  - size
  - color
- titleside
- tickvalssrc
- ticktextsrc
- symbolsrc
- sizesrc
- colorsrc
- fill
- fillcolor
- textfont
  - family
  - size
  - color
- textposition
- subplot
- lonsrc
- latsrc
- textsrc

---

ohlc 简化蜡烛图

```
import plotly.graph_objs as go
go.Ohlc
```
A `Ohlc` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the x coordinates. If absent, linear coordinate will be generated.
- open (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  default:
  Sets the open values.
- high (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  default:
  Sets the high values.
- low (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  default:
  Sets the low values.
- close (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  default:
  Sets the close values.
- line
  - width (`number greater than or equal to 0`)
    default: `2`
    [object Object] Note that this style setting can also be set per direction via `increasing.line.width` and `decreasing.line.width`.
  - dash (`string`)
    default: `"solid"`
    [object Object] Note that this style setting can also be set per direction via `increasing.line.dash` and `decreasing.line.dash`.
- increasing
  - name (`string`)
    Sets the segment name. The segment name appear as the legend item and on hover.

- showlegend (boolean)
  default: True
  Determines whether or not an item corresponding to this segment is shown in the legend.
- line
  - color (color)
    default: "#3D9970"
    Sets the line color.
  - width (number greater than or equal to 0)
    default: 2
    Sets the line width (in px).
  - dash (string)
    default: "solid"
    Sets the style of the lines. Set to a dash string type or a dash length in px.
- decreasing
  - name (string)
    Sets the segment name. The segment name appear as the legend item and on hover.
  - showlegend (boolean)
    default: True
    Determines whether or not an item corresponding to this segment is shown in the legend.
  - line
    - color (color)
      default: "#FF4136"
      Sets the line color.
    - width (number greater than or equal to 0)
      default: 2
      Sets the line width (in px).
    - dash (string)
      default: "solid"
      Sets the style of the lines. Set to a dash string type or a dash length in px.
- text (string)
  default: ""
  Sets hover text elements associated with each sample point. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to this trace's sample points.
- tickwidth (number between or equal to 0 and 0.5)
  default: 0.3
  Sets the width of the open/close tick marks relative to the "x" minimal interval.
- xaxis (subplotid)
  default: x
  Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
- yaxis (subplotid)
  default: y
  Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
- xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use with `x` date data.
- xsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- opensrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for open . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's open data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- highsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for high . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's high data. It's

useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- lowsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for low . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's low data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- closesrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for close . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's close data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- textsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

ohlc

visible
showlegend
legendgroup
opacity
name
hoverinfo
stream
token
maxpoints
x
open
high
low
close
line
width
dash
increasing
name
showlegend
line
color
width
dash
decreasing
name
showlegend
line
color
width
dash
text
tickwidth
xaxis
yaxis
xcalendar
xsrc
opensrc
highsrc
lowsrc
closesrc
textsrc

蜡烛图

```
import plotly.graph_objs as go
go.Candlestick
```
A `Candlestick` trace is a graph object with any of the named arguments or attributes listed below.

visible (`enumerated: True | False | "legendonly"`)
default: `True`
Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
showlegend (`boolean`)
default: `True`
Determines whether or not an item corresponding to this trace is shown in the legend.
legendgroup (`string`)
default: `""`
Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
opacity (`number between or equal to 0 and 1`)
default: `1`
Sets the opacity of the trace.
name (`string`)
Sets the trace name. The trace name appear as the legend item and on hover.
hoverinfo (`flaglist string`)
Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`.
examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
default: `"all"`
Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
stream
token (`string`)
The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
maxpoints (`number between or equal to 0 and 10000`)
default: `500`
Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
x (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
Sets the x coordinates. If absent, linear coordinate will be generated.
open (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
default:
Sets the open values.
high (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
default:
Sets the high values.
low (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
default:
Sets the low values.
close (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
default:
Sets the close values.
line
width (`number greater than or equal to 0`)
default: `2`
Sets the width (in px) of line bounding the box(es). Note that this style setting can also be set per direction via `increasing.line.width` and `decreasing.line.width`.
increasing
name (`string`)
Sets the segment name. The segment name appear as the legend item and on hover.

showlegend (boolean)
default: True
Determines whether or not an item corresponding to this segment is shown in the legend.
line
color (color)
default: "#3D9970"
Sets the color of line bounding the box(es).
width (number greater than or equal to 0)
default: 2
Sets the width (in px) of line bounding the box(es).
fillcolor (color)
Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
decreasing
name (string)
Sets the segment name. The segment name appear as the legend item and on hover.
showlegend (boolean)
default: True
Determines whether or not an item corresponding to this segment is shown in the legend.
line
color (color)
default: "#FF4136"
Sets the color of line bounding the box(es).
width (number greater than or equal to 0)
default: 2
Sets the width (in px) of line bounding the box(es).
fillcolor (color)
Sets the fill color. Defaults to a half-transparent variant of the line color, marker color, or marker line color, whichever is available.
text (string)
default: ""
Sets hover text elements associated with each sample point. If a single string, the same string appears over all the data points. If an array of string, the items are mapped in order to this trace's sample points.
whiskerwidth (number between or equal to 0 and 1)
default: 0
Sets the width of the whiskers relative to the box' width. For example, with 1, the whiskers are as wide as the box(es).
xaxis (subplotid)
default: x
Sets a reference between this trace's x coordinates and a 2D cartesian x axis. If "x" (the default value), the x coordinates refer to `layout.xaxis`. If "x2", the x coordinates refer to `layout.xaxis2`, and so on.
yaxis (subplotid)
default: y
Sets a reference between this trace's y coordinates and a 2D cartesian y axis. If "y" (the default value), the y coordinates refer to `layout.yaxis`. If "y2", the y coordinates refer to `layout.xaxis2`, and so on.
xcalendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
default: "gregorian"
Sets the calendar system to use with `x` date data.
xsrc (source string | plotly.grid objs.Column)
Sets the source reference on plot.ly for x . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's x data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
opensrc (source string | plotly.grid objs.Column)
Sets the source reference on plot.ly for open . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's open data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
highsrc (source string | plotly.grid objs.Column)
Sets the source reference on plot.ly for high . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's high data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

lowsrc (`source string | plotly.grid objs.Column`)
Sets the source reference on plot.ly for low . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's low data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
closesrc (`source string | plotly.grid objs.Column`)
Sets the source reference on plot.ly for close . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's close data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
textsrc (`source string | plotly.grid objs.Column`)
Sets the source reference on plot.ly for text . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's text data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

candlestick

visible
showlegend
legendgroup
opacity
name
hoverinfo
stream
token
maxpoints
x
open
high
low
close
line
width
increasing
name
showlegend
line
color
width
fillcolor
decreasing
name
showlegend
line
color
width
fillcolor
text
whiskerwidth
xaxis
yaxis
xcalendar
xsrc
opensrc
highsrc
lowsrc
closesrc
textsrc

面积图

```
import plotly.graph_objs as go
go.Area
```
A `Area` trace is a graph object with any of the named arguments or attributes listed below.

- visible (`enumerated: True | False | "legendonly"`)
  default: `True`
  Determines whether or not this trace is visible. If "legendonly", the trace is not drawn, but can appear as a legend item (provided that the legend itself is visible).
- showlegend (`boolean`)
  default: `True`
  Determines whether or not an item corresponding to this trace is shown in the legend.
- legendgroup (`string`)
  default: `""`
  Sets the legend group for this trace. Traces part of the same legend group hide/show at the same time when toggling legend items.
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the trace.
- name (`string`)
  Sets the trace name. The trace name appear as the legend item and on hover.
- hoverinfo (`flaglist string`)
  Any combination of `"x"`, `"y"`, `"z"`, `"text"`, `"name"` joined with a `"+"` OR `"all"` or `"none"` or `"skip"`. examples: `"x"`, `"y"`, `"x+y"`, `"x+y+z"`, `"all"`
  default: `"all"`
  Determines which trace information appear on hover. If `none` or `skip` are set, no information is displayed upon hovering. But, if `none` is set, click and hover events are still fired.
- stream
  - token (`string`)
    The stream id number links a data trace on a plot with a stream. See https://plot.ly/settings for more details.
  - maxpoints (`number between or equal to 0 and 10000`)
    default: `500`
    Sets the maximum number of points to keep on the plots from an incoming stream. If `maxpoints` is set to "50", only the newest 50 points will be displayed on the plot.
- r (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  For polar chart only.Sets the radial coordinates.
- t (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  For polar chart only.Sets the angular coordinates.
- marker
  - color (`color`)
    Sets the marker color. It accepts either a specific color or an array of numbers that are mapped to the colorscale relative to the max and min values of the array or relative to `cmin` and `cmax` if set.
  - size (`number greater than or equal to 0`)
    default: `6`
    Sets the marker size (in px).
  - symbol (`enumerated: "0" | "circle" | "100" | "circle-open" | "200" | "circle-dot" | "300" | "circle-open-dot" | "1" | "square" | "101" | "square-open" | "201" | "square-dot" | "301" | "square-open-dot" | "2" | "diamond" | "102" | "diamond-open" | "202" | "diamond-dot" | "302" | "diamond-open-dot" | "3" | "cross" | "103" | "cross-open" | "203" | "cross-dot" | "303" | "cross-open-dot" | "4" | "x" | "104" | "x-open" | "204" | "x-dot" | "304" | "x-open-dot" | "5" | "triangle-up" | "105" | "triangle-up-open" | "205" | "triangle-up-dot" | "305" | "triangle-up-open-dot" | "6" | "triangle-down" | "106" | "triangle-down-open" | "206" | "triangle-down-dot" | "306" | "triangle-down-open-dot" | "7" | "triangle-left" | "107" | "triangle-left-open" | "207" | "triangle-left-dot" | "307" | "triangle-left-open-dot" | "8" | "triangle-right" | "108" | "triangle-right-open" | "208" | "triangle-right-dot" | "308" | "triangle-right-open-dot" | "9" | "triangle-ne" | "109" | "triangle-`

ne-open" | "209" | "triangle-ne-dot" | "309" | "triangle-ne-open-dot" | "10" | "triangle-se" | "110" | "triangle-se-open" | "210" | "triangle-se-dot" | "310" | "triangle-se-open-dot" | "11" | "triangle-sw" | "111" | "triangle-sw-open" | "211" | "triangle-sw-dot" | "311" | "triangle-sw-open-dot" | "12" | "triangle-nw" | "112" | "triangle-nw-open" | "212" | "triangle-nw-dot" | "312" | "triangle-nw-open-dot" | "13" | "pentagon" | "113" | "pentagon-open" | "213" | "pentagon-dot" | "313" | "pentagon-open-dot" | "14" | "hexagon" | "114" | "hexagon-open" | "214" | "hexagon-dot" | "314" | "hexagon-open-dot" | "15" | "hexagon2" | "115" | "hexagon2-open" | "215" | "hexagon2-dot" | "315" | "hexagon2-open-dot" | "16" | "octagon" | "116" | "octagon-open" | "216" | "octagon-dot" | "316" | "octagon-open-dot" | "17" | "star" | "117" | "star-open" | "217" | "star-dot" | "317" | "star-open-dot" | "18" | "hexagram" | "118" | "hexagram-open" | "218" | "hexagram-dot" | "318" | "hexagram-open-dot" | "19" | "star-triangle-up" | "119" | "star-triangle-up-open" | "219" | "star-triangle-up-dot" | "319" | "star-triangle-up-open-dot" | "20" | "star-triangle-down" | "120" | "star-triangle-down-open" | "220" | "star-triangle-down-dot" | "320" | "star-triangle-down-open-dot" | "21" | "star-square" | "121" | "star-square-open" | "221" | "star-square-dot" | "321" | "star-square-open-dot" | "22" | "star-diamond" | "122" | "star-diamond-open" | "222" | "star-diamond-dot" | "322" | "star-diamond-open-dot" | "23" | "diamond-tall" | "123" | "diamond-tall-open" | "223" | "diamond-tall-dot" | "323" | "diamond-tall-open-dot" | "24" | "diamond-wide" | "124" | "diamond-wide-open" | "224" | "diamond-wide-dot" | "324" | "diamond-wide-open-dot" | "25" | "hourglass" | "125" | "hourglass-open" | "26" | "bowtie" | "126" | "bowtie-open" | "27" | "circle-cross" | "127" | "circle-cross-open" | "28" | "circle-x" | "128" | "circle-x-open" | "29" | "square-cross" | "129" | "square-cross-open" | "30" | "square-x" | "130" | "square-x-open" | "31" | "diamond-cross" | "131" | "diamond-cross-open" | "32" | "diamond-x" | "132" | "diamond-x-open" | "33" | "cross-thin" | "133" | "cross-thin-open" | "34" | "x-thin" | "134" | "x-thin-open" | "35" | "asterisk" | "135" | "asterisk-open" | "36" | "hash" | "136" | "hash-open" | "236" | "hash-dot" | "336" | "hash-open-dot" | "37" | "y-up" | "137" | "y-up-open" | "38" | "y-down" | "138" | "y-down-open" | "39" | "y-left" | "139" | "y-left-open" | "40" | "y-right" | "140" | "y-right-open" | "41" | "line-ew" | "141" | "line-ew-open" | "42" | "line-ns" | "142" | "line-ns-open" | "43" | "line-ne" | "143" | "line-ne-open" | "44" | "line-nw" | "144" | "line-nw-open")
default: "circle"
Sets the marker symbol type. Adding 100 is equivalent to appending "-open" to a symbol name. Adding 200 is equivalent to appending "-dot" to a symbol name. Adding 300 is equivalent to appending "-open-dot" or "dot-open" to a symbol name.

- opacity (number between or equal to 0 and 1)
  Sets the marker opacity.
- colorsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for color . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's color data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- sizesrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for size . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's size data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- symbolsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for symbol . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's symbol data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- opacitysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for opacity . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's opacity data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- rsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for r . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's r data. It's useful

if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- tsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for t . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's t data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

area

- visible
- showlegend
- legendgroup
- opacity
- name
- hoverinfo
- stream
  - token
  - maxpoints
- r
- t
- marker
  - color
  - size
  - symbol
  - opacity
  - colorsrc
  - sizesrc
  - symbolsrc
  - opacitysrc
- rsrc
- tsrc

画面布局

- font
  Sets the global font. Note that fonts used in traces and other layout components inherit from the global font.
  - family (`string`)
    default: `""Open Sans", verdana, arial, sans-serif"`
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (`number greater than or equal to 1`)
    default: `12`
  - color (`color`)
    default: `"#444"`
- title (`string`)
  default: `"Click to enter Plot title"`
  Sets the plot's title.
- titlefont
  Sets the title font.

- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)

- autosize (boolean)
  Determines whether or not a layout width or height that has been left undefined by the user is initialized on each relayout. Note that, regardless of this attribute, an undefined layout width or height is always initialized on the first call to plot.
- width (number greater than or equal to 10)
  default: 700
  Sets the plot's width (in px).
- height (number greater than or equal to 10)
  default: 450
  Sets the plot's height (in px).
- margin
  - l (number greater than or equal to 0)
    default: 80
    Sets the left margin (in px).
  - r (number greater than or equal to 0)
    default: 80
    Sets the right margin (in px).
  - t (number greater than or equal to 0)
    default: 100
    Sets the top margin (in px).
  - b (number greater than or equal to 0)
    default: 80
    Sets the bottom margin (in px).
  - pad (number greater than or equal to 0)
    default: 0
    Sets the amount of padding (in px) between the plotting area and the axis lines
  - autoexpand (boolean)
    default: True
- paper_bgcolor (color)
  default: "#fff"
  Sets the color of paper where the graph is drawn.
- plot_bgcolor (color)
  default: "#fff"
  Sets the color of plotting area in-between x and y axes.
- separators (string)
  default: ".,"
  Sets the decimal and thousand separators. For example, ". " puts a '.' before decimals and a space between thousands.
- hidesources (boolean)
  Determines whether or not a text link citing the data source is placed at the bottom-right cored of the figure. Has only an effect only on graphs that have been generated via forked graphs from the plotly service (at https://plot.ly or on-premise).
- showlegend (boolean)
  Determines whether or not a legend is drawn.
- dragmode (enumerated: "zoom" | "pan" | "select" | "lasso" | "orbit" | "turntable")
  default: "zoom"
  Determines the mode of drag interactions. "select" and "lasso" apply only to scatter traces with markers or text. "orbit" and "turntable" apply only to 3D scenes.
- hovermode (enumerated: "x" | "y" | "closest" | False)
  Determines the mode of hover interactions.
- xaxis

- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- title (string)
  Sets the title of this axis.
- titlefont
  Sets this axis' title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- type (enumerated: "-" | "linear" | "log" | "date" | "category")
  default: "-"
  Sets the axis type. By default, plotly attempts to determined the axis type by looking into the data of the traces that referenced the axis in question.
- autorange (enumerated: True | False | "reversed")
  default: True
  Determines whether or not the range of this axis is computed in relation to the input data. See `rangemode` for more info. If `range` is provided, then `autorange` is set to "False".
- rangemode (enumerated: "normal" | "tozero" | "nonnegative")
  default: "normal"
  If "normal", the range is computed in relation to the extrema of the input data. If "tozero"`, the range extends to 0, regardless of the input data If "nonnegative", the range is non-negative, regardless of the input data.
- range (list)
  Sets the range of this axis. If the axis `type` is "log", then you must take the log of your desired range (e.g. to set the range from 1 to 100, set the range from 0 to 2). If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
  Each dict has one or more of the keys listed below.
- fixedrange (boolean)
  Determines whether or not this axis is zoom-able. If True, then zoom is disabled.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n`

must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"

- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- mirror (enumerated: True | "ticks" | False | "all" | "allticks")
  Determines if the axis lines or/and ticks are mirrored to the opposite side of the plotting area. If "True", the axis lines are mirrored. If "ticks", the axis lines and ticks are mirrored. If "False", mirroring is disable. If "all", axis lines are mirrored on all shared-axes subplots. If "allticks", axis lines and ticks are mirrored on all shared-axes subplots.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it

appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.

- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (boolean)
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "#eee"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- zeroline (boolean)
  Determines whether or not a line is drawn at along the 0 value of this axis. If "True", the zero line is drawn on top of the grid lines.
- zerolinecolor (color)
  default: "#444"
  Sets the line color of the zero line.
- zerolinewidth (number)
  default: 1
  Sets the width (in px) of the zero line.
- anchor (enumerated: "free" | "/^x([2-9]|[1-9][0-9]+)?$/" | "/^y([2-9]|[1-9][0-9]+)?$/")
  If set to an opposite-letter axis id (e.g. `xaxis2`, `yaxis`), this axis is bound to the corresponding opposite-letter axis. If set to "free", this axis' position is determined by `position`.
- side (enumerated: "top" | "bottom" | "left" | "right")
  Determines whether a x (y) axis is positioned at the "bottom" ("left") or "top" ("right") of the plotting area.
- overlaying (enumerated: "free" | "/^x([2-9]|[1-9][0-9]+)?$/" | "/^y([2-9]|[1-9][0-9]+)?$/")
  If set a same-letter axis id, this axis is overlaid on top of the corresponding same-letter axis. If "False", this axis does not overlay any same-letter axes.
- domain (list)
  default: [0, 1]
  Sets the domain of this axis (in plot fraction).
  Each dict has one or more of the keys listed below.
- position (number between or equal to 0 and 1)
  default: 0
  Sets the position of this axis in the plotting space (in normalized coordinates). Only has an effect if `anchor` is set to "free".
- categoryorder (enumerated: "trace" | "category ascending" | "category descending" | "array")
  default: "trace"
  Specifies the ordering logic for the case of categorical variables. By default, plotly uses "trace", which specifies the

order that is present in the data supplied. Set `categoryorder` to "category ascending" or "category descending" if order should be determined by the alphanumerical order of the category names. Set `categoryorder` to "array" to derive the ordering from the attribute `categoryarray`. If a category is not found in the `categoryarray` array, the sorting behavior for that attribute will be identical to the "trace" mode. The unspecified categories will follow the categories in `categoryarray`.

- categoryarray (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the order in which categories on this axis appear. Only has an effect if `categoryorder` is set to "array". Used with `categoryorder`.
- rangeslider
  - bgcolor (`color`)
    default: `"#fff"`
    Sets the background color of the range slider.
  - bordercolor (`color`)
    default: `"#444"`
    Sets the border color of the range slider.
  - borderwidth (`integer greater than or equal to 0`)
    default: `0`
    Sets the border color of the range slider.
  - range (`list`)
    Sets the range of the range slider. If not set, defaults to the full xaxis range. If the axis `type` is "log", then you must take the log of your desired range. If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
    Each dict has one or more of the keys listed below.
  - thickness (`number between or equal to 0 and 1`)
    default: `0.15`
    The height of the range slider as a fraction of the total plot area height.
  - visible (`boolean`)
    default: `True`
    Determines whether or not the range slider will be visible. If visible, perpendicular axes will be set to `fixedrange`
- rangeselector
  - visible (`boolean`)
    Determines whether or not this range selector is visible. Note that range selectors are only available for x axes of `type` set to or auto-typed to "date".
  - buttons
    Each dict has one or more of the keys listed below.
  - step (`enumerated: "month" | "year" | "day" | "hour" | "minute" | "second" | "all"`)
    default: `"month"`
    The unit of measurement that the `count` value will set the range by.
  - stepmode (`enumerated: "backward" | "todate"`)
    default: `"backward"`
    Sets the range update mode. If "backward", the range update shifts the start of range back "count" times "step" milliseconds. If "todate", the range update shifts the start of range back to the first timestamp from "count" times "step" milliseconds back. For example, with `step` set to "year" and `count` set to "1" the range update shifts the start of the range back to January 01 of the current year. Month and year "todate" are currently available only for the built-in (Gregorian) calendar.
  - count (`number greater than or equal to 0`)
    default: `1`
    Sets the number of steps to take to update the range. Use with `step` to specify the update interval.
  - label (`string`)
    Sets the text label to appear on the button.
  - x (`number between or equal to -2 and 3`)
    Sets the x position (in normalized coordinates) of the range selector.
  - xanchor (`enumerated: "auto" | "left" | "center" | "right"`)
    default: `"left"`
    Sets the range selector's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the range selector.
  - y (`number between or equal to -2 and 3`)
    Sets the y position (in normalized coordinates) of the range selector.
  - yanchor (`enumerated: "auto" | "top" | "middle" | "bottom"`)
    default: `"bottom"`

Sets the range selector's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the range selector.

- font
  Sets the font of the range selector button text.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- bgcolor (color)
  default: "#eee"
  Sets the background color of the range selector buttons.
- activecolor (color)
  Sets the background color of the active range selector button.
- bordercolor (color)
  default: "#444"
  Sets the color of the border enclosing the range selector.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) of the border enclosing the range selector.
  - calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
    default: "gregorian"
    Sets the calendar system to use for `range` and `tick0` if this is a date axis. This does not set the calendar for interpreting data on this axis, that's specified in the trace or via the global `layout.calendar`
  - tickvalssrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - ticktextsrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - categoryarraysrc (source string | plotly.grid objs.Column)
    Sets the source reference on plot.ly for categoryarray . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's categoryarray data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- yaxis
  - color (color)
    default: "#444"
    Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
  - title (string)
    Sets the title of this axis.
  - titlefont
    Sets this axis' title font.
    - family (string)
      HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
    - size (number greater than or equal to 1)

- color (color)
  - type (enumerated: "-" | "linear" | "log" | "date" | "category")
    default: "-"
    Sets the axis type. By default, plotly attempts to determined the axis type by looking into the data of the traces that referenced the axis in question.
  - autorange (enumerated: True | False | "reversed")
    default: True
    Determines whether or not the range of this axis is computed in relation to the input data. See `rangemode` for more info. If `range` is provided, then `autorange` is set to "False".
  - rangemode (enumerated: "normal" | "tozero" | "nonnegative")
    default: "normal"
    If "normal", the range is computed in relation to the extrema of the input data. If "tozero"`, the range extends to 0, regardless of the input data If "nonnegative", the range is non-negative, regardless of the input data.
  - range (list)
    Sets the range of this axis. If the axis `type` is "log", then you must take the log of your desired range (e.g. to set the range from 1 to 100, set the range from 0 to 2). If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
    Each dict has one or more of the keys listed below.
  - fixedrange (boolean)
    Determines whether or not this axis is zoom-able. If True, then zoom is disabled.
  - tickmode (enumerated: "auto" | "linear" | "array")
    Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
  - nticks (integer greater than or equal to 0)
    default: 0
    Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
  - tick0 (number or categorical coordinate string)
    Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
  - dtick (number or categorical coordinate string)
    Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
  - tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
  - ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
    Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
  - ticks (enumerated: "outside" | "inside" | "")
    Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
  - mirror (enumerated: True | "ticks" | False | "all" | "allticks")
    Determines if the axis lines or/and ticks are mirrored to the opposite side of the plotting area. If "True", the axis lines are mirrored. If "ticks", the axis lines and ticks are mirrored. If "False", mirroring is disable. If "all", axis lines are mirrored on all shared-axes subplots. If "allticks", axis lines and ticks are mirrored on all shared-axes subplots.

- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date

formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"

- showline (boolean)
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "#eee"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- zeroline (boolean)
  Determines whether or not a line is drawn at along the 0 value of this axis. If "True", the zero line is drawn on top of the grid lines.
- zerolinecolor (color)
  default: "#444"
  Sets the line color of the zero line.
- zerolinewidth (number)
  default: 1
  Sets the width (in px) of the zero line.
- anchor (enumerated: "free" | "/^x([2-9]|[1-9][0-9]+)?$/" | "/^y([2-9]|[1-9][0-9]+)?$/")
  If set to an opposite-letter axis id (e.g. `xaxis2`, `yaxis`), this axis is bound to the corresponding opposite-letter axis. If set to "free", this axis' position is determined by `position`.
- side (enumerated: "top" | "bottom" | "left" | "right")
  Determines whether a x (y) axis is positioned at the "bottom" ("left") or "top" ("right") of the plotting area.
- overlaying (enumerated: "free" | "/^x([2-9]|[1-9][0-9]+)?$/" | "/^y([2-9]|[1-9][0-9]+)?$/")
  If set a same-letter axis id, this axis is overlaid on top of the corresponding same-letter axis. If "False", this axis does not overlay any same-letter axes.
- domain (list)
  default: [0, 1]
  Sets the domain of this axis (in plot fraction).
  Each dict has one or more of the keys listed below.
- position (number between or equal to 0 and 1)
  default: 0
  Sets the position of this axis in the plotting space (in normalized coordinates). Only has an effect if `anchor` is set to "free".
- categoryorder (enumerated: "trace" | "category ascending" | "category descending" | "array")
  default: "trace"
  Specifies the ordering logic for the case of categorical variables. By default, plotly uses "trace", which specifies the order that is present in the data supplied. Set `categoryorder` to "category ascending" or "category descending" if order should be determined by the alphanumerical order of the category names. Set `categoryorder` to "array" to derive the ordering from the attribute `categoryarray`. If a category is not found in the `categoryarray` array, the sorting behavior for that attribute will be identical to the "trace" mode. The unspecified categories will follow the categories in `categoryarray`.
- categoryarray (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the order in which categories on this axis appear. Only has an effect if `categoryorder` is set to "array". Used with `categoryorder`.
- calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use for `range` and `tick0` if this is a date axis. This does not set the calendar for interpreting data on this axis, that's specified in the trace or via the global `layout.calendar`

- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- categoryarraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for categoryarray . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's categoryarray data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- ternary
- domain
  - x (list)
    default: [0, 1]
    Sets the horizontal domain of this subplot (in plot fraction).
    Each dict has one or more of the keys listed below.
  - y (list)
    default: [0, 1]
    Sets the vertical domain of this subplot (in plot fraction).
    Each dict has one or more of the keys listed below.
- bgcolor (color)
  default: "#fff"
  Set the background color of the subplot
- sum (number greater than or equal to 0)
  default: 1
  The number each triplet should sum to, and the maximum range of each axis
- aaxis
  - title (string)
    Sets the title of this axis.
  - titlefont
    Sets this axis' title font.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
  - color (color)
    default: "#444"
    Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
  - tickmode (enumerated: "auto" | "linear" | "array")
    Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
  - nticks (integer greater than or equal to 1)
    default: 6
    Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
  - tick0 (number or categorical coordinate string)
    Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is

"category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.

- dtick (`number or categorical coordinate string`)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showexponent (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (`enumerated: "none" | "e" | "E" | "power" | "SI" | "B"`)
  default: `"B"`
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas,

to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".

- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (`string`)
  default: `""`
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (`string`)
  default: `""`
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (`boolean`)
  default: `True`
  Determines whether or not a line bounding this axis is drawn.
- linecolor (`color`)
  default: `"#444"`
  Sets the axis line color.
- linewidth (`number greater than or equal to 0`)
  default: `1`
  Sets the width (in px) of the axis line.
- showgrid (`boolean`)
  default: `True`
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (`color`)
  default: `"#eee"`
  Sets the color of the grid lines.
- gridwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the width (in px) of the grid lines.
- tickvalssrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for tickvals . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (`source string | plotly.grid objs.Column`)
  Sets the source reference on plot.ly for ticktext . A `source string` refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- baxis
- title (`string`)
  Sets the title of this axis.
- titlefont
  Sets this axis' title font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono",

"Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".

- size (number greater than or equal to 1)
- color (color)
- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 1)
  default: 6
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.

- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (boolean)
  default: True
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  default: True
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "#eee"
  Sets the color of the grid lines.

- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
  - caxis
- title (string)
  Sets the title of this axis.
- titlefont
  Sets this axis' title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 1)
  default: 6
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.

- ticks (enumerated: "outside" | "inside" | "")
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""

Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"

- showline (boolean)
  default: True
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  default: True
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "#eee"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- scene
  - bgcolor (color)
    default: "rgba(0,0,0,0)"
  - camera
    - up
      Sets the (x,y,z) components of the 'up' camera vector. This vector determines the up direction of this scene with respect to the page. The default is "{x: 0, y: 0, z: 1}" which means that the z axis points up.
      - x (number)
        default: 0
      - y (number)
        default: 0
      - z (number)
        default: 1
    - center
      Sets the (x,y,z) components of the 'center' camera vector This vector determines the translation (x,y,z) space about the center of this scene. By default, there is no such translation.
      - x (number)
        default: 0
      - y (number)
        default: 0
      - z (number)
        default: 0
    - eye
      Sets the (x,y,z) components of the 'eye' camera vector. This vector determines the view point about the origin of this scene.
      - x (number)
        default: 1.25
      - y (number)
        default: 1.25

- z (number)
  default: 1.25
- domain
- x (list)
  default: [0, 1]
  Sets the horizontal domain of this scene (in plot fraction).
  Each dict has one or more of the keys listed below.
- y (list)
  default: [0, 1]
  Sets the vertical domain of this scene (in plot fraction).
  Each dict has one or more of the keys listed below.
- aspectmode (enumerated: "auto" | "cube" | "data" | "manual")
  default: "auto"
  If "cube", this scene's axes are drawn as a cube, regardless of the axes' ranges. If "data", this scene's axes are drawn in proportion with the axes' ranges. If "manual", this scene's axes are drawn in proportion with the input of "aspectratio" (the default behavior if "aspectratio" is provided). If "auto", this scene's axes are drawn using the results of "data" except when one axis is more than four times the size of the two others, where in that case the results of "cube" are used.
- aspectratio
  Sets this scene's axis aspectratio.
- x (number greater than or equal to 0)
- y (number greater than or equal to 0)
- z (number greater than or equal to 0)
- xaxis
- showspikes (boolean)
  default: True
  Sets whether or not spikes starting from data points to this axis' wall are shown on hover.
- spikesides (boolean)
  default: True
  Sets whether or not spikes extending from the projection data points to this axis' wall boundaries are shown on hover.
- spikethickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the spikes.
- spikecolor (color)
  default: "#444"
  Sets the color of the spikes.
- showbackground (boolean)
  Sets whether or not this axis' wall has a background color.
- backgroundcolor (color)
  default: "rgba(204, 204, 204, 0.5)"
  Sets the background color of this axis' wall.
- showaxeslabels (boolean)
  default: True
  Sets whether or not this axis is labeled
- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- categoryorder (enumerated: "trace" | "category ascending" | "category descending" | "array")
  default: "trace"
  Specifies the ordering logic for the case of categorical variables. By default, plotly uses "trace", which specifies the order that is present in the data supplied. Set `categoryorder` to "category ascending" or "category descending" if order should be determined by the alphanumerical order of the category names. Set `categoryorder` to "array" to derive the ordering from the attribute `categoryarray`. If a category is not found in the `categoryarray` array, the sorting behavior for that attribute will be identical to the "trace" mode. The unspecified categories will follow the categories in `categoryarray`.
- categoryarray (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the order in which categories on this axis appear. Only has an effect if `categoryorder` is set to "array". Used with `categoryorder`.
- title (string)
  Sets the title of this axis.

- titlefont
  Sets this axis' title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- type (enumerated: "-" | "linear" | "log" | "date" | "category")
  default: "-"
  Sets the axis type. By default, plotly attempts to determined the axis type by looking into the data of the traces that referenced the axis in question.
- autorange (enumerated: True | False | "reversed")
  default: True
  Determines whether or not the range of this axis is computed in relation to the input data. See `rangemode` for more info. If `range` is provided, then `autorange` is set to "False".
- rangemode (enumerated: "normal" | "tozero" | "nonnegative")
  default: "normal"
  If "normal", the range is computed in relation to the extrema of the input data. If "tozero"`, the range extends to 0, regardless of the input data If "nonnegative", the range is non-negative, regardless of the input data.
- range (list)
  Sets the range of this axis. If the axis `type` is "log", then you must take the log of your desired range (e.g. to set the range from 1 to 100, set the range from 0 to 2). If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
  Each dict has one or more of the keys listed below.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n"dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.

- ticks (enumerated: "outside" | "inside" | "")
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- mirror (enumerated: True | "ticks" | False | "all" | "allticks")
  Determines if the axis lines or/and ticks are mirrored to the opposite side of the plotting area. If "True", the axis lines are mirrored. If "ticks", the axis lines and ticks are mirrored. If "False", mirroring is disable. If "all", axis lines are mirrored on all shared-axes subplots. If "allticks", axis lines and ticks are mirrored on all shared-axes subplots.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see:

https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"

- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (boolean)
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "rgb(204, 204, 204)"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- zeroline (boolean)
  Determines whether or not a line is drawn at along the 0 value of this axis. If "True", the zero line is drawn on top of the grid lines.
- zerolinecolor (color)
  default: "#444"
  Sets the line color of the zero line.
- zerolinewidth (number)
  default: 1
  Sets the width (in px) of the zero line.
- calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use for `range` and `tick0` if this is a date axis. This does not set the calendar for interpreting data on this axis, that's specified in the trace or via the global `layout.calendar`
- categoryarraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for categoryarray . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's categoryarray data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.

- yaxis
- showspikes (boolean)
  default: True
  Sets whether or not spikes starting from data points to this axis' wall are shown on hover.
- spikesides (boolean)
  default: True
  Sets whether or not spikes extending from the projection data points to this axis' wall boundaries are shown on hover.

- spikethickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the spikes.
- spikecolor (color)
  default: "#444"
  Sets the color of the spikes.
- showbackground (boolean)
  Sets whether or not this axis' wall has a background color.
- backgroundcolor (color)
  default: "rgba(204, 204, 204, 0.5)"
  Sets the background color of this axis' wall.
- showaxeslabels (boolean)
  default: True
  Sets whether or not this axis is labeled
- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- categoryorder (enumerated: "trace" | "category ascending" | "category descending" | "array")
  default: "trace"
  Specifies the ordering logic for the case of categorical variables. By default, plotly uses "trace", which specifies the order that is present in the data supplied. Set `categoryorder` to "category ascending" or "category descending" if order should be determined by the alphanumerical order of the category names. Set `categoryorder` to "array" to derive the ordering from the attribute `categoryarray`. If a category is not found in the `categoryarray` array, the sorting behavior for that attribute will be identical to the "trace" mode. The unspecified categories will follow the categories in `categoryarray`.
- categoryarray (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the order in which categories on this axis appear. Only has an effect if `categoryorder` is set to "array". Used with `categoryorder`.
- title (string)
  Sets the title of this axis.
- titlefont
  Sets this axis' title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- type (enumerated: "-" | "linear" | "log" | "date" | "category")
  default: "-"
  Sets the axis type. By default, plotly attempts to determined the axis type by looking into the data of the traces that referenced the axis in question.
- autorange (enumerated: True | False | "reversed")
  default: True
  Determines whether or not the range of this axis is computed in relation to the input data. See `rangemode` for more info. If `range` is provided, then `autorange` is set to "False".
- rangemode (enumerated: "normal" | "tozero" | "nonnegative")
  default: "normal"
  If "normal", the range is computed in relation to the extrema of the input data. If "tozero"`, the range extends to 0, regardless of the input data If "nonnegative", the range is non-negative, regardless of the input data.
- range (list)
  Sets the range of this axis. If the axis `type` is "log", then you must take the log of your desired range (e.g. to set the range from 1 to 100, set the range from 0 to 2). If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
  Each dict has one or more of the keys listed below.

- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.
- ticktext (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (enumerated: "outside" | "inside" | "")
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- mirror (enumerated: True | "ticks" | False | "all" | "allticks")
  Determines if the axis lines or/and ticks are mirrored to the opposite side of the plotting area. If "True", the axis lines are mirrored. If "ticks", the axis lines and ticks are mirrored. If "False", mirroring is disable. If "all", axis lines are mirrored on all shared-axes subplots. If "allticks", axis lines and ticks are mirrored on all shared-axes subplots.
- ticklen (number greater than or equal to 0)
  default: 5
  Sets the tick length (in px).
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- tickcolor (color)
  default: "#444"
  Sets the tick color.
- showticklabels (boolean)
  default: True
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)

- tickangle (angle)
  default: auto
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickprefix (string)
  default: ""
  Sets a tick label prefix.
- showtickprefix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (string)
  default: ""
  Sets a tick label suffix.
- showticksuffix (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  Same as `showtickprefix` but for tick suffixes.
- showexponent (enumerated: "all" | "first" | "last" | "none")
  default: "all"
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (enumerated: "none" | "e" | "E" | "power" | "SI" | "B")
  default: "B"
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (boolean)
  If "True", even 4-digit integers are separated
- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (boolean)
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "rgb(204, 204, 204)"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- zeroline (boolean)
  Determines whether or not a line is drawn at along the 0 value of this axis. If "True", the zero line is drawn on top of the grid lines.
- zerolinecolor (color)
  default: "#444"
  Sets the line color of the zero line.

- zerolinewidth (number)
  default: 1
  Sets the width (in px) of the zero line.
- calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use for `range` and `tick0` if this is a date axis. This does not set the calendar for interpreting data on this axis, that's specified in the trace or via the global `layout.calendar`
- categoryarraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for categoryarray . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's categoryarray data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- zaxis
- showspikes (boolean)
  default: True
  Sets whether or not spikes starting from data points to this axis' wall are shown on hover.
- spikesides (boolean)
  default: True
  Sets whether or not spikes extending from the projection data points to this axis' wall boundaries are shown on hover.
- spikethickness (number greater than or equal to 0)
  default: 2
  Sets the thickness (in px) of the spikes.
- spikecolor (color)
  default: "#444"
  Sets the color of the spikes.
- showbackground (boolean)
  Sets whether or not this axis' wall has a background color.
- backgroundcolor (color)
  default: "rgba(204, 204, 204, 0.5)"
  Sets the background color of this axis' wall.
- showaxeslabels (boolean)
  default: True
  Sets whether or not this axis is labeled
- color (color)
  default: "#444"
  Sets default for all colors associated with this axis all at once: line, font, tick, and grid colors. Grid color is lightened by blending this with the plot background Individual pieces can override this.
- categoryorder (enumerated: "trace" | "category ascending" | "category descending" | "array")
  default: "trace"
  Specifies the ordering logic for the case of categorical variables. By default, plotly uses "trace", which specifies the order that is present in the data supplied. Set `categoryorder` to "category ascending" or "category descending" if order should be determined by the alphanumerical order of the category names. Set `categoryorder` to "array" to derive the ordering from the attribute `categoryarray`. If a category is not found in the `categoryarray` array, the sorting behavior for that attribute will be identical to the "trace" mode. The unspecified categories will follow the categories in `categoryarray`.
- categoryarray (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the order in which categories on this axis appear. Only has an effect if `categoryorder` is set to "array". Used with `categoryorder`.

- title (string)
  Sets the title of this axis.
- titlefont
  Sets this axis' title font.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- type (enumerated: "-" | "linear" | "log" | "date" | "category")
  default: "-"
  Sets the axis type. By default, plotly attempts to determined the axis type by looking into the data of the traces that referenced the axis in question.
- autorange (enumerated: True | False | "reversed")
  default: True
  Determines whether or not the range of this axis is computed in relation to the input data. See `rangemode` for more info. If `range` is provided, then `autorange` is set to "False".
- rangemode (enumerated: "normal" | "tozero" | "nonnegative")
  default: "normal"
  If "normal", the range is computed in relation to the extrema of the input data. If "tozero"`, the range extends to 0, regardless of the input data If "nonnegative", the range is non-negative, regardless of the input data.
- range (list)
  Sets the range of this axis. If the axis `type` is "log", then you must take the log of your desired range (e.g. to set the range from 1 to 100, set the range from 0 to 2). If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
  Each dict has one or more of the keys listed below.
- tickmode (enumerated: "auto" | "linear" | "array")
  Sets the tick mode for this axis. If "auto", the number of ticks is set via `nticks`. If "linear", the placement of the ticks is determined by a starting position `tick0` and a tick step `dtick` ("linear" is the default value if `tick0` and `dtick` are provided). If "array", the placement of the ticks is set via `tickvals` and the tick text is `ticktext`. ("array" is the default value if `tickvals` is provided).
- nticks (integer greater than or equal to 0)
  default: 0
  Specifies the maximum number of ticks for the particular axis. The actual number of ticks will be chosen automatically to be less than or equal to `nticks`. Has an effect only if `tickmode` is set to "auto".
- tick0 (number or categorical coordinate string)
  Sets the placement of the first tick on this axis. Use with `dtick`. If the axis `type` is "log", then you must take the log of your starting tick (e.g. to set the starting tick to 100, set the `tick0` to 2) except when `dtick`="L<f>" (see `dtick` for more info). If the axis `type` is "date", it should be a date string, like date data. If the axis `type` is "category", it should be a number, using the scale where each category is assigned a serial number from zero in the order it appears.
- dtick (number or categorical coordinate string)
  Sets the step in-between ticks on this axis. Use with `tick0`. Must be a positive number, or special strings available to "log" and "date" axes. If the axis `type` is "log", then ticks are set every 10^(n*dtick) where n is the tick number. For example, to set a tick mark at 1, 10, 100, 1000, ... set dtick to 1. To set tick marks at 1, 100, 10000, ... set dtick to 2. To set tick marks at 1, 5, 25, 125, 625, 3125, ... set dtick to log_10(5), or 0.69897000433. "log" has several special values; "L<f>", where `f` is a positive number, gives ticks linearly spaced in value (but not position). For example `tick0` = 0.1, `dtick` = "L0.5" will put ticks at 0.1, 0.6, 1.1, 1.6 etc. To show powers of 10 plus small digits between, use "D1" (all digits) or "D2" (only 2 and 5). `tick0` is ignored for "D1" and "D2". If the axis `type` is "date", then you must convert the time to milliseconds. For example, to set the interval between ticks to one day, set `dtick` to 86400000.0. "date" also has special values "M<n>" gives ticks spaced by a number of months. `n` must be a positive integer. To set ticks on the 15th of every third month, set `tick0` to "2000-01-15" and `dtick` to "M3". To set ticks every 4 years, set `dtick` to "M48"
- tickvals (list, numpy array, or Pandas series of numbers, strings, or datetimes.)
  Sets the values at which ticks on this axis appear. Only has an effect if `tickmode` is set to "array". Used with `ticktext`.

- ticktext (`list, numpy array, or Pandas series of numbers, strings, or datetimes.`)
  Sets the text displayed at the ticks position via `tickvals`. Only has an effect if `tickmode` is set to "array". Used with `tickvals`.
- ticks (`enumerated: "outside" | "inside" | ""`)
  Determines whether ticks are drawn or not. If "", this axis' ticks are not drawn. If "outside" ("inside"), this axis' are drawn outside (inside) the axis lines.
- mirror (`enumerated: True | "ticks" | False | "all" | "allticks"`)
  Determines if the axis lines or/and ticks are mirrored to the opposite side of the plotting area. If "True", the axis lines are mirrored. If "ticks", the axis lines and ticks are mirrored. If "False", mirroring is disable. If "all", axis lines are mirrored on all shared-axes subplots. If "allticks", axis lines and ticks are mirrored on all shared-axes subplots.
- ticklen (`number greater than or equal to 0`)
  default: `5`
  Sets the tick length (in px).
- tickwidth (`number greater than or equal to 0`)
  default: `1`
  Sets the tick width (in px).
- tickcolor (`color`)
  default: `"#444"`
  Sets the tick color.
- showticklabels (`boolean`)
  default: `True`
  Determines whether or not the tick labels are drawn.
- tickfont
  Sets the tick font.
- family (`string`)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (`number greater than or equal to 1`)
- color (`color`)
- tickangle (`angle`)
  default: `auto`
  Sets the angle of the tick labels with respect to the horizontal. For example, a `tickangle` of -90 draws the tick labels vertically.
- tickprefix (`string`)
  default: `""`
  Sets a tick label prefix.
- showtickprefix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all tick labels are displayed with a prefix. If "first", only the first tick is displayed with a prefix. If "last", only the last tick is displayed with a suffix. If "none", tick prefixes are hidden.
- ticksuffix (`string`)
  default: `""`
  Sets a tick label suffix.
- showticksuffix (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  Same as `showtickprefix` but for tick suffixes.
- showexponent (`enumerated: "all" | "first" | "last" | "none"`)
  default: `"all"`
  If "all", all exponents are shown besides their significands. If "first", only the exponent of the first tick is shown. If "last", only the exponent of the last tick is shown. If "none", no exponents appear.
- exponentformat (`enumerated: "none" | "e" | "E" | "power" | "SI" | "B"`)
  default: `"B"`
  Determines a formatting rule for the tick exponents. For example, consider the number 1,000,000,000. If "none", it appears as 1,000,000,000. If "e", 1e+9. If "E", 1E+9. If "power", 1x10^9 (with 9 in a super script). If "SI", 1G. If "B", 1B.
- separatethousands (`boolean`)
  If "True", even 4-digit integers are separated

- tickformat (string)
  default: ""
  Sets the tick label formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- hoverformat (string)
  default: ""
  Sets the hover text formatting rule using d3 formatting mini-languages which are very similar to those in Python. For numbers, see: https://github.com/d3/d3-format/blob/master/README.md#locale_format And for dates see: https://github.com/d3/d3-time-format/blob/master/README.md#locale_format We add one item to d3's date formatter: "%{n}f" for fractional seconds with n digits. For example, "2016-10-13 09:15:23.456" with tickformat "%H~%M~%S.%2f" would display "09~15~23.46"
- showline (boolean)
  Determines whether or not a line bounding this axis is drawn.
- linecolor (color)
  default: "#444"
  Sets the axis line color.
- linewidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the axis line.
- showgrid (boolean)
  Determines whether or not grid lines are drawn. If "True", the grid lines are drawn at every tick mark.
- gridcolor (color)
  default: "rgb(204, 204, 204)"
  Sets the color of the grid lines.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the grid lines.
- zeroline (boolean)
  Determines whether or not a line is drawn at along the 0 value of this axis. If "True", the zero line is drawn on top of the grid lines.
- zerolinecolor (color)
  default: "#444"
  Sets the line color of the zero line.
- zerolinewidth (number)
  default: 1
  Sets the width (in px) of the zero line.
- calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the calendar system to use for `range` and `tick0` if this is a date axis. This does not set the calendar for interpreting data on this axis, that's specified in the trace or via the global `layout.calendar`
- categoryarraysrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for categoryarray . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's categoryarray data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- tickvalssrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for tickvals . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's tickvals data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- ticktextsrc (source string | plotly.grid objs.Column)
  Sets the source reference on plot.ly for ticktext . A source string refers to a unique identifier that is assigned to a plotly grid. Assigning data to a graph with a source string is an alternative way to set this graph's ticktext data. It's useful if you want to keep your data separate from your graphs, create several graphs from a single dataset, or edit your graph's data after creation. Learn more.
- dragmode (enumerated: "orbit" | "turntable" | "zoom" | "pan")
  default: "turntable"
  Determines the mode of drag interactions for this scene.

- hovermode (enumerated: "closest" | False)
  default: "closest"
  Determines the mode of hover interactions for this scene.
- geo
- domain
  - x (list)
    default: [0, 1]
    Sets the horizontal domain of this map (in plot fraction).
    Each dict has one or more of the keys listed below.
  - y (list)
    default: [0, 1]
    Sets the vertical domain of this map (in plot fraction).
    Each dict has one or more of the keys listed below.
- resolution (enumerated: "110" | "50")
  default: "110"
  Sets the resolution of the base layers. The values have units of km/mm e.g. 110 corresponds to a scale ratio of 1:110,000,000.
- scope (enumerated: "world" | "usa" | "europe" | "asia" | "africa" | "north america" | "south america")
  default: "world"
  Set the scope of the map.
- projection
  - type (enumerated: "equirectangular" | "mercator" | "orthographic" | "natural earth" | "kavrayskiy7" | "miller" | "robinson" | "eckert4" | "azimuthal equal area" | "azimuthal equidistant" | "conic equal area" | "conic conformal" | "conic equidistant" | "gnomonic" | "stereographic" | "mollweide" | "hammer" | "transverse mercator" | "albers usa" | "winkel tripel")
    Sets the projection type.
  - rotation
  - lon (number)
    Rotates the map along parallels (in degrees East).
  - lat (number)
    Rotates the map along meridians (in degrees North).
  - roll (number)
    Roll the map (in degrees) For example, a roll of "180" makes the map appear upside down.
  - parallels (list)
    For conic projection types only. Sets the parallels (tangent, secant) where the cone intersects the sphere.
    Each dict has one or more of the keys listed below.
  - scale (number between or equal to 0 and 10)
    default: 1
    Zooms in or out on the map view.
- showcoastlines (boolean)
  Sets whether or not the coastlines are drawn.
- coastlinecolor (color)
  default: "#444"
  Sets the coastline color.
- coastlinewidth (number greater than or equal to 0)
  default: 1
  Sets the coastline stroke width (in px).
- showland (boolean)
  Sets whether or not land masses are filled in color.
- landcolor (color)
  default: "#F0DC82"
  Sets the land mass color.
- showocean (boolean)
  Sets whether or not oceans are filled in color.
- oceancolor (color)
  default: "#3399FF"
  Sets the ocean color
- showlakes (boolean)
  Sets whether or not lakes are drawn.

- lakecolor (color)
  default: "#3399FF"
  Sets the color of the lakes.
- showrivers (boolean)
  Sets whether or not rivers are drawn.
- rivercolor (color)
  default: "#3399FF"
  Sets color of the rivers.
- riverwidth (number greater than or equal to 0)
  default: 1
  Sets the stroke width (in px) of the rivers.
- showcountries (boolean)
  Sets whether or not country boundaries are drawn.
- countrycolor (color)
  default: "#444"
  Sets line color of the country boundaries.
- countrywidth (number greater than or equal to 0)
  default: 1
  Sets line width (in px) of the country boundaries.
- showsubunits (boolean)
  Sets whether or not boundaries of subunits within countries (e.g. states, provinces) are drawn.
- subunitcolor (color)
  default: "#444"
  Sets the color of the subunits boundaries.
- subunitwidth (number greater than or equal to 0)
  default: 1
  Sets the stroke width (in px) of the subunits boundaries.
- showframe (boolean)
  Sets whether or not a frame is drawn around the map.
- framecolor (color)
  default: "#444"
  Sets the color the frame.
- framewidth (number greater than or equal to 0)
  default: 1
  Sets the stroke width (in px) of the frame.
- bgcolor (color)
  default: "#fff"
  Set the background color of the map
- lonaxis
  - range (list)
    Sets the range of this axis (in degrees).
    Each dict has one or more of the keys listed below.
  - showgrid (boolean)
    Sets whether or not graticule are shown on the map.
  - tick0 (number)
    Sets the graticule's starting tick longitude/latitude.
  - dtick (number)
    Sets the graticule's longitude/latitude tick step.
  - gridcolor (color)
    default: "#eee"
    Sets the graticule's stroke color.
  - gridwidth (number greater than or equal to 0)
    default: 1
    Sets the graticule's stroke width (in px).
- lataxis
  - range (list)
    Sets the range of this axis (in degrees).
    Each dict has one or more of the keys listed below.
  - showgrid (boolean)
    Sets whether or not graticule are shown on the map.
  - tick0 (number)
    Sets the graticule's starting tick longitude/latitude.
  - dtick (number)
    Sets the graticule's longitude/latitude tick step.

- gridcolor (color)
  default: "#eee"
  Sets the graticule's stroke color.
- gridwidth (number greater than or equal to 0)
  default: 1
  Sets the graticule's stroke width (in px).
- mapbox
  - domain
  - x (list)
    default: [0, 1]
    Sets the horizontal domain of this subplot (in plot fraction).
    Each dict has one or more of the keys listed below.
  - y (list)
    default: [0, 1]
    Sets the vertical domain of this subplot (in plot fraction).
    Each dict has one or more of the keys listed below.
  - accesstoken (string)
    Sets the mapbox access token to be used for this mapbox map. Alternatively, the mapbox access token can be set in the configuration options under `mapboxAccessToken`.
  - style (number or categorical coordinate string)
    default: basic
    Sets the Mapbox map style. Either input one of the default Mapbox style names or the URL to a custom style or a valid Mapbox style JSON.
  - center
    - lon (number)
      default: 0
      Sets the longitude of the center of the map (in degrees East).
    - lat (number)
      default: 0
      Sets the latitude of the center of the map (in degrees North).
  - zoom (number)
    default: 1
    Sets the zoom level of the map.
  - bearing (number)
    default: 0
    Sets the bearing angle of the map (in degrees counter-clockwise from North).
  - pitch (number)
    default: 0
    Sets the pitch angle of the map (in degrees, where "0" means perpendicular to the surface of the map).
  - layers
    Each dict has one or more of the keys listed below.
    - sourcetype (enumerated: "geojson" | "vector")
      default: "geojson"
      Sets the source type for this layer. Support for "raster", "image" and "video" source types is coming soon.
    - source (number or categorical coordinate string)
      Sets the source data for this layer. Source can be either a URL, a geojson object (with `sourcetype` set to "geojson") or an array of tile URLS (with `sourcetype` set to "vector").
    - sourcelayer (string)
      default: ""
      Specifies the layer to use from a vector tile source. Required for "vector" source type that supports multiple layers.
    - type (enumerated: "circle" | "line" | "fill" | "symbol")
      default: "circle"
      Sets the layer type. Support for "raster", "background" types is coming soon. Note that "line" and "fill" are not compatible with Point GeoJSON geometries.
    - below (string)
      default: ""
      Determines if the layer will be inserted before the layer with the specified ID. If omitted or set to '', the layer will be inserted above every existing layer.
    - color (color)
      default: "#444"
      Sets the primary layer color. If `type` is "circle", color corresponds to the circle color If `type` is "line", color corresponds to the line color If `type` is "fill", color corresponds to the fill color If `type` is "symbol", color corresponds to the icon color

- opacity (number between or equal to 0 and 1)
  default: 1
  Sets the opacity of the layer.
- circle
- radius (number)
  default: 15
  Sets the circle radius. Has an effect only when `type` is set to "circle".
- line
- width (number)
  default: 2
  Sets the line width. Has an effect only when `type` is set to "line".
- fill
- outlinecolor (color)
  default: "#444"
  Sets the fill outline color. Has an effect only when `type` is set to "fill".
- symbol
- icon (string)
  default: "marker"
  Sets the symbol icon image. Full list: https://www.mapbox.com/maki-icons/
- iconsize (number)
  default: 10
  Sets the symbol icon size. Has an effect only when `type` is set to "symbol".
- text (string)
  default: ""
  Sets the symbol text.
- textfont
  Sets the icon text font. Has an effect only when `type` is set to "symbol".
- family (string)
  default: "Open Sans Regular, Arial Unicode MS Regular"
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- textposition (enumerated: "top left" | "top center" | "top right" | "middle left" | "middle center" | "middle right" | "bottom left" | "bottom center" | "bottom right")
  default: "middle center"
  Sets the positions of the `text` elements with respects to the (x,y) coordinates.

- radialaxis
  - range (list)
    Defines the start and end point of this radial axis.
    Each dict has one or more of the keys listed below.
  - domain (list)
    default: [0, 1]
    Polar chart subplots are not supported yet. This key has currently no effect.
    Each dict has one or more of the keys listed below.
  - orientation (number)
    Sets the orientation (an angle with respect to the origin) of the radial axis.
  - showline (boolean)
    Determines whether or not the line bounding this radial axis will be shown on the figure.
  - showticklabels (boolean)
    Determines whether or not the radial axis ticks will feature tick labels.
  - tickorientation (enumerated: "horizontal" | "vertical")
    Sets the orientation (from the paper perspective) of the radial axis tick labels.
  - ticklen (number greater than or equal to 0)
    Sets the length of the tick lines on this radial axis.
  - tickcolor (color)
    Sets the color of the tick lines on this radial axis.

- ticksuffix (string)
  Sets the length of the tick lines on this radial axis.
- endpadding (number)
- visible (boolean)
  Determines whether or not this axis will be visible.

- angularaxis
- range (list)
  Defines the start and end point of this angular axis.
  Each dict has one or more of the keys listed below.
- domain (list)
  default: [0, 1]
  Polar chart subplots are not supported yet. This key has currently no effect.
  Each dict has one or more of the keys listed below.
- showline (boolean)
  Determines whether or not the line bounding this angular axis will be shown on the figure.
- showticklabels (boolean)
  Determines whether or not the angular axis ticks will feature tick labels.
- tickorientation (enumerated: "horizontal" | "vertical")
  Sets the orientation (from the paper perspective) of the angular axis tick labels.
- ticklen (number greater than or equal to 0)
  Sets the length of the tick lines on this angular axis.
- tickcolor (color)
  Sets the color of the tick lines on this angular axis.
- ticksuffix (string)
  Sets the length of the tick lines on this angular axis.
- endpadding (number)
- visible (boolean)
  Determines whether or not this axis will be visible.

- direction (enumerated: "clockwise" | "counterclockwise")
  For polar plots only. Sets the direction corresponding to positive angles.
- orientation (angle)
  For polar plots only. Rotates the entire polar by the given angle.

- legend
- bgcolor (color)
  Sets the legend background color.
- bordercolor (color)
  default: "#444"
  Sets the color of the border enclosing the legend.
- borderwidth (number greater than or equal to 0)
  default: 0
  Sets the width (in px) of the border enclosing the legend.
- font
  Sets the font used to text the legend items.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- orientation (enumerated: "v" | "h")
  default: "v"
  Sets the orientation of the legend.
- traceorder (flaglist string)
  Any combination of "reversed", "grouped" joined with a "+" OR "normal".
  examples: "reversed", "grouped", "reversed+grouped", "normal"
  Determines the order at which the legend items are displayed. If "normal", the items are displayed top-to-bottom in the same order as the input data. If "reversed", the items are displayed in the opposite order as "normal". If "grouped", the items are displayed in groups (when a trace `legendgroup` is provided). if "grouped+reversed", the items are displayed in the opposite order as "grouped".

- tracegroupgap (number greater than or equal to 0)
  default: 10
  Sets the amount of vertical space (in px) between legend groups.
- x (number between or equal to -2 and 3)
  default: 1.02
  Sets the x position (in normalized coordinates) of the legend.
- xanchor (enumerated: "auto" | "left" | "center" | "right")
  default: "left"
  Sets the legend's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the legend.
- y (number between or equal to -2 and 3)
  default: 1
  Sets the y position (in normalized coordinates) of the legend.
- yanchor (enumerated: "auto" | "top" | "middle" | "bottom")
  default: "auto"
  Sets the legend's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the legend.

- annotations
  Each dict has one or more of the keys listed below.
  An annotation is a text element that can be placed anywhere in the plot. It can be positioned with respect to relative coordinates in the plot or with respect to the actual data coordinates of the graph. Annotations can be shown with or without an arrow.
  - visible (boolean)
    default: True
    Determines whether or not this annotation is visible.
  - text (string)
    Sets the text associated with this annotation. Plotly uses a subset of HTML tags to do things like newline (
), bold (), italics (), hyperlinks (). Tags , <sup>, <sub> are also supported.
  - textangle (angle)
    default: 0
    Sets the angle at which the `text` is drawn with respect to the horizontal.
  - font
    Sets the annotation text font.
    - family (string)
      HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
    - size (number greater than or equal to 1)
    - color (color)
  - opacity (number between or equal to 0 and 1)
    default: 1
    Sets the opacity of the annotation (text + arrow).
  - align (enumerated: "left" | "center" | "right")
    default: "center"
    Sets the vertical alignment of the `text` with respect to the set `x` and `y` position. Has only an effect if `text` spans more two or more lines (i.e. `text` contains one or more
 HTML tags).
  - bgcolor (color)
    default: "rgba(0,0,0,0)"
    Sets the background color of the annotation.
  - bordercolor (color)
    default: "rgba(0,0,0,0)"
    Sets the color of the border enclosing the annotation `text`.
  - borderpad (number greater than or equal to 0)
    default: 1
    Sets the padding (in px) between the `text` and the enclosing border.
  - borderwidth (number greater than or equal to 0)
    default: 1
    Sets the width (in px) of the border enclosing the annotation `text`.

- showarrow (boolean)
  default: True
  Determines whether or not the annotation is drawn with an arrow. If "True", `text` is placed near the arrow's tail. If "False", `text` lines up with the `x` and `y` provided.
- arrowcolor (color)
  Sets the color of the annotation arrow.
- arrowhead (integer between or equal to 0 and 8)
  default: 1
  Sets the annotation arrow head style.
- arrowsize (number greater than or equal to 0.3)
  default: 1
  Sets the size (in px) of annotation arrow head.
- arrowwidth (number greater than or equal to 0.1)
  Sets the width (in px) of annotation arrow.
- standoff (number greater than or equal to 0)
  default: 0
  Sets a distance, in pixels, to move the arrowhead away from the position it is pointing at, for example to point at the edge of a marker independent of zoom.
- ax (number or categorical coordinate string)
  Sets the x component of the arrow tail about the arrow head. If `axref` is `pixel`, a positive (negative) component corresponds to an arrow pointing from right to left (left to right). If `axref` is an axis, this is an absolute value on that axis, like `x`, NOT a relative value.
- ay (number or categorical coordinate string)
  Sets the y component of the arrow tail about the arrow head. If `ayref` is `pixel`, a positive (negative) component corresponds to an arrow pointing from bottom to top (top to bottom). If `ayref` is an axis, this is an absolute value on that axis, like `y`, NOT a relative value.
- axref (enumerated: "pixel" | "/^x([2-9]|[1-9][0-9]+)?$/")
  default: "pixel"
  Indicates in what terms the tail of the annotation (ax,ay) is specified. If `pixel`, `ax` is a relative offset in pixels from `x`. If set to an x axis id (e.g. "x" or "x2"), `ax` is specified in the same terms as that axis. This is useful for trendline annotations which should continue to indicate the correct trend when zoomed.
- ayref (enumerated: "pixel" | "/^y([2-9]|[1-9][0-9]+)?$/")
  default: "pixel"
  Indicates in what terms the tail of the annotation (ax,ay) is specified. If `pixel`, `ay` is a relative offset in pixels from `y`. If set to a y axis id (e.g. "y" or "y2"), `ay` is specified in the same terms as that axis. This is useful for trendline annotations which should continue to indicate the correct trend when zoomed.
- xref (enumerated: "paper" | "/^x([2-9]|[1-9][0-9]+)?$/")
  Sets the annotation's x coordinate axis. If set to an x axis id (e.g. "x" or "x2"), the `x` position refers to an x coordinate If set to "paper", the `x` position refers to the distance from the left side of the plotting area in normalized coordinates where 0 (1) corresponds to the left (right) side.
- x (number or categorical coordinate string)
  Sets the annotation's x position. If the axis `type` is "log", then you must take the log of your desired range. If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
- xanchor (enumerated: "auto" | "left" | "center" | "right")
  default: "auto"
  Sets the text box's horizontal position anchor This anchor binds the `x` position to the "left", "center" or "right" of the annotation. For example, if `x` is set to 1, `xref` to "paper" and `xanchor` to "right" then the right-most portion of the annotation lines up with the right-most edge of the plotting area. If "auto", the anchor is equivalent to "center" for data-referenced annotations or if there is an arrow, whereas for paper-referenced with no arrow, the anchor picked corresponds to the closest side.
- yref (enumerated: "paper" | "/^y([2-9]|[1-9][0-9]+)?$/")
  Sets the annotation's y coordinate axis. If set to an y axis id (e.g. "y" or "y2"), the `y` position refers to an y coordinate If set to "paper", the `y` position refers to the distance from the bottom of the plotting area in normalized coordinates where 0 (1) corresponds to the bottom (top).
- y (number or categorical coordinate string)
  Sets the annotation's y position. If the axis `type` is "log", then you must take the log of your desired range. If the axis `type` is "date", it should be date strings, like date data, though Date objects and unix milliseconds will be accepted and converted to strings. If the axis `type` is "category", it should be numbers, using the scale where each category is assigned a serial number from zero in the order it appears.
- yanchor (enumerated: "auto" | "top" | "middle" | "bottom")
  default: "auto"
  Sets the text box's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the annotation. For example, if `y` is set to 1, `yref` to "paper" and `yanchor` to "top" then the top-most portion of

the annotation lines up with the top-most edge of the plotting area. If "auto", the anchor is equivalent to "middle" for data-referenced annotations or if there is an arrow, whereas for paper-referenced with no arrow, the anchor picked corresponds to the closest side.

- clicktoshow (`enumerated: False | "onoff" | "onout"`)
  Makes this annotation respond to clicks on the plot. If you click a data point that exactly matches the `x` and `y` values of this annotation, and it is hidden (visible: False), it will appear. In "onoff" mode, you must click the same point again to make it disappear, so if you click multiple points, you can show multiple annotations. In "onout" mode, a click anywhere else in the plot (on another data point or not) will hide this annotation. If you need to show/hide this annotation in response to different `x` or `y` values, you can set `xclick` and/or `yclick`. This is useful for example to label the side of a bar. To label markers though, `standoff` is preferred over `xclick` and `yclick`.
- xclick (`number or categorical coordinate string`)
  Toggle this annotation when clicking a data point whose `x` value is `xclick` rather than the annotation's `x` value.
- yclick (`number or categorical coordinate string`)
  Toggle this annotation when clicking a data point whose `y` value is `yclick` rather than the annotation's `y` value.

- shapes
  Each dict has one or more of the keys listed below.
- visible (`boolean`)
  default: `True`
  Determines whether or not this shape is visible.
- type (`enumerated: "circle" | "rect" | "path" | "line"`)
  Specifies the shape type to be drawn. If "line", a line is drawn from (`x0`,`y0`) to (`x1`,`y1`) If "circle", a circle is drawn from ((`x0`+`x1`)/2, (`y0`+`y1`)/2)) with radius (|(`x0`+`x1`)/2 - `x0`|, |(`y0`+`y1`)/2 -`y0`)|) If "rect", a rectangle is drawn linking (`x0`,`y0`), (`x1`,`y0`), (`x1`,`y1`), (`x0`,`y1`), (`x0`,`y0`) If "path", draw a custom SVG path using `path`.
- layer (`enumerated: "below" | "above"`)
  default: `"above"`
  Specifies whether shapes are drawn below or above traces.
- xref (`enumerated: "paper" | "/^x([2-9]|[1-9][0-9]+)?$/"`)
  Sets the shape's x coordinate axis. If set to an x axis id (e.g. "x" or "x2"), the `x` position refers to an x coordinate If set to "paper", the `x` position refers to the distance from the left side of the plotting area in normalized coordinates where "0" ("1") corresponds to the left (right) side. If the axis `type` is "log", then you must take the log of your desired range. If the axis `type` is "date", then you must convert the date to unix time in milliseconds.
- x0 (`number or categorical coordinate string`)
  Sets the shape's starting x position. See `type` for more info.
- x1 (`number or categorical coordinate string`)
  Sets the shape's end x position. See `type` for more info.
- yref (`enumerated: "paper" | "/^y([2-9]|[1-9][0-9]+)?$/"`)
  Sets the annotation's y coordinate axis. If set to an y axis id (e.g. "y" or "y2"), the `y` position refers to an y coordinate If set to "paper", the `y` position refers to the distance from the bottom of the plotting area in normalized coordinates where "0" ("1") corresponds to the bottom (top).
- y0 (`number or categorical coordinate string`)
  Sets the shape's starting y position. See `type` for more info.
- y1 (`number or categorical coordinate string`)
  Sets the shape's end y position. See `type` for more info.
- path (`string`)
  For `type` "path" - a valid SVG path but with the pixel values replaced by data values. There are a few restrictions / quirks only absolute instructions, not relative. So the allowed segments are: M, L, H, V, Q, C, T, S, and Z arcs (A) are not allowed because radius rx and ry are relative. In the future we could consider supporting relative commands, but we would have to decide on how to handle date and log axes. Note that even as is, Q and C Bezier paths that are smooth on linear axes may not be smooth on log, and vice versa. no chained "polybezier" commands - specify the segment type for each one. On category axes, values are numbers scaled to the serial numbers of categories because using the categories themselves there would be no way to describe fractional positions On data axes: because space and T are both normal components of path strings, we can't use either to separate date from time parts. Therefore we'll use underscore for this purpose: 2015-02-21_13:45:56.789
- opacity (`number between or equal to 0 and 1`)
  default: `1`
  Sets the opacity of the shape.
- line
  - color (`color`)
    Sets the line color.
  - width (`number greater than or equal to 0`)
    default: `2`
    Sets the line width (in px).

- dash (string)
  default: "solid"
  Sets the style of the lines. Set to a dash string type or a dash length in px.
  - fillcolor (color)
    default: "rgba(0,0,0,0)"
    Sets the color filling the shape's interior.
- images
  Each dict has one or more of the keys listed below.
  - visible (boolean)
    default: True
    Determines whether or not this image is visible.
  - source (string)
    Specifies the URL of the image to be used. The URL must be accessible from the domain where the plot code is run, and can be either relative or absolute.
  - layer (enumerated: "below" | "above")
    default: "above"
    Specifies whether images are drawn below or above traces. When `xref` and `yref` are both set to `paper`, image is drawn below the entire plot area.
  - sizex (number)
    default: 0
    Sets the image container size horizontally. The image will be sized based on the `position` value. When `xref` is set to `paper`, units are sized relative to the plot width.
  - sizey (number)
    default: 0
    Sets the image container size vertically. The image will be sized based on the `position` value. When `yref` is set to `paper`, units are sized relative to the plot height.
  - sizing (enumerated: "fill" | "contain" | "stretch")
    default: "contain"
    Specifies which dimension of the image to constrain.
  - opacity (number between or equal to 0 and 1)
    default: 1
    Sets the opacity of the image.
  - x (number or categorical coordinate string)
    default: 0
    Sets the image's x position. When `xref` is set to `paper`, units are sized relative to the plot height. See `xref` for more info
  - y (number or categorical coordinate string)
    default: 0
    Sets the image's y position. When `yref` is set to `paper`, units are sized relative to the plot height. See `yref` for more info
  - xanchor (enumerated: "left" | "center" | "right")
    default: "left"
    Sets the anchor for the x position
  - yanchor (enumerated: "top" | "middle" | "bottom")
    default: "top"
    Sets the anchor for the y position.
  - xref (enumerated: "paper" | "/^x([2-9]|[1-9][0-9]+)?$/")
    default: "paper"
    Sets the images's x coordinate axis. If set to a x axis id (e.g. "x" or "x2"), the `x` position refers to an x data coordinate If set to "paper", the `x` position refers to the distance from the left of plot in normalized coordinates where "0" ("1") corresponds to the left (right).
  - yref (enumerated: "paper" | "/^y([2-9]|[1-9][0-9]+)?$/")
    default: "paper"
    Sets the images's y coordinate axis. If set to a y axis id (e.g. "y" or "y2"), the `y` position refers to a y data coordinate. If set to "paper", the `y` position refers to the distance from the bottom of the plot in normalized coordinates where "0" ("1") corresponds to the bottom (top).
- updatemenus
  Each dict has one or more of the keys listed below.
  - visible (boolean)
    Determines whether or not the update menu is visible.
  - type (enumerated: "dropdown" | "buttons")
    default: "dropdown"
    Determines whether the buttons are accessible via a dropdown menu or whether the buttons are stacked horizontally or vertically

- direction (enumerated: "left" | "right" | "up" | "down")
  default: "down"
  Determines the direction in which the buttons are laid out, whether in a dropdown menu or a row/column of buttons. For `left` and `up`, the buttons will still appear in left-to-right or top-to-bottom order respectively.
- active (integer greater than or equal to -1)
  default: 0
  Determines which button (by index starting from 0) is considered active.
- showactive (boolean)
  default: True
  Highlights active dropdown item or active button if True.
- buttons
  Each dict has one or more of the keys listed below.
  - method (enumerated: "restyle" | "relayout" | "animate" | "update")
    default: "restyle"
    Sets the Plotly method to be called on click.
  - args (list)
    Sets the arguments values to be passed to the Plotly method set in `method` on click.
    Each dict has one or more of the keys listed below.
  - label (string)
    default: ""
    Sets the text label to appear on the button.
- x (number between or equal to -2 and 3)
  default: -0.05
  Sets the x position (in normalized coordinates) of the update menu.
- xanchor (enumerated: "auto" | "left" | "center" | "right")
  default: "right"
  Sets the update menu's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the range selector.
- y (number between or equal to -2 and 3)
  default: 1
  Sets the y position (in normalized coordinates) of the update menu.
- yanchor (enumerated: "auto" | "top" | "middle" | "bottom")
  default: "top"
  Sets the update menu's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the range selector.
- pad
  Sets the padding around the buttons or dropdown menu.
  - t (number)
    default: 0
    The amount of padding (in px) along the top of the component.
  - r (number)
    default: 0
    The amount of padding (in px) on the right side of the component.
  - b (number)
    default: 0
    The amount of padding (in px) along the bottom of the component.
  - l (number)
    default: 0
    The amount of padding (in px) on the left side of the component.
- font
  Sets the font of the update menu button text.
  - family (string)
    HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
  - size (number greater than or equal to 1)
  - color (color)
- bgcolor (color)
  Sets the background color of the update menu buttons.

- bordercolor (color)
  default: "#BEC8D9"
  Sets the color of the border enclosing the update menu.
- borderwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the border enclosing the update menu.

- sliders
  Each dict has one or more of the keys listed below.
- visible (boolean)
  default: True
  Determines whether or not the slider is visible.
- active (number greater than or equal to 0)
  default: 0
  Determines which button (by index starting from 0) is considered active.
- steps
  Each dict has one or more of the keys listed below.
  - method (enumerated: "restyle" | "relayout" | "animate" | "update")
    default: "restyle"
    Sets the Plotly method to be called when the slider value is changed.
  - args (list)
    Sets the arguments values to be passed to the Plotly method set in `method` on slide.
    Each dict has one or more of the keys listed below.
  - label (string)
    Sets the text label to appear on the slider
  - value (string)
    Sets the value of the slider step, used to refer to the step programatically. Defaults to the slider label if not provided.
- lenmode (enumerated: "fraction" | "pixels")
  default: "fraction"
  Determines whether this slider length is set in units of plot "fraction" or in "pixels. Use `len` to set the value.
- len (number greater than or equal to 0)
  default: 1
  Sets the length of the slider This measure excludes the padding of both ends. That is, the slider's length is this length minus the padding on both ends.
- x (number between or equal to -2 and 3)
  default: 0
  Sets the x position (in normalized coordinates) of the slider.
- pad
  Set the padding of the slider component along each side.
  - t (number)
    default: 20
    The amount of padding (in px) along the top of the component.
  - r (number)
    default: 0
    The amount of padding (in px) on the right side of the component.
  - b (number)
    default: 0
    The amount of padding (in px) along the bottom of the component.
  - l (number)
    default: 0
    The amount of padding (in px) on the left side of the component.
- xanchor (enumerated: "auto" | "left" | "center" | "right")
  default: "left"
  Sets the slider's horizontal position anchor. This anchor binds the `x` position to the "left", "center" or "right" of the range selector.
- y (number between or equal to -2 and 3)
  default: 0
  Sets the y position (in normalized coordinates) of the slider.
- yanchor (enumerated: "auto" | "top" | "middle" | "bottom")
  default: "top"
  Sets the slider's vertical position anchor This anchor binds the `y` position to the "top", "middle" or "bottom" of the range selector.
- transition

- duration (number greater than or equal to 0)
  default: 150
  Sets the duration of the slider transition
- easing (enumerated: "linear" | "quad" | "cubic" | "sin" | "exp" | "circle" | "elastic" | "back" | "bounce" | "linear-in" | "quad-in" | "cubic-in" | "sin-in" | "exp-in" | "circle-in" | "elastic-in" | "back-in" | "bounce-in" | "linear-out" | "quad-out" | "cubic-out" | "sin-out" | "exp-out" | "circle-out" | "elastic-out" | "back-out" | "bounce-out" | "linear-in-out" | "quad-in-out" | "cubic-in-out" | "sin-in-out" | "exp-in-out" | "circle-in-out" | "elastic-in-out" | "back-in-out" | "bounce-in-out")
  default: "cubic-in-out"
  Sets the easing function of the slider transition
- currentvalue
- visible (boolean)
  default: True
  Shows the currently-selected value above the slider.
- xanchor (enumerated: "left" | "center" | "right")
  default: "left"
  The alignment of the value readout relative to the length of the slider.
- offset (number)
  default: 10
  The amount of space, in pixels, between the current value label and the slider.
- prefix (string)
  When currentvalue.visible is True, this sets the prefix of the label.
- suffix (string)
  When currentvalue.visible is True, this sets the suffix of the label.
- font
  Sets the font of the current value label text.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- font
  Sets the font of the slider step labels.
- family (string)
  HTML font family - the typeface that will be applied by the web browser. The web browser will only be able to apply a font if it is available on the system which it operates. Provide multiple font families, separated by commas, to indicate the preference in which to apply fonts if they aren't available on the system. The plotly service (at https://plot.ly or on-premise) generates images on a server, where only a select number of fonts are installed and supported. These include "Arial", "Balto", "Courier New", "Droid Sans",, "Droid Serif", "Droid Sans Mono", "Gravitas One", "Old Standard TT", "Open Sans", "Overpass", "PT Sans Narrow", "Raleway", "Times New Roman".
- size (number greater than or equal to 1)
- color (color)
- activebgcolor (color)
  default: "#dbdde0"
  Sets the background color of the slider grip while dragging.
- bgcolor (color)
  default: "#f8fafc"
  Sets the background color of the slider.
- bordercolor (color)
  default: "#bec8d9"
  Sets the color of the border enclosing the slider.
- borderwidth (number greater than or equal to 0)
  default: 1
  Sets the width (in px) of the border enclosing the slider.

- ticklen (number greater than or equal to 0)
  default: 7
  Sets the length in pixels of step tick marks
- tickcolor (color)
  default: "#333"
  Sets the color of the border enclosing the slider.
- tickwidth (number greater than or equal to 0)
  default: 1
  Sets the tick width (in px).
- minorticklen (number greater than or equal to 0)
  default: 4
  Sets the length in pixels of minor step tick marks

- calendar (enumerated: "gregorian" | "chinese" | "coptic" | "discworld" | "ethiopian" | "hebrew" | "islamic" | "julian" | "mayan" | "nanakshahi" | "nepali" | "persian" | "jalali" | "taiwan" | "thai" | "ummalqura")
  default: "gregorian"
  Sets the default calendar system to use for interpreting and displaying dates throughout the plot.

layout

- font
  - family
  - size
  - color
- title
- titlefont
  - family
  - size
  - color
- autosize
- width
- height
- margin
  - l
  - r
  - t
  - b
  - pad
  - autoexpand
- paper_bgcolor
- plot_bgcolor
- separators
- hidesources
- showlegend
- dragmode
- hovermode
- xaxis
  - color
  - title
  - titlefont
    - family
    - size
    - color
  - type
  - autorange
  - rangemode
  - range
    Each dict has one or more of the keys listed below.
  - fixedrange
  - tickmode
  - nticks
  - tick0

- dtick
- tickvals
- ticktext
- ticks
- mirror
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
  - family
  - size
  - color
- tickangle
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- showexponent
- exponentformat
- separatethousands
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- zeroline
- zerolinecolor
- zerolinewidth
- anchor
- side
- overlaying
- domain

  Each dict has one or more of the keys listed below.
- position
- categoryorder
- categoryarray
- rangeslider
  - bgcolor
  - bordercolor
  - borderwidth
  - range

    Each dict has one or more of the keys listed below.
  - thickness
  - visible
- rangeselector
  - visible
  - buttons

    Each dict has one or more of the keys listed below.
  - step
  - stepmode
  - count
  - label
  - x
  - xanchor
  - y
  - yanchor
  - font
  - family
  - size
  - color
  - bgcolor

- activecolor
- bordercolor
- borderwidth
- calendar
- tickvalssrc
- ticktextsrc
- categoryarraysrc
- vaxis
- color
- title
- titlefont
- family
- size
- color
- type
- autorange
- rangemode
- range

  Each dict has one or more of the keys listed below.
- fixedrange
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- mirror
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
- family
- size
- color
- tickangle
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- showexponent
- exponentformat
- separatethousands
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- zeroline
- zerolinecolor
- zerolinewidth
- anchor
- side
- overlaying
- domain

  Each dict has one or more of the keys listed below.
- position
- categoryorder
- categoryarray
- calendar
- tickvalssrc
- ticktextsrc

- categoryarraysrc
- ternary
- domain
  - x
    Each dict has one or more of the keys listed below.
  - y
    Each dict has one or more of the keys listed below.
- bgcolor
- sum
- aaxis
  - title
  - titlefont
  - family
  - size
  - color
  - color
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen
  - tickwidth
  - tickcolor
  - showticklabels
  - showtickprefix
  - tickprefix
  - showticksuffix
  - ticksuffix
  - showexponent
  - exponentformat
  - separatethousands
  - tickfont
  - family
  - size
  - color
  - tickangle
  - tickformat
  - hoverformat
  - showline
  - linecolor
  - linewidth
  - showgrid
  - gridcolor
  - gridwidth
  - tickvalssrc
  - ticktextsrc
- baxis
  - title
  - titlefont
  - family
  - size
  - color
  - color
  - tickmode
  - nticks
  - tick0
  - dtick
  - tickvals
  - ticktext
  - ticks
  - ticklen

- tickwidth
- tickcolor
- showticklabels
- showtickprefix
- tickprefix
- showticksuffix
- ticksuffix
- showexponent
- exponentformat
- separatethousands
- tickfont
- family
- size
- color
- tickangle
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- tickvalssrc
- ticktextsrc
- caxis
- title
- titlefont
- family
- size
- color
- color
- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- ticklen
- tickwidth
- tickcolor
- showticklabels
- showtickprefix
- tickprefix
- showticksuffix
- ticksuffix
- showexponent
- exponentformat
- separatethousands
- tickfont
- family
- size
- color
- tickangle
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth

- tickvalssrc
- ticktextsrc
- scene
  - bgcolor
  - camera
    - up
      - x
      - y
      - z
    - center
      - x
      - y
      - z
    - eye
      - x
      - y
      - z
  - domain
    - x
      Each dict has one or more of the keys listed below.
    - y
      Each dict has one or more of the keys listed below.
  - aspectmode
  - aspectratio
    - x
    - y
    - z
  - xaxis
    - showspikes
    - spikesides
    - spikethickness
    - spikecolor
    - showbackground
    - backgroundcolor
    - showaxeslabels
    - color
    - categoryorder
    - categoryarray
    - title
    - titlefont
      - family
      - size
      - color
    - type
    - autorange
    - rangemode
    - range
      Each dict has one or more of the keys listed below.
    - tickmode
    - nticks
    - tick0
    - dtick
    - tickvals
    - ticktext
    - ticks
    - mirror
    - ticklen
    - tickwidth
    - tickcolor
    - showticklabels
    - tickfont
      - family
      - size
      - color

- tickangle
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- showexponent
- exponentformat
- separatethousands
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- zeroline
- zerolinecolor
- zerolinewidth
- calendar
- categoryarraysrc
- tickvalssrc
- ticktextsrc
  - yaxis
- showspikes
- spikesides
- spikethickness
- spikecolor
- showbackground
- backgroundcolor
- showaxeslabels
- color
- categoryorder
- categoryarray
- title
- titlefont
- family
- size
- color
- type
- autorange
- rangemode
- range

Each dict has one or more of the keys listed below.

- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- mirror
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
- family
- size
- color
- tickangle
- tickprefix
- showtickprefix

- ticksuffix
- showticksuffix
- showexponent
- exponentformat
- separatethousands
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- zeroline
- zerolinecolor
- zerolinewidth
- calendar
- categoryarraysrc
- tickvalssrc
- ticktextsrc
- zaxis
- showspikes
- spikesides
- spikethickness
- spikecolor
- showbackground
- backgroundcolor
- showaxeslabels
- color
- categoryorder
- categoryarray
- title
- titlefont
- family
- size
- color
- type
- autorange
- rangemode
- range

Each dict has one or more of the keys listed below.

- tickmode
- nticks
- tick0
- dtick
- tickvals
- ticktext
- ticks
- mirror
- ticklen
- tickwidth
- tickcolor
- showticklabels
- tickfont
- family
- size
- color
- tickangle
- tickprefix
- showtickprefix
- ticksuffix
- showticksuffix
- showexponent

- exponentformat
- separatethousands
- tickformat
- hoverformat
- showline
- linecolor
- linewidth
- showgrid
- gridcolor
- gridwidth
- zeroline
- zerolinecolor
- zerolinewidth
- calendar
- categoryarraysrc
- tickvalssrc
- ticktextsrc
- dragmode
- hovermode
- geo
- domain
- x
  Each dict has one or more of the keys listed below.
- y
  Each dict has one or more of the keys listed below.
- resolution
- scope
- projection
- type
- rotation
- lon
- lat
- roll
- parallels
  Each dict has one or more of the keys listed below.
- scale
- showcoastlines
- coastlinecolor
- coastlinewidth
- showland
- landcolor
- showocean
- oceancolor
- showlakes
- lakecolor
- showrivers
- rivercolor
- riverwidth
- showcountries
- countrycolor
- countrywidth
- showsubunits
- subunitcolor
- subunitwidth
- showframe
- framecolor
- framewidth
- bgcolor
- lonaxis
- range
  Each dict has one or more of the keys listed below.
- showgrid
- tick0
- dtick
- gridcolor

- gridwidth
  - lataxis
    - range
      Each dict has one or more of the keys listed below.
    - showgrid
    - tick0
    - dtick
    - gridcolor
    - gridwidth
  - mapbox
    - domain
      - x
        Each dict has one or more of the keys listed below.
      - y
        Each dict has one or more of the keys listed below.
    - accesstoken
    - style
    - center
      - lon
      - lat
    - zoom
    - bearing
    - pitch
    - layers
      Each dict has one or more of the keys listed below.
      - sourcetype
      - source
      - sourcelayer
      - type
      - below
      - color
      - opacity
      - circle
      - radius
      - line
      - width
      - fill
      - outlinecolor
      - symbol
      - icon
      - iconsize
      - text
      - textfont
      - family
      - size
      - color
      - textposition
  - radialaxis
    - range
      Each dict has one or more of the keys listed below.
    - domain
      Each dict has one or more of the keys listed below.
    - orientation
    - showline
    - showticklabels
    - tickorientation
    - ticklen
    - tickcolor
    - ticksuffix
    - endpadding
    - visible
  - angularaxis
    - range
      Each dict has one or more of the keys listed below.

- domain
  Each dict has one or more of the keys listed below.
- showline
- showticklabels
- tickorientation
- ticklen
- tickcolor
- ticksuffix
- endpadding
- visible
- direction
- orientation
- legend
- bgcolor
- bordercolor
- borderwidth
- font
  - family
  - size
  - color
- orientation
- traceorder
- tracegroupgap
- x
- xanchor
- y
- yanchor
- annotations
  Each dict has one or more of the keys listed below.
  An annotation is a text element that can be placed anywhere in the plot. It can be positioned with respect to relative coordinates in the plot or with respect to the actual data coordinates of the graph. Annotations can be shown with or without an arrow.
  - visible
  - text
  - textangle
  - font
    - family
    - size
    - color
  - opacity
  - align
  - bgcolor
  - bordercolor
  - borderpad
  - borderwidth
  - showarrow
  - arrowcolor
  - arrowhead
  - arrowsize
  - arrowwidth
  - standoff
  - ax
  - ay
  - axref
  - ayref
  - xref
  - x
  - xanchor
  - yref
  - y
  - yanchor
  - clicktoshow
  - xclick
  - yclick
- shapes
  Each dict has one or more of the keys listed below.

- visible
- type
- layer
- xref
- x0
- x1
- yref
- y0
- y1
- path
- opacity
- line
  - color
  - width
  - dash
- fillcolor

- images
  Each dict has one or more of the keys listed below.
  - visible
  - source
  - layer
  - sizex
  - sizey
  - sizing
  - opacity
  - x
  - y
  - xanchor
  - yanchor
  - xref
  - yref

- updatemenus
  Each dict has one or more of the keys listed below.
  - visible
  - type
  - direction
  - active
  - showactive
  - buttons
    Each dict has one or more of the keys listed below.
    - method
    - args
      Each dict has one or more of the keys listed below.
    - label
  - x
  - xanchor
  - y
  - yanchor
  - pad
    - t
    - r
    - b
    - l
  - font
    - family
    - size
    - color
  - bgcolor
  - bordercolor
  - borderwidth

- sliders
  Each dict has one or more of the keys listed below.
  - visible
  - active
  - steps
    Each dict has one or more of the keys listed below.

- method
- args

Each dict has one or more of the keys listed below.

- label
- value
  - lenmode
  - len
  - x
  - pad
    - t
    - r
    - b
    - l
  - xanchor
  - y
  - yanchor
  - transition
    - duration
    - easing
  - currentvalue
    - visible
    - xanchor
    - offset
    - prefix
    - suffix
    - font
      - family
      - size
      - color
  - font
    - family
    - size
    - color
  - activebgcolor
  - bgcolor
  - bordercolor
  - borderwidth
  - ticklen
  - tickcolor
  - tickwidth
  - minorticklen
- calendar

Still need help?

Contact Us

- community.plot.ly
- support.plot.ly
- github.com/plotly

For guaranteed 24 hour response turnarounds, upgrade to our Premium or Enterprise plans.

- **API**
  - Documentation
  - API Libraries
  - REST APIs
  - Plotly.js
  - Hardware
- **About Us**
  - Team
  - Careers
  - Plotly Blog
  - Modern Data
- **Help**
  - Knowledge Base
  - Benchmarks
  - Workshop

- **Solutions**
  - Plans & Pricing
  - Enterprise
  - Education
  - Plotly.js
- **Connect**
  - Twitter
  - Facebook
  - Github
  - LinkedIn
  - Google+