p5.chart.js Library Documentation

Reference Manual of Methods Exposed to the User

Siddharth Chattoraj

December 12, 2025

Contents

1 Global Configuration & State

The library attaches to the `p5.prototype`. The following global properties and methods affect the overall behavior of the library.

1.1 Global Properties

These properties are accessible via `p5.prototype.chart`.

Property	Value / Description
`chart.palette`	Array of default hex strings
`chart.defaultTooltipColumns`	Global default tooltip column specification (Default: `null`). If set to an array like `[col:label,label:Label,col:value,label:Value]`, charts will build tooltips from these columns when hovering.
`chart.defaultSubtitle`	Global default subtitle (Default: `null`). If a chart does not provide `options.subtitle`, this value is used.
`chart.nanPolicy`	Default policy for handling NaN values. Defaults to `warn`. Options: `warn`, `silent`, `strict`.
`chart.autoFitCanvas`	Boolean (Default: `true`). If true, resizes the p5 canvas to match the CSS display size to prevent distortion.
`chart.inputs`	Internal cache for DOM elements (e.g., Table search inputs).
`chart.hoverState`	Object storing global tooltip state (`active`, `x`, `y`, `content`, `tooltipScale`, `tooltipTextSize`, `tooltipFont`).

1.2 Preload Methods

These methods are registered to be used within the p5.js `preload()` function.

`loadDataFrame(path, callback)`

Loads a CSV file and converts it to a DataFrame. Returns a proxy object that mimics the Table/DataFrame structure.

2 Data Structure: DataFrame

The `DataFrame` class is the core structure for data manipulation.

2.1 Constructor

```
new p5.prototype.chart.DataFrame(data, [columns])
```

- **data**: Array of objects OR Array of Arrays.
- **columns**: Array of strings (required if `data` is an Array of Arrays).

2.2 Properties

- **columns**: Returns a copy of the column list.
- **rows**: Returns a copy of the rows array.

2.3 Methods

Method Signature	Description
`col(name)`	Returns an array of values for the specified column.
`filter(colNameOrFn, operator, value)`	Filters rows. Can use a function `(row) => bool`, or comparison syntax: `filter(Age, >, 25)`. Operators: >, <, >=, <=, ===, ==, !==, !=.

`transform(colName, fn)`	Mutates the DataFrame by applying `fn(value)` to every entry in `colName`.
`addColumn(colName, valueOrFn)`	Adds a column. `valueOrFn` can be a static value or a function `(row, index) => val`.
`rename(oldName, newName)`	Renames a column in place.
`select(columnNames)`	Returns a new DataFrame containing only the specified columns.
`drop(columnNames)`	Returns a new DataFrame excluding the specified columns.
`group(groupCols, agg)`	Groups data. `agg` can be a function or an object map (e.g., `{Sales: sum}`). Supported agg strings: `sum`, `mean`, `avg`, `min`, `max`, `count`.
`pivot(indexCol, columnCol, valueCol, aggFunc)`	Creates a cross-tabulation. `aggFunc` defaults to `sum`.
`sort(colName, order)`	Sorts rows. `order` is `ascending` (default) or `descending`.
`head(n)`	Returns a new DataFrame with the first `n` rows (default 5).
`tail(n)`	Returns a new DataFrame with the last `n` rows (default 5).
`unique(colName)`	Returns an array of unique values in a column.
`convertDate(colName, parserOrOptions)`	Converts a column to Date objects. `parserOrOptions` can be a format string, a parser function, or an object: `{format, parser, output, unit, onInvalid}`.

2.4 Helper Functions

`createDataFrame(data, cols)` Alias for instantiating `new DataFrame`.

`tableToDataFrame(pathOrTable, type, options)` Converts a p5.Table or loads a file into a DataFrame.

3 Chart Visualization Methods

All chart methods follow the signature `p5.method(data, options)`. The `data` argument accepts a DataFrame, an Array of objects, or a p5.Table. The `options` argument is a key–alue object that describes how the chart should look and behave.

3.1 Universal Layout & Metadata Options

The following options apply to **all** chart types (Bar, Pie, Series, Scatter, Hist, Geo, Table).

Option Key	**Description / Default**
`width`	Overrides canvas width.
`height`	Overrides canvas height.
`margin`	Object `{top, right, bottom, left}`. Defaults: `{top:60, right:40, bottom:60, left:80}`.
`background`	Hex string, color object, or `transparent` / `none`.
`font`	Font family string.
`palette`	Array of hex strings for data coloring.
`responsive`	Boolean. If false, disables mobile scaling logic.
`mobilePadding`	Boolean. Force extra right padding on mobile.
`responsiveScale`	Number. Multiplier (0.4 - 2.0) for responsive scaling.
`autoFitCanvas`	Boolean. Overrides the global `autoFitCanvas` setting.
`nanPolicy`	String: `warn`, `silent`, `strict`.
Titles & Meta	
`title`	String. Main chart title.
`titleSize`	Number (px).
`titleX`, `titleY`	Number. Exact pixel overrides.
`titleWrap`	Boolean. Default `true`.

`subtitle`	String. Subtitle text.
`subtitleSize`	Number (px).
`subtitleBold`	Boolean.
`subtitleX`, `subtitleY`	Number. Exact pixel overrides.
`subtitleWrap`	Boolean. Default `true`.
`source`, `author`	Strings. displayed in the footer.
`showSourceAuthor`	Boolean. Enables footer.
`sourceAuthorWrap`	Boolean. Wrap footer text. Default `true`.
`sourceAuthorSize`	Number.
`sourceAuthorX/Y`	Number. Exact pixel overrides.
`textAlign`	p5 constants: `LEFT`, `CENTER`, `RIGHT`.
Legend	
`legend`	Boolean, String (`horizontal`/`vertical`), or Object `{enabled, layout}`.
`legend.items`	If `legend` is an Object, `legend.items` can be provided as `[label,color, ...]` or `[Label, ...]`.
`legendItems`	Array of objects `{label, color}`.
`legendLabels`	Array of strings.
`legendColors`	Array of colors.
`legendTextSize`	Number.
`legendBoxSize`	Number.
`legendBarWidth`	Number. Width of gradient legend bar (used when a gradient legend is active).
`legendBarHeight`	Number. Height of gradient legend bar (used when a gradient legend is active).
`legendX`, `legendY`	Number. Exact pixel overrides.
`legendTextColor`	Color.
Tooltips	
`tooltip`	Function for custom tooltip content. Signature varies by chart type; should return a string or array of strings.
`tooltipColumns`	Array of objects `[col, label, ...]` used to build tooltip lines from row fields (or special tooltip columns per chart; see each chart section). Falls back to `chart.defaultTooltipColumns`.
`tooltipScale`	Number. Scale multiplier.
`tooltipTextSize`	Number (px).
`tooltipFont`	String.

3.2 Axis Configuration Options

Applies to Cartesian charts (Bar, Series, Scatter, Hist, Table(search input)).

Option Key	**Description**
`xLabel`	String. X-axis label.
`yLabel`	String. Y-axis label.
`axisLabelSize`	Number.
`xLabelWrap`	Boolean. Default `true`.
`yLabelWrap`	Boolean. Default `true`.
`xLabelX`, `xLabelY`	Number. Exact pixel overrides.
`yLabelX`, `yLabelY`	Number. Exact pixel overrides or padding adjustments.
`yLabelGapToPlot`	Number. Distance between plot and Y-label.
`yLabelGapToTicks`	Number. Distance between ticks and Y-label.

3.3 Chart-Specific: Bar Chart

Method: `bar(data, options)`
NaN Behavior: Skips invalid bars.

Option Key	**Description**
`x` (or `category`)	Column name for categories.

`y` (or `values`)	Column name or Array of names for values.
`orientation`	`horizontal` (default) or `vertical`.
`mode`	`stacked` or default grouped.
`labelPos`	`auto`, `inside`, `outside`, `bottom`, `none`.
`labelBold`	Boolean.
`labelSpace`	Number. Fixed space for labels (horizontal bars).
`numTicks` / `tickCount`	Target number of ticks.
`showGrid` / `grid`	Boolean.
`gridColor` / `gridColour`	Color string.
`gridWeight`	Number.

3.4 Chart-Specific: Pie Chart

Method: `pie(data, options)`
NaN Behavior: Defaults to `strict` (Error). Can be set to `warn`.

Option Key	Description
`label`	Column name for slice labels.
`value`	Column name for slice values.
`style`	`pie` (default) or `donut`.
`radius`	Number. Radius in pixels.
`holeRadius`	Number (0.0 - 1.0). Default 0.6.
`labelSize`	Number. Overrides default pie label font size.
`labelPos`	`inside`, `outside`, `none`.
`labelContent`	`value`, `percent`, `name_value`, `name_percent`, `all` (alias `name_value_percent`).
`showConnectors`	Boolean. Draws lines for outside labels.
`lineColor`	Stroke color.
`lineSize`	Stroke weight.
`outsideLabelMaxWidth`	Number. Max width for truncated outside labels.

3.5 Chart-Specific: Series (Line) Chart

Method: `series(data, options)`
NaN Behavior: Creates a break (gap) in the line.

Option Key	Description
`x`	Column name for X-axis.
`y`	Column name or Array of names for Y-axis.
`minX`, `maxX`	Number. Manual X-axis bounds.
`minY`, `maxY`	Number. Manual Y-axis bounds.
`lineSize`	Number. Line stroke weight.
`pointSize`	Number. Point diameter.
`dots`	Boolean. Default `true`.
`pointStyle`	`filled` (default) or `hollow`.
`showValues`	`true`, `false`, or `click`.
`labelPos`	`auto`, `bottom`.
`seriesTooltipColumn`	String. If set and `tooltipColumns` is not provided, uses `row[seriesTooltipColumn]` as a single-line tooltip label on hover.

3.6 Chart-Specific: Scatter Plot

Method: `scatter(data, options)`
NaN Behavior: Skips points with NaN X or Y values.

Option Key	Description
`x`	Column name for X-axis.
`y`	Column name for Y-axis.

`size`	Column name for variable point size.
`color`	Column name for variable point color (Numeric or Categorical).
`minSize`, `maxSize`	Number. Pixel range for variable sizing.
`pointSize`	Number. Fixed size if `size` column is not used.
`minX`, `maxX`	Number. Manual X-axis bounds.
`minY`, `maxY`	Number. Manual Y-axis bounds.
`connect`	Boolean. Draws a line connecting points in order.
`lineColor`	Color. Used if `connect` is true.
`lineSize`	Number. Stroke weight of connecting line (when `connect` is true).
`baseColor`	Color. Used if `color` column is not used.
`pointStyle`	`filled` (default) or `hollow`.
`showValues`	`true`, `false`, or `click`.
`labelPos`	`auto` (default) or `bottom`. Controls value label placement when `showValues` is enabled.

3.7 Chart-Specific: Histogram

Method: `hist(data, options)`
NaN Behavior: Filters out NaN values before binning.

Option Key	**Description**
`x` (or `column`)	Column name to analyze.
`bins`	Number. Target bin count. Default 10.
`showLabels`	Boolean. Show counts above bars. Default `true`.
`borderColor`	Color string.
`borderWeight`	Number.

3.8 Chart-Specific: Table

Method: `table(data, options)`
NaN Behavior: Displays "—" in gray.

Option Key	**Description**
`x`, `y`	Number. Canvas coordinates.
`width`	Number. Table width.
`maxRows`	Number. Rows per page. Default 10.
`searchable`	Boolean. Adds search input. Default `true` if rows > 10.
`sortable`	Boolean. Click headers to sort. Default `true`.
`pagination`	Boolean. Show controls. Default `true` if rows > 10.
`page`	Number. Current page index.
`onPageChange`	Function callback.
`id`	String. DOM ID for search input.
`headerColor`	Color.
`rowColor1`, `rowColor2`	Colors for alternating rows.
`hoverColor`	Color.
`borderColor`	Color. Table border color.
`nanIndicator`	Boolean. Default `true`.

3.9 Chart-Specific: Geo Chart

Method: `geo(data, options)`
NaN Behavior: Skips points with invalid Lat/Lon.

Option Key	**Description**
`lat` (or `latitude`)	Column name.
`lon` (or `longitude`)	Column name.
`label`	Column name for labels.
`value`	Column name for values (used in tooltips).
`centerLat`, `centerLon`	Number. Initial map center. Auto-calculated if omitted.

`zoom`	Number (2-18). Initial zoom. Auto-calculated if omitted.
`pointColor`	Color.
`pointSize`	Number.
`showLabels`	Boolean. Default `true`.
`showControls`	Boolean. Display Zoom/Pan help box.

4 Export Utilities

`toPNG(filename)`

Exports the current canvas as a PNG. Default filename: `chart.png`.

`toCSV(data, filename, options)`

Exports data to a CSV file.

- **data**: DataFrame, Array, or Object.
- **filename**: Default `data.csv`.
- **options**:
  - `delimiter`: Default `,`.
  - `includeHeader`: Boolean. Default `true`.
  - `columns`: Array of column names to include.