Package 'muiCharts'

July 28, 2026

Type Package

Title 'MUI X Charts' for 'shiny' Apps and 'Quarto'

Version 0.1.0

Maintainer Felix Luginbuhl <felix.luginbuhl@protonmail.ch>

Description 'MUI X Charts' React chart components for data visualization for building 'shiny' applications and 'quarto' documents. Bundles the MIT-licensed community edition of the '@mui/x-charts' JavaScript library (the commercial 'Pro' tier is not included).

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports muiMaterial (>= 0.2.1), htmltools, shiny.react (>= 0.4.0), utils

Suggests dplyr, tidyr, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/lgnbhl/muiCharts

BugReports https://github.com/lgnbhl/muiCharts/issues

NeedsCompilation no

Author Felix Luginbuhl [aut, cre, cph] (ORCID: <https://orcid.org/0009-0008-6625-2899>), MUI [cph] (Copyright holder of the bundled '@mui/x-charts' JavaScript library and its '@mui/x-charts-vendor', '@mui/x-internals', '@mui/x-internal-gestures' and '@mui/system' (subpath imports) helpers; see LICENSE.note), Mike Bostock [cph] (Copyright holder of the bundled D3 modules and 'internmap' (ISC license), vendored via '@mui/x-charts-vendor'; see LICENSE.note),

Repository CRAN

Date/Publication 2026-07-28 16:20:02 UTC

Contents

AnimatedArea *AnimatedArea*

Description

https://mui.com/x/api/charts/animated-area/

Usage

```
AnimatedArea(...)
```

Arguments

...	Props to pass to the component.

Details

- skipAnimation bool
  Default is FALSE If true, animations are skipped.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

AnimatedLine *AnimatedLine*

Description

https://mui.com/x/api/charts/animated-line/

Usage

```
AnimatedLine(...)
```

Arguments

...	Props to pass to the component.

Details

- skipAnimation bool
  Default is FALSE If true, animations are skipped.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

AreaElement *AreaElement*

Description

https://mui.com/x/api/charts/area-element/

Usage

```
AreaElement(...)
```

Arguments

...	Props to pass to the component.

Details

- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

AreaPlot *AreaPlot*

Description

https://mui.com/x/api/charts/area-plot/

Usage

```
AreaPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- onItemClick `func`
  Default is - Callback fired when a line area item is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, lineItemIdentifier: LineItemIdentifier) => voidevent The event source of the callback.lineItemIdentifier The line item identifier.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`BarChart` *BarChart*

Description

https://mui.com/x/api/charts/bar-chart/

Usage

```
BarChart(...)
```

Arguments

...	Props to pass to the component.

Details

- series `Array<object>`
  Default is - The series to display in the bar chart. An array of BarSeries objects.
- axisHighlight `x?: 'band', 'line', 'none', y?: 'band', 'line', 'none'`
  Default is - The configuration of axes highlight. Default is set to 'band' in the bar direction. Depends on layout prop.See highlighting docs for more details.
- borderRadius `number`
  Default is - Defines the border radius of the bar element.
- brushConfig `enabled?: bool, preventHighlight?: bool, preventTooltip?: bool`
  Default is - Configuration for the brush interaction.
- colors `Array<string>, func`
  Default is rainbowSurgePalette Color palette used to colorize multiple series.

- dataset `Array<object>`
  Default is - An array of objects that can be used to populate series and axes data using their dataKey property.
- disableAxisListener `bool`
  Default is FALSE If true, the charts will not listen to the mouse move event. It might break interactive features, but will improve performance.
- grid `horizontal?: bool, vertical?: bool`
  Default is - Option to display a cartesian grid in the background.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- hideLegend `bool`
  Default is - If true, the legend is not rendered.
- highlightedAxis `Array< axisId: number, string, dataIndex: number >`
  Default is - The controlled axis highlight. Identified by the axis id, and data index.
- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- layout `'horizontal', 'vertical'`
  Default is 'vertical' The direction of the bar elements.
- loading `bool`
  Default is FALSE If true, a loading overlay is displayed.
- localeText `object`
  Default is - Localized text for chart components.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onAxisClick `func`
  Default is - The function called for onClick events. The second argument contains information about all line/bar elements at the current mouse position.Signature:function(event: MouseEvent, data: null | ChartsAxisData) => voidevent The mouse event recorded on the <svg/> element.data The data about the clicked axis and items associated with it.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onHighlightedAxisChange `func`
  Default is - The function called when the pointer position corresponds to a new axis data item. This update can either be caused by a pointer movement, or an axis update. In case of multiple axes, the function is called if at least one axis is updated. The argument contains the identifier for all axes with a data property.Signature:function(axisItems: Array<AxisItemIdentifier>) => voidaxisItems The array of axes item identifiers.

- onItemClick `func`
  Default is - Callback fired when a bar item is clicked.Signature:function(event: React.MouseEvent<SVGElement, MouseEvent>, barItemIdentifier: BarItemIdentifier) => voidevent The event source of the callback.barItemIdentifier The bar item identifier.
- showToolbar `bool`
  Default is FALSE If true, shows the default chart toolbar.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.
- xAxis `Array< axis?: 'x', barGapRatio?: number, categoryGapRatio?: number, classes?: object, colo`

`BarElement` *BarElement*

Description

https://mui.com/x/api/charts/bar-element/

Usage

```
BarElement(...)
```

Arguments

...	Props to pass to the component.

Details

- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`BarLabel` *BarLabel*

Description

https://mui.com/x/api/charts/bar-label/

Usage

```
BarLabel(...)
```

Arguments

...	Props to pass to the component.

Details

- height `number`
  Default is - Height of the bar this label belongs to.
- width `number`
  Default is - Width of the bar this label belongs to.
- x `number`
  Default is - Position in the x-axis of the bar this label belongs to.
- xOrigin `number`
  Default is - The x-coordinate of the stack this bar label belongs to.
- y `number`
  Default is - Position in the y-axis of the bar this label belongs to.
- yOrigin `number`
  Default is - The y-coordinate of the stack this bar label belongs to.
- placement `'center', 'outside'`
  Default is 'center' The placement of the bar label. It controls whether the label is rendered in the center or outside the bar.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`BarPlot` *BarPlot*

Description

https://mui.com/x/api/charts/bar-plot/

Usage

```
BarPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- barLabel `'value', func`
  Default is - If provided, the function will be used to format the label of the bar. It can be set to 'value' to display the current value.
- barLabelPlacement `'outside', 'inside'`
  Default is - The placement of the bar label. It controls whether the label is rendered inside or outside the bar.
- borderRadius `number`
  Default is - Defines the border radius of the bar element.
- onItemClick `func`
  Default is - Callback fired when a bar item is clicked.Signature:function(event: React.MouseEvent<SVGElement, MouseEvent>, barItemIdentifier: BarItemIdentifier) => voidevent The event source of the callback.barItemIdentifier The bar item identifier.
- skipAnimation `bool`
  Default is undefined If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsAxis *ChartsAxis*

Description

https://mui.com/x/api/charts/charts-axis/

Usage

```
ChartsAxis(...)
```

Arguments

...	Props to pass to the component.

Details

- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsAxisHighlight *ChartsAxisHighlight*

Description

https://mui.com/x/api/charts/charts-axis-highlight/

Usage

```
ChartsAxisHighlight(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsAxisHighlightPath` *ChartsAxisHighlightPath*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsAxisHighlightPath(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsAxisTooltipContent` *ChartsAxisTooltipContent*

Description

https://mui.com/x/api/charts/charts-axis-tooltip-content/

Usage

```
ChartsAxisTooltipContent(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsBrushOverlay *ChartsBrushOverlay*

Description

https://mui.com/x/api/charts/charts-brush-overlay/

Usage

```
ChartsBrushOverlay(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsClipPath *ChartsClipPath*

Description

https://mui.com/x/api/charts/charts-clip-path/

Usage

```
ChartsClipPath(...)
```

Arguments

...	Props to pass to the component.

Details

- id `string`
  Default is NA The id of the clip path.
- offset `bottom?: number, left?: number, right?: number, top?: number`
  Default is NA Offset, in pixels, of the clip path rectangle from the drawing area.A positive value will move the rectangle outside the drawing area.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsContainer *ChartsContainer*

Description

https://mui.com/x/api/charts/charts-container/

Usage

```
ChartsContainer(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsDataProvider *ChartsDataProvider*

Description

https://mui.com/x/api/charts/charts-data-provider/

Usage

```
ChartsDataProvider(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsGrid *ChartsGrid*

Description

https://mui.com/x/api/charts/charts-grid/

Usage

```
ChartsGrid(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.
- horizontal `bool`
  Default is NA Displays horizontal grid.
- vertical `bool`
  Default is NA Displays vertical grid.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsItemTooltipContent *ChartsItemTooltipContent*

Description

https://mui.com/x/api/charts/charts-item-tooltip-content/

Usage

```
ChartsItemTooltipContent(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsLabel *ChartsLabel*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsLabel(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsLabelMark *ChartsLabelMark*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsLabelMark(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsLayerContainer *ChartsLayerContainer*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsLayerContainer(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsLegend *ChartsLegend*

Description

https://mui.com/x/api/charts/charts-legend/

Usage

```
ChartsLegend(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- direction `'horizontal', 'vertical'`
  Default is - The direction of the legend layout. The default depends on the chart.
- onItemClick `func`
  Default is - Callback fired when a legend item is clicked.Signature:function(event: React.MouseEvent<HTMLButtonEle MouseEvent>, legendItem: SeriesLegendItemContext, index: number) => voidevent The click event.legendItem The legend item data.index The index of the clicked legend item.

- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsLocalizationProvider` *ChartsLocalizationProvider*

Description

https://mui.com/x/api/charts/charts-localization-provider/

Usage

```
ChartsLocalizationProvider(...)
```

Arguments

`...`	Props to pass to the component.

Details

- localeText `object`
  Default is NA Localized text for chart components.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsReferenceLine` *ChartsReferenceLine*

Description

https://mui.com/x/api/charts/charts-reference-line/

Usage

```
ChartsReferenceLine(...)
```

Arguments

...	Props to pass to the component.

Details

- axisId `number, string`
  Default is The `id` of the first defined axis. The id of the axis used for the reference value.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- label `string`
  Default is - The label to display along the reference line.
- labelAlign `'end', 'middle', 'start'`
  Default is 'middle' The alignment if the label is in the chart drawing area.
- labelStyle `object`
  Default is - The style applied to the label.
- lineStyle `object`
  Default is - The style applied to the line.
- spacing `number, x?: number, y?: number`
  Default is x: 0, y: 5 on a horizontal line and x: 5, y: 0 on a vertical line. Additional space around the label in px. Can be a number or an object x, y to distinguish space with the reference line and space with axes.
- x `Date, number, string`
  Default is - The x value associated with the reference line. If defined the reference line will be vertical.
- y `Date, number, string`
  Default is - The y value associated with the reference line. If defined the reference line will be horizontal.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsSurface` *ChartsSurface*

Description

https://mui.com/x/api/charts/charts-surface/

Usage

```
ChartsSurface(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsSvgLayer *ChartsSvgLayer*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsSvgLayer(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsText *ChartsText*

Description

https://mui.com/x/api/charts/charts-text/

Usage

```
ChartsText(...)
```

Arguments

...	Props to pass to the component.

Details

- text `string`
  Default is - Text displayed.
- lineHeight `number`
  Default is - Height of a text line (in em).
- needsComputation `bool`
  Default is FALSE If true, the line width is computed.
- style `object`
  Default is - Style applied to text elements.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsToolbar` *ChartsToolbar*

Description

https://mui.com/x/api/charts/toolbar/

Wraps the `Toolbar` component of `@mui/x-charts`. It is exported as `ChartsToolbar` in R because `muiMaterial` already exports Material UI's `Toolbar` (the AppBar one) and the two packages are commonly attached together.

Usage

```
ChartsToolbar(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsToolbarButton *ChartsToolbarButton*

Description

https://mui.com/x/api/charts/toolbar-button/

Wraps the ToolbarButton component of @mui/x-charts (renamed ChartsToolbarButton in R for consistency with ChartsToolbar).

Usage

```
ChartsToolbarButton(...)
```

Arguments

...	Props to pass to the component.

Value

Object with shiny.tag class suitable for use in the UI of a Shiny app.

ChartsTooltip *ChartsTooltip*

Description

https://mui.com/x/api/charts/charts-tooltip/

Usage

```
ChartsTooltip(...)
```

Arguments

...	Props to pass to the component.

Details

- anchor 'node', 'pointer'
  Default is 'pointer' Determine if the tooltip should be placed on the pointer location or on the node.
- anchorEl HTML element, object, func
  Default is - An HTML element, virtualElement, or a function that returns either. It's used to set the position of the popper. The return value will passed as the reference object of the Popper instance.

- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- component `elementType`
  Default is - The component used for the root node. Either a string to use a HTML element or a component.
- components `Root?: elementType`
  Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component.Deprecated-use the slots prop instead. This prop will be removed in a future major release. How to migrate.
- componentsProps `root?: func, object`
  Default is The props used for each slot inside the Popper.Deprecated-use the slotProps prop instead. This prop will be removed in a future major release. How to migrate.
- container `HTML element, func`
  Default is - An HTML element or function that returns one. The container will have the portal children appended to it.
- disablePortal `bool`
  Default is FALSE The children will be under the DOM hierarchy of the parent component.
- keepMounted `bool`
  Default is FALSE Always keep the children in the DOM. This prop can be useful in SEO situation or when you want to maximize the responsiveness of the Popper.
- modifiers `Array< data?: object, effect?: func, enabled?: bool, fn?: func, name?: any, options?: objec`
  Default is - Popper.js is based on a "plugin-like" architecture, most of its features are fully encapsulated "modifiers".A modifier is a function that is called each time Popper.js needs to compute the position of the popper. For this reason, modifiers should be very performant to avoid bottlenecks. To learn how to create a modifier, read the modifiers documentation.
- open `bool`
  Default is - If true, the component is shown.
- placement `'auto-end', 'auto-start', 'auto', 'bottom-end', 'bottom-start', 'bottom', 'left-end', 'le`
  Default is 'bottom' Popper placement.
- popperOptions `modifiers?: array, onFirstUpdate?: func, placement?: 'auto-end', 'auto-start', 'auto'`
  Default is Options provided to the Popper.js instance.
- popperRef `func, current?: destroy: func, forceUpdate: func, setOptions: func, state: attributes: `
  Default is - A ref that points to the used popper instance.
- position `'bottom', 'left', 'right', 'top'`
  Default is - Determines the tooltip position relatively to the anchor.
- slotProps `object`
  Default is The props used for each slot inside the Popper.
- slots `object`
  Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component.See Slots API below for more details.
- sx `Array<func, object, bool>, func, object`
  Default is - The system prop that allows defining system overrides as well as additional CSS styles.See the sx page for more details.

- transition `bool`
  Default is FALSE Help supporting a react-transition-group/Transition component.
- trigger `'axis', 'item', 'none'`
  Default is 'axis' Select the kind of tooltip to display - 'item': Shows data about the item below the mouse; - 'axis': Shows values associated with the hovered x value; - 'none': Does not display tooltip.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsTooltipCell` *ChartsTooltipCell*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsTooltipCell(...)
```

Arguments

`...`	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsTooltipContainer` *ChartsTooltipContainer*

Description

https://mui.com/x/api/charts/charts-tooltip-container/

Usage

```
ChartsTooltipContainer(...)
```

Arguments

`...`	Props to pass to the component.

Details

- anchor `'node', 'pointer'`
  Default is 'pointer' Determine if the tooltip should be placed on the pointer location or on the node.
- anchorEl `HTML element, object, func`
  Default is - An HTML element, virtualElement, or a function that returns either. It's used to set the position of the popper. The return value will passed as the reference object of the Popper instance.
- children `node`
  Default is - Popper render function or node.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- component `elementType`
  Default is - The component used for the root node. Either a string to use a HTML element or a component.
- components `Root?: elementType`
  Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component.Deprecated-use the slots prop instead. This prop will be removed in a future major release. How to migrate.
- componentsProps `root?: func, object`
  Default is The props used for each slot inside the Popper.Deprecated-use the slotProps prop instead. This prop will be removed in a future major release. How to migrate.
- container `HTML element, func`
  Default is - An HTML element or function that returns one. The container will have the portal children appended to it.
- disablePortal `bool`
  Default is FALSE The children will be under the DOM hierarchy of the parent component.
- keepMounted `bool`
  Default is FALSE Always keep the children in the DOM. This prop can be useful in SEO situation or when you want to maximize the responsiveness of the Popper.
- modifiers `Array< data?: object, effect?: func, enabled?: bool, fn?: func, name?: any, options?: objec`
  Default is - Popper.js is based on a "plugin-like" architecture, most of its features are fully encapsulated "modifiers".A modifier is a function that is called each time Popper.js needs to compute the position of the popper. For this reason, modifiers should be very performant to avoid bottlenecks. To learn how to create a modifier, read the modifiers documentation.
- open `bool`
  Default is - If true, the component is shown.
- placement `'auto-end', 'auto-start', 'auto', 'bottom-end', 'bottom-start', 'bottom', 'left-end', 'le`
  Default is 'bottom' Popper placement.
- popperOptions `modifiers?: array, onFirstUpdate?: func, placement?: 'auto-end', 'auto-start', 'auto'`
  Default is Options provided to the Popper.js instance.
- popperRef `func, current?: destroy: func, forceUpdate: func, setOptions: func, state: attributes: `
  Default is - A ref that points to the used popper instance.

- position `'bottom', 'left', 'right', 'top'`
  Default is - Determines the tooltip position relatively to the anchor.
- slotProps `object`
  Default is The props used for each slot inside the Popper.
- slots `object`
  Default is The components used for each slot inside the Popper. Either a string to use a HTML element or a component.See Slots API below for more details.
- sx `Array<func, object, bool>, func, object`
  Default is - The system prop that allows defining system overrides as well as additional CSS styles.See the `sx` page for more details.
- transition `bool`
  Default is FALSE Help supporting a react-transition-group/Transition component.
- trigger `'axis', 'item', 'none'`
  Default is 'axis' Select the kind of tooltip to display - 'item': Shows data about the item below the mouse; - 'axis': Shows values associated with the hovered x value; - 'none': Does not display tooltip.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsTooltipPaper` *ChartsTooltipPaper*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsTooltipPaper(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsTooltipRow *ChartsTooltipRow*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsTooltipRow(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsTooltipTable *ChartsTooltipTable*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ChartsTooltipTable(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsWrapper *ChartsWrapper*

Description

https://mui.com/x/api/charts/charts-wrapper/

Usage

```
ChartsWrapper(...)
```

Arguments

...	Props to pass to the component.

Details

- extendVertically `bool`
  Default is `FALSE` if the `height` prop is set. And `TRUE` otherwise. If true, the chart wrapper set height: 100%.
- hideLegend `bool`
  Default is FALSE If true, the legend is not rendered.
- legendDirection `'horizontal', 'vertical'`
  Default is 'horizontal' The direction of the legend.
- legendPosition `horizontal?: 'center', 'end', 'start', vertical?: 'bottom', 'middle', 'top'`
  Default is horizontal: 'center', vertical: 'bottom' The position of the legend.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ChartsXAxis *ChartsXAxis*

Description

https://mui.com/x/api/charts/charts-x-axis/

Usage

```
ChartsXAxis(...)
```

Arguments

...	Props to pass to the component.

Details

- axisId `number, string`
  Default is - The id of the axis to render. If undefined, it will be the first defined axis.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- disableLine `bool`
  Default is FALSE If true, the axis line is disabled.
- disableTicks `bool`
  Default is FALSE If true, the ticks are disabled.
- label `string`
  Default is - The label of the axis.
- labelStyle `object`
  Default is - The style applied to the axis label.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- tickInterval `'auto', array, func`
  Default is 'auto' Defines which ticks are displayed. Its value can be: - 'auto' In such case the ticks are computed based on axis scale and other parameters. - a filtering function of the form (value, index) => boolean which is available only if the axis has "point" scale. - an array containing the values where ticks should be displayed.
- tickLabelInterval `'auto', func`
  Default is 'auto' Defines which ticks get its label displayed. Its value can be: - 'auto' In such case, labels are displayed if they do not overlap with the previous one. - a filtering function of the form (value, index) => boolean. Warning: the index is tick index, not data ones.
- tickLabelMinGap `number`
  Default is 4 The minimum gap in pixels between two tick labels. If two tick labels are closer than this minimum gap, one of them will be hidden.
- tickLabelPlacement `'middle', 'tick'`
  Default is 'middle' The placement of ticks label. Can be the middle of the band, or the tick position. Only used if scale is 'band'.
- tickLabelStyle `object`
  Default is - The style applied to ticks text.
- tickMaxStep `number`
  Default is - Maximal step between two ticks. When using time data, the value is assumed to be in ms. Not supported by categorical axis (band, points).
- tickMinStep `number`
  Default is - Minimal step between two ticks. When using time data, the value is assumed to be in ms. Not supported by categorical axis (band, points).
- tickNumber `number`
  Default is - The number of ticks. This number is not guaranteed. Not supported by categorical axis (band, points).

- tickPlacement `'end', 'extremities', 'middle', 'start'`
  Default is 'extremities' The placement of ticks in regard to the band interval. Only used if scale is 'band'.
- tickSize `number`
  Default is 6 The size of the ticks.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ChartsYAxis` *ChartsYAxis*

Description

https://mui.com/x/api/charts/charts-y-axis/

Usage

```
ChartsYAxis(...)
```

Arguments

...	Props to pass to the component.

Details

- axisId `number, string`
  Default is - The id of the axis to render. If undefined, it will be the first defined axis.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- disableLine `bool`
  Default is FALSE If true, the axis line is disabled.
- disableTicks `bool`
  Default is FALSE If true, the ticks are disabled.
- label `string`
  Default is - The label of the axis.
- labelStyle `object`
  Default is - The style applied to the axis label.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

- tickInterval `'auto', array, func`
  Default is 'auto' Defines which ticks are displayed. Its value can be: - 'auto' In such case the ticks are computed based on axis scale and other parameters. - a filtering function of the form (value, index) => boolean which is available only if the axis has "point" scale. - an array containing the values where ticks should be displayed.
- tickLabelInterval `'auto', func`
  Default is 'auto' Defines which ticks get its label displayed. Its value can be: - 'auto' In such case, labels are displayed if they do not overlap with the previous one. - a filtering function of the form (value, index) => boolean. Warning: the index is tick index, not data ones.
- tickLabelPlacement `'middle', 'tick'`
  Default is 'middle' The placement of ticks label. Can be the middle of the band, or the tick position. Only used if scale is 'band'.
- tickLabelStyle `object`
  Default is - The style applied to ticks text.
- tickMaxStep `number`
  Default is - Maximal step between two ticks. When using time data, the value is assumed to be in ms. Not supported by categorical axis (band, points).
- tickMinStep `number`
  Default is - Minimal step between two ticks. When using time data, the value is assumed to be in ms. Not supported by categorical axis (band, points).
- tickNumber `number`
  Default is - The number of ticks. This number is not guaranteed. Not supported by categorical axis (band, points).
- tickPlacement `'end', 'extremities', 'middle', 'start'`
  Default is 'extremities' The placement of ticks in regard to the band interval. Only used if scale is 'band'.
- tickSize `number`
  Default is 6 The size of the ticks.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ContinuousColorLegend` *ContinuousColorLegend*

Description

https://mui.com/x/api/charts/continuous-color-legend/

Usage

```
ContinuousColorLegend(...)
```

Arguments

...	Props to pass to the component.

Details

- axisDirection `'x', 'y', 'z'`
  Default is 'z' The axis direction containing the color configuration to represent.
- axisId `number, string`
  Default is The first axis item. The id of the axis item with the color configuration to represent.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- direction `'horizontal', 'vertical'`
  Default is 'horizontal' The direction of the legend layout.
- gradientId `string`
  Default is auto-generated id The id for the gradient to use. If not provided, it will use the generated gradient from the axis configuration. The gradientId will be used as fill="url(#gradientId)".
- labelPosition `'start', 'end', 'extremes'`
  Default is 'end' Where to position the labels relative to the gradient.
- maxLabel `func, string`
  Default is formattedValue The label to display at the maximum side of the gradient. Can either be a string, or a function. If not defined, the formatted maximal value is display.
- minLabel `func, string`
  Default is formattedValue The label to display at the minimum side of the gradient. Can either be a string, or a function.
- reverse `bool`
  Default is FALSE If true, the gradient and labels will be reversed.
- rotateGradient `bool`
  Default is - If provided, the gradient will be rotated by 90deg. Useful for linear gradients that are not in the correct orientation.
- thickness `number`
  Default is 12 The thickness of the gradient

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`FocusedBar` *FocusedBar*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
FocusedBar(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

FocusedLineMark *FocusedLineMark*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
FocusedLineMark(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

FocusedPieArc *FocusedPieArc*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
FocusedPieArc(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`FocusedRadarMark` *FocusedRadarMark*

Description

https://mui.com/x/api/charts/focused-radar-mark/

Usage

```
FocusedRadarMark(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`FocusedScatterMark` *FocusedScatterMark*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
FocusedScatterMark(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

Gauge *Gauge*

Description

https://mui.com/x/api/charts/gauge/

Usage

```
Gauge(...)
```

Arguments

...	Props to pass to the component.

Details

- cornerRadius `number, string`
  Default is 0 The radius applied to arc corners (similar to border radius). Set it to '50%' to get rounded arc.
- cx `number, string`
  Default is - The x coordinate of the arc center. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the width the drawing area.
- cy `number, string`
  Default is - The y coordinate of the arc center. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the height the drawing area.
- endAngle `number`
  Default is 360 The end angle (deg).
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- innerRadius `number, string`
  Default is '80%' The radius between circle center and the beginning of the arc. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the maximal radius that fit into the drawing area.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- outerRadius `number, string`
  Default is '100%' The radius between circle center and the end of the arc. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the maximal radius that fit into the drawing area.

- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- startAngle `number`
  Default is 0 The start angle (deg).
- value `number`
  Default is - The value of the gauge. Set to null to not display a value.
- valueMax `number`
  Default is 100 The maximal value of the gauge.
- valueMin `number`
  Default is 0 The minimal value of the gauge.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`GaugeContainer` *GaugeContainer*

Description

https://mui.com/x/api/charts/gauge-container/

Usage

```
GaugeContainer(...)
```

Arguments

...	Props to pass to the component.

Details

- cornerRadius `number, string`
  Default is 0 The radius applied to arc corners (similar to border radius). Set it to '50%' to get rounded arc.
- cx `number, string`
  Default is - The x coordinate of the arc center. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the width the drawing area.
- cy `number, string`
  Default is - The y coordinate of the arc center. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the height the drawing area.

- endAngle `number`
  Default is 360 The end angle (deg).
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- innerRadius `number, string`
  Default is '80%' The radius between circle center and the beginning of the arc. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the maximal radius that fit into the drawing area.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- outerRadius `number, string`
  Default is '100%' The radius between circle center and the end of the arc. Can be a number (in px) or a string with a percentage such as '50%'. The '100%' is the maximal radius that fit into the drawing area.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- startAngle `number`
  Default is 0 The start angle (deg).
- value `number`
  Default is - The value of the gauge. Set to null to not display a value.
- valueMax `number`
  Default is 100 The maximal value of the gauge.
- valueMin `number`
  Default is 0 The minimal value of the gauge.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`GaugeReferenceArc` *GaugeReferenceArc*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
GaugeReferenceArc(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`GaugeValueArc` *GaugeValueArc*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
GaugeValueArc(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

GaugeValueText *GaugeValueText*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
GaugeValueText(...)
```

Arguments

...	Props to pass to the component.

Details

- text `func, string`
  Default is NA The text to display in the gauge. Can be a string or a function receiving the value and returning a string.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

LineChart *LineChart*

Description

https://mui.com/x/api/charts/line-chart/

Usage

```
LineChart(...)
```

Arguments

...	Props to pass to the component.

Details

- series `Array<object>`
  Default is - The series to display in the line chart. An array of LineSeries objects.
- axisHighlight `x?: 'band', 'line', 'none', y?: 'band', 'line', 'none'`
  Default is x: 'line' The configuration of axes highlight.See highlighting docs for more details.
- brushConfig `enabled?: bool, preventHighlight?: bool, preventTooltip?: bool`
  Default is - Configuration for the brush interaction.
- colors `Array<string>, func`
  Default is rainbowSurgePalette Color palette used to colorize multiple series.
- dataset `Array<object>`
  Default is - An array of objects that can be used to populate series and axes data using their dataKey property.
- disableAxisListener `bool`
  Default is FALSE If true, the charts will not listen to the mouse move event. It might break interactive features, but will improve performance.
- disableLineItemHighlight `bool`
  Default is - If true, render the line highlight item.
- experimentalFeatures `preferStrictDomainInLineCharts?: bool`
  Default is - Options to enable features planned for the next major.
- grid `horizontal?: bool, vertical?: bool`
  Default is - Option to display a cartesian grid in the background.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- hideLegend `bool`
  Default is - If true, the legend is not rendered.
- highlightedAxis `Array< axisId: number, string, dataIndex: number >`
  Default is - The controlled axis highlight. Identified by the axis id, and data index.
- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- loading `bool`
  Default is FALSE If true, a loading overlay is displayed.
- localeText `object`
  Default is - Localized text for chart components.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onAreaClick `func`
  Default is - Callback fired when an area element is clicked.

- onAxisClick `func`
  Default is - The function called for onClick events. The second argument contains information about all line/bar elements at the current mouse position.Signature:function(event: MouseEvent, data: null | ChartsAxisData) => voidevent The mouse event recorded on the <svg/> element.data The data about the clicked axis and items associated with it.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onHighlightedAxisChange `func`
  Default is - The function called when the pointer position corresponds to a new axis data item. This update can either be caused by a pointer movement, or an axis update. In case of multiple axes, the function is called if at least one axis is updated. The argument contains the identifier for all axes with a data property.Signature:function(axisItems: Array<AxisItemIdentifier>) => voidaxisItems The array of axes item identifiers.
- onLineClick `func`
  Default is - Callback fired when a line element is clicked.
- onMarkClick `func`
  Default is - Callback fired when a mark element is clicked.
- showToolbar `bool`
  Default is FALSE If true, shows the default chart toolbar.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.
- xAxis `Array< axis?: 'x', barGapRatio?: number, categoryGapRatio?: number, classes?: object, colo`

LineElement *LineElement*

Description

https://mui.com/x/api/charts/line-element/

Usage

```
LineElement(...)
```

Arguments

...	Props to pass to the component.

Details

- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

LineHighlightElement *LineHighlightElement*

Description

https://mui.com/x/api/charts/line-highlight-element/

Usage

```
LineHighlightElement(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

LineHighlightPlot *LineHighlightPlot*

Description

https://mui.com/x/api/charts/line-highlight-plot/

Usage

```
LineHighlightPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

LinePlot *LinePlot*

Description

https://mui.com/x/api/charts/line-plot/

Usage

```
LinePlot(...)
```

Arguments

...	Props to pass to the component.

Details

- onItemClick `func`
  Default is - Callback fired when a line item is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, lineItemIdentifier: LineItemIdentifier) => voidevent The event source of the callback.lineItemIdentifier The line item identifier.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`MarkElement` *MarkElement*

Description

https://mui.com/x/api/charts/mark-element/

Usage

```
MarkElement(...)
```

Arguments

...	Props to pass to the component.

Details

- dataIndex `number`
  Default is - The index to the element in the series' data array.
- shape `'circle', 'cross', 'diamond', 'square', 'star', 'triangle', 'wye'`
  Default is - The shape of the marker.
- isFaded `bool`
  Default is FALSE If true, the marker is faded.
- isHighlighted `bool`
  Default is FALSE If true, the marker is highlighted.
- skipAnimation `bool`
  Default is - If true, animations are skipped.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`MarkPlot` *MarkPlot*

Description

https://mui.com/x/api/charts/mark-plot/

Usage

```
MarkPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- onItemClick `func`
  Default is - Callback fired when a line mark item is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, lineItemIdentifier: LineItemIdentifier) => voidevent The event source of the callback.lineItemIdentifier The line mark item identifier.
- skipAnimation `bool`
  Default is - If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`muiChartsDependency` *Mui Charts UI JS dependency*

Description

Mui Charts UI JS dependency

Usage

```
muiChartsDependency()
```

Value

A list of HTML dependency objects (`htmltools::htmlDependency`) that load the bundled MUI X Charts JavaScript and its shared MUI Material runtime. The muiMaterial dependency is listed first so its bundle (which provides react, react-dom, `@mui/material`, `@mui/system` and the emotion runtime that the charts consume as externals) loads before the chart script. Attach it to custom HTML when building a chart by hand.

Examples

```
# Inspect the dependencies attached to every muiCharts element.
muiChartsDependency()
```

`PieArc` *PieArc*

Description

https://mui.com/x/api/charts/pie-arc/

Usage

```
PieArc(...)
```

Arguments

`...`	Props to pass to the component.

Details

- skipAnimation `bool`
  Default is NA If true, the animation is disabled.
- skipInteraction `bool`
  Default is NA If true, the default event handlers are disabled. Those are used, for example, to display a tooltip or highlight the arc on hover.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

PieArcLabel *PieArcLabel*

Description

https://mui.com/x/api/charts/pie-arc-label/

Usage

```
PieArcLabel(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

PieArcLabelPlot *PieArcLabelPlot*

Description

https://mui.com/x/api/charts/pie-arc-label-plot/

Usage

```
PieArcLabelPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- id `number, string`
  Default is - The id of this series.
- outerRadius `number`
  Default is - The radius between circle center and the end of the arc.
- arcLabel `'formattedValue', 'label', 'value', func`
  Default is - The label displayed into the arc.
- arcLabelMinAngle `number`
  Default is 0 The minimal angle required to display the arc label.

- arcLabelRadius `number`
  Default is (innerRadius - outerRadius) / 2 The radius between circle center and the arc label in px.
- cornerRadius `number`
  Default is 0 The radius applied to arc corners (similar to border radius).
- faded `additionalRadius?: number, arcLabelRadius?: number, color?: string, cornerRadius?: number, ir`
  Default is additionalRadius: -5 Override the arc attributes when it is faded.
- highlighted `additionalRadius?: number, arcLabelRadius?: number, color?: string, cornerRadius?: numbe`
  Default is - Override the arc attributes when it is highlighted.
- innerRadius `number`
  Default is 0 The radius between circle center and the beginning of the arc.
- paddingAngle `number`
  Default is 0 The padding angle (deg) between two arcs.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`PieArcPlot` *PieArcPlot*

Description

https://mui.com/x/api/charts/pie-arc-plot/

Usage

```
PieArcPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- id `number, string`
  Default is - The id of this series.
- outerRadius `number`
  Default is - The radius between circle center and the end of the arc.
- arcLabelRadius `number`
  Default is (innerRadius - outerRadius) / 2 The radius between circle center and the arc label in px.
- cornerRadius `number`
  Default is 0 The radius applied to arc corners (similar to border radius).
- faded `additionalRadius?: number, arcLabelRadius?: number, color?: string, cornerRadius?: number, ir`
  Default is additionalRadius: -5 Override the arc attributes when it is faded.
- highlighted `additionalRadius?: number, arcLabelRadius?: number, color?: string, cornerRadius?: numbe`
  Default is - Override the arc attributes when it is highlighted.
- innerRadius `number`
  Default is 0 The radius between circle center and the beginning of the arc.
- onItemClick `func`
  Default is - Callback fired when a pie item is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, pieItemIdentifier: PieItemIdentifier, item: DefaultizedPieValueType) => voidevent The event source of the callback.pieItemIdentifier The pie item identifier.item The pie item.
- paddingAngle `number`
  Default is 0 The padding angle (deg) between two arcs.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`PiecewiseColorLegend` *PiecewiseColorLegend*

Description

https://mui.com/x/api/charts/piecewise-color-legend/

Usage

```
PiecewiseColorLegend(...)
```

Arguments

...	Props to pass to the component.

Details

- axisDirection `'x', 'y', 'z'`
  Default is 'z' The axis direction containing the color configuration to represent.
- axisId `number, string`
  Default is The first axis item. The id of the axis item with the color configuration to represent.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- direction `'horizontal', 'vertical'`
  Default is 'horizontal' The direction of the legend layout.
- labelFormatter `func`
  Default is - Format the legend labels.Signature:function(params: PiecewiseLabelFormatterParams) => string | nullparams The bound of the piece to format.
- labelPosition `'start', 'end', 'extremes', 'inline-start', 'inline-end'`
  Default is 'extremes' Where to position the labels relative to the gradient.
- markType `'square', 'circle', 'line'`
  Default is 'square' The type of the mark.
- onItemClick `func`
  Default is - Callback fired when a legend item is clicked.Signature:function(event: React.MouseEvent<HTMLButtonEle
  MouseEvent>, legendItem: PiecewiseColorLegendItemContext, index: number) => voidevent
  The click event.legendItem The legend item data.index The index of the clicked legend item.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`PieChart` *PieChart*

Description

https://mui.com/x/api/charts/pie-chart/

Usage

```
PieChart(...)
```

Arguments

...	Props to pass to the component.

Details

- series `Array<object>`
  Default is - The series to display in the pie chart. An array of PieSeries objects.
- colors `Array<string>, func`
  Default is rainbowSurgePalette Color palette used to colorize multiple series.
- dataset `Array<object>`
  Default is - An array of objects that can be used to populate series and axes data using their dataKey property.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- hideLegend `bool`
  Default is - If true, the legend is not rendered.
- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- loading `bool`
  Default is FALSE If true, a loading overlay is displayed.
- localeText `object`
  Default is - Localized text for chart components.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onItemClick `func`
  Default is - Callback fired when a pie arc is clicked.
- showToolbar `bool`
  Default is FALSE If true, shows the default chart toolbar.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`PiePlot` *PiePlot*

Description

https://mui.com/x/api/charts/pie-plot/

Usage

```
PiePlot(...)
```

Arguments

...	Props to pass to the component.

Details

- onItemClick `func`
  Default is - Callback fired when a pie item is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, pieItemIdentifier: PieItemIdentifier, item: DefaultizedPieValueType) => voidevent The event source of the callback.pieItemIdentifier The pie item identifier.item The pie item.
- skipAnimation `bool`
  Default is FALSE If true, animations are skipped.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

print.muiCharts *Print muiCharts components*

Description

When called interactively, renders the component in the IDE viewer panel. Otherwise, falls back to standard shiny.tag printing (raw HTML text).

Usage

```
## S3 method for class 'muiCharts'
print(x, browse = interactive(), ...)
```

Arguments

x	A muiCharts object (also inherits shiny.tag).
browse	Whether to render in viewer. Defaults to TRUE in interactive sessions.
...	Additional arguments passed to print.

Value

Invisibly returns x.

Examples

```
chart <- BarChart(
  series = list(list(data = c(1, 2, 3))),
  height = 300
)
# browse = FALSE prints the underlying HTML instead of opening the viewer.
print(chart, browse = FALSE)
```

RadarAxis *RadarAxis*

Description

https://mui.com/x/api/charts/radar-axis/

Usage

```
RadarAxis(...)
```

Arguments

...	Props to pass to the component.

Details

- angle `number`
  Default is - The absolute rotation angle of the metrics (in degree) If not defined the metric angle will be used.
- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- divisions `number`
  Default is 1 The number of divisions with label.
- dominantBaseline `'alphabetic', 'auto', 'central', 'hanging', 'ideographic', 'inherit', 'mathematica`
  Default is - The labels dominant baseline or a function returning the dominant baseline for a given axis angle (in degree).
- labelOrientation `'horizontal', 'rotated'`
  Default is 'horizontal' Defines how label align with the axis. - 'horizontal': labels stay horizontal and their placement change with the axis angle. - 'rotated': labels are rotated 90deg relatively to their axis.
- metric `string`
  Default is - The metric to get. If undefined, the hook returns null
- textAnchor `'end', 'inherit', 'middle', 'start', func`
  Default is - The labels text anchor or a function returning the text anchor for a given axis angle (in degree).

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarAxisHighlight` *RadarAxisHighlight*

Description

https://mui.com/x/api/charts/radar-axis-highlight/

Usage

```
RadarAxisHighlight(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarChart` *RadarChart*

Description

https://mui.com/x/api/charts/radar-chart/

Usage

```
RadarChart(...)
```

Arguments

...	Props to pass to the component.

Details

- radar `labelFormatter?: func, labelGap?: number, max?: number, metrics: Array<string>, Array< max?: r`
  Default is - The configuration of the radar scales.
- series `Array<object>`
  Default is - The series to display in the bar chart. An array of RadarSeries objects.
- colors `Array<string>, func`
  Default is rainbowSurgePalette Color palette used to colorize multiple series.
- disableAxisListener `bool`
  Default is FALSE If true, the charts will not listen to the mouse move event. It might break interactive features, but will improve performance.
- divisions `number`
  Default is 5 The number of divisions in the radar grid.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- hideLegend `bool`
  Default is - If true, the legend is not rendered.
- highlight `'axis', 'none', 'series'`
  Default is 'axis' Indicates if the chart should highlight items per axis or per series.
- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.

- loading `bool`
  Default is FALSE If true, a loading overlay is displayed.
- localeText `object`
  Default is - Localized text for chart components.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onAreaClick `func`
  Default is - Callback fired when an area is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- onAxisClick `func`
  Default is - The function called for onClick events. The second argument contains information about all line/bar elements at the current mouse position.Signature:function(event: MouseEvent, data: null | ChartsAxisData) => voidevent The mouse event recorded on the <svg/> element.data The data about the clicked axis and items associated with it.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onMarkClick `func`
  Default is - Callback fired when a mark is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- shape `'circular', 'sharp'`
  Default is 'sharp' The grid shape.
- showToolbar `bool`
  Default is FALSE If true, shows the default chart toolbar.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- stripeColor `func`
  Default is (index) => index % 2 === 1 ? (theme.vars || theme).palette.text.secondary : 'none' Get stripe fill color. Set it to null to remove stripesSignature:function(index: number) => stringindex The index of the stripe band.
- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

RadarDataProvider *RadarDataProvider*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
RadarDataProvider(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

RadarGrid *RadarGrid*

Description

https://mui.com/x/api/charts/radar-grid/

Usage

```
RadarGrid(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is - Override or extend the styles applied to the component.See CSS classes API below for more details.
- divisions `number`
  Default is 5 The number of divisions in the radar grid.
- shape `'circular', 'sharp'`
  Default is 'sharp' The grid shape.
- stripeColor `func`
  Default is (index) => index % 2 === 1 ? (theme.vars || theme).palette.text.secondary : 'none' Get stripe fill color. Set it to null to remove stripesSignature:function(index: number) => stringindex The index of the stripe band.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarMetricLabels` *RadarMetricLabels*

Description

https://mui.com/x/api/charts/radar-metric-labels/

Usage

```
RadarMetricLabels(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarSeriesArea` *RadarSeriesArea*

Description

https://mui.com/x/api/charts/radar-series-area/

Usage

```
RadarSeriesArea(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.
- onItemClick `func`
  Default is NA Callback fired when an area is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- seriesId `string`
  Default is NA The id of the series to display. If undefined all series are displayed.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarSeriesMarks` *RadarSeriesMarks*

Description

https://mui.com/x/api/charts/radar-series-marks/

Usage

```
RadarSeriesMarks(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.
- onItemClick `func`
  Default is NA Callback fired when a mark is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- seriesId `string`
  Default is NA The id of the series to display. If undefined all series are displayed.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`RadarSeriesPlot` *RadarSeriesPlot*

Description

https://mui.com/x/api/charts/radar-series-plot/

Usage

```
RadarSeriesPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- classes `object`
  Default is NA Override or extend the styles applied to the component.See CSS classes API below for more details.
- onAreaClick `func`
  Default is NA Callback fired when an area is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- onMarkClick `func`
  Default is NA Callback fired when a mark is clicked.Signature:function(event: React.MouseEvent<SVGPathElement, MouseEvent>, radarItemIdentifier: RadarItemIdentifier) => voidevent The event source of the callback.radarItemIdentifier The radar item identifier.
- seriesId `string`
  Default is NA The id of the series to display. If undefined all series are displayed.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`Scatter` *Scatter*

Description

https://mui.com/x/api/charts/scatter/

Usage

```
Scatter(...)
```

Arguments

...	Props to pass to the component.

Details

- colorGetter `func`
  Default is NA Function to get the color of a scatter item given its data index. The data index argument is optional. If not provided, the color for the entire series is returned. If provided, the color for the specific scatter item is returned.
- onItemClick `func`
  Default is NA Callback fired when clicking on a scatter item.Signature:function(event: MouseEvent, scatterItemIdentifier: ScatterItemIdentifier) => voidevent Mouse event recorded on the <svg/> element.scatterItemIdentifier The scatter item identifier.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

`ScatterChart` *ScatterChart*

Description

https://mui.com/x/api/charts/scatter-chart/

Usage

```
ScatterChart(...)
```

Arguments

...	Props to pass to the component.

Details

- series `Array<object>`
  Default is - The series to display in the scatter chart. An array of ScatterSeries objects.
- axisHighlight `x?: 'band', 'line', 'none', y?: 'band', 'line', 'none'`
  Default is x: 'none', y: 'none' The configuration of axes highlight.See highlighting docs for more details.
- brushConfig `enabled?: bool, preventHighlight?: bool, preventTooltip?: bool`
  Default is - Configuration for the brush interaction.
- colors `Array<string>, func`
  Default is rainbowSurgePalette Color palette used to colorize multiple series.
- dataset `Array<object>`
  Default is - An array of objects that can be used to populate series and axes data using their dataKey property.
- disableAxisListener `bool`
  Default is FALSE If true, the charts will not listen to the mouse move event. It might break interactive features, but will improve performance.
- disableVoronoi `bool`
  Default is FALSE If true, the interaction will not use the Voronoi cell and fall back to hover events.
- grid `horizontal?: bool, vertical?: bool`
  Default is - Option to display a cartesian grid in the background.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- hideLegend `bool`
  Default is - If true, the legend is not rendered.

- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- loading `bool`
  Default is FALSE If true, a loading overlay is displayed.
- localeText `object`
  Default is - Localized text for chart components.
- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is - The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onAxisClick `func`
  Default is - The function called for onClick events. The second argument contains information about all line/bar elements at the current mouse position.Signature:function(event: MouseEvent, data: null | ChartsAxisData) => voidevent The mouse event recorded on the <svg/> element.data The data about the clicked axis and items associated with it.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onItemClick `func`
  Default is - Callback fired when clicking on a scatter item.Signature:function(event: MouseEvent, scatterItemIdentifier: ScatterItemIdentifier) => voidevent The mouse event recorded on the <svg/> element if using Voronoi cells. Or the Mouse event from the scatter element, when disableVoronoi=true.scatterItemIdentifier The scatter item identifier.
- renderer `'svg-batch', 'svg-single'`
  Default is 'svg-single' The type of renderer to use for the scatter plot. - svg-single: Renders every scatter item in a <circle /> element. - svg-batch: Batch renders scatter items in <path /> elements for better performance with large datasets, at the cost of some limitations. Read more: https://mui.com/x/react-charts/scatter/#performance
- showToolbar `bool`
  Default is FALSE If true, shows the default chart toolbar.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- voronoiMaxRadius `'item', number`
  Default is - Defines the maximum distance between a scatter point and the pointer that triggers the interaction. If set to 'item', the radius is the markerSize. If undefined, the radius is assumed to be infinite.

- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.
- xAxis `Array< axis?: 'x', barGapRatio?: number, categoryGapRatio?: number, classes?: object, colo`

`ScatterMarker` *ScatterMarker*

Description

Low-level composition component of MUI X Charts. See https://mui.com/x/react-charts/composition/.

Usage

```
ScatterMarker(...)
```

Arguments

...	Props to pass to the component.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

ScatterPlot *ScatterPlot*

Description

https://mui.com/x/api/charts/scatter-plot/

Usage

```
ScatterPlot(...)
```

Arguments

...	Props to pass to the component.

Details

- onItemClick `func`
  Default is - Callback fired when clicking on a scatter item.Signature:function(event: MouseEvent, scatterItemIdentifier: ScatterItemIdentifier) => voidevent Mouse event recorded on the <svg/> element.scatterItemIdentifier The scatter item identifier.
- renderer `'svg-batch', 'svg-single'`
  Default is 'svg-single' The type of renderer to use for the scatter plot. - svg-single: Renders every scatter item in a <circle /> element. - svg-batch: Batch renders scatter items in <path /> elements for better performance with large datasets, at the cost of some limitations. Read more: https://mui.com/x/react-charts/scatter/#performance
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.

Value

Object with `shiny.tag` class suitable for use in the UI of a Shiny app.

SparkLineChart *SparkLineChart*

Description

https://mui.com/x/api/charts/spark-line-chart/

Usage

```
SparkLineChart(...)
```

Arguments

...	Props to pass to the component.

Details

- data `Array<number>`
  Default is - Data to plot.
- area `bool`
  Default is FALSE Set to true to fill spark line area. Has no effect if plotType='bar'.
- baseline `'max', 'min', number`
  Default is 0 The value of the line at the base of the series area. 'min' the area will fill the space under the line. 'max' the area will fill the space above the line. number the area will fill the space between this value and the line
- brushConfig `enabled?: bool, preventHighlight?: bool, preventTooltip?: bool`
  Default is - Configuration for the brush interaction.
- clipAreaOffset `bottom?: number, left?: number, right?: number, top?: number`
  Default is top: 1, right: 1, bottom: 1, left: 1 The clipped area offset in pixels.This prevents partial clipping of lines when they are drawn on the edge of the drawing area.
- color `func, string`
  Default is rainbowSurgePalette 0 Color used to colorize the sparkline.
- dataset `Array<object>`
  Default is - An array of objects that can be used to populate series and axes data using their dataKey property.
- disableAxisListener `bool`
  Default is FALSE If true, the charts will not listen to the mouse move event. It might break interactive features, but will improve performance.
- disableClipping `bool`
  Default is FALSE When true, the chart's drawing area will not be clipped and elements within can visually overflow the chart.
- disableVoronoi `bool`
  Default is - If true, the voronoi interaction are ignored.
- height `number`
  Default is - The height of the chart in px. If not defined, it takes the height of the parent element.
- highlightedAxis `Array< axisId: number, string, dataIndex: number >`
  Default is - The controlled axis highlight. Identified by the axis id, and data index.
- highlightedItem `dataIndex?: number, seriesId: number, string`
  Default is - The highlighted item. Used when the highlight is controlled.
- id `string`
  Default is - This prop is used to help implement the accessibility logic. If you don't provide this prop. It falls back to a randomly generated id.
- localeText `object`
  Default is - Localized text for chart components.

- margin `number, bottom?: number, left?: number, right?: number, top?: number`
  Default is 5 The margin between the SVG and the drawing area. It's used for leaving some space for extra information such as the x- and y-axis or legend.Accepts a number to be used on all sides or an object with the optional properties: top, bottom, left, and right.
- onAxisClick `func`
  Default is - The function called for onClick events. The second argument contains information about all line/bar elements at the current mouse position.Signature:function(event: MouseEvent, data: null | ChartsAxisData) => voidevent The mouse event recorded on the <svg/> element.data The data about the clicked axis and items associated with it.
- onHighlightChange `func`
  Default is - The callback fired when the highlighted item changes.Signature:function(highlightedItem: HighlightItemData | null) => voidhighlightedItem The newly highlighted item.
- onHighlightedAxisChange `func`
  Default is - The function called when the pointer position corresponds to a new axis data item. This update can either be caused by a pointer movement, or an axis update. In case of multiple axes, the function is called if at least one axis is updated. The argument contains the identifier for all axes with a data property.Signature:function(axisItems: Array<AxisItemIdentifier>) => voidaxisItems The array of axes item identifiers.
- onItemClick `func`
  Default is - Callback fired when clicking close to an item. This is only available for scatter plot for now.Signature:function(event: MouseEvent, scatterItemIdentifier: ScatterItemIdentifier) => voidevent Mouse event caught at the svg levelscatterItemIdentifier Identify which item got clicked
- plotType `'bar', 'line'`
  Default is 'line' Type of plot used.
- showHighlight `bool`
  Default is FALSE Set to true to highlight the value. With line, it shows a point. With bar, it shows a highlight band.
- showTooltip `bool`
  Default is FALSE Set to true to enable the tooltip in the sparkline.
- skipAnimation `bool`
  Default is - If true, animations are skipped. If unset or false, the animations respects the user's prefers-reduced-motion setting.
- slotProps `object`
  Default is The props used for each component slot.
- slots `object`
  Default is Overridable component slots.See Slots API below for more details.
- valueFormatter `func`
  Default is (value: number | null) => (value === null ? '' : value.toString()) Formatter used by the tooltip.Signature:function(value: number) => stringvalue The value to format.
- voronoiMaxRadius `'item', number`
  Default is - Defines the maximum distance between a scatter point and the pointer that triggers the interaction. If set to 'item', the radius is the markerSize. If undefined, the radius is assumed to be infinite.

- width `number`
  Default is - The width of the chart in px. If not defined, it takes the width of the parent element.
- xAxis `axis?: 'x', barGapRatio?: number, categoryGapRatio?: number, classes?: object, colorMap?:`

`starwars_films` *Star Wars Films*

Description

A dataset containing information about the Star Wars films, fetched from the Star Wars API (SWAPI).

Usage

```
starwars_films
```

Format

A data frame with 6 rows and 10 variables:

title `character`. The title of the film.

episode_id `integer`. The episode number of the film.

director `character`. The name of the director of the film.

producer `character`. The name(s) of the producer(s) of the film.

release_date `Date`. The release date of the film in ISO 8601 format.

characters `list` of `character` vectors. A list of character names appearing in the film.

planets `list` of `character` vectors. A list of planet names appearing in the film.

starships `list` of `character` vectors. A list of starship names appearing in the film.

vehicles `list` of `character` vectors. A list of vehicle names appearing in the film.

species `list` of `character` vectors. A list of species names appearing in the film.

Source

https://swapi.info/api/films

`starwars_people` *Star Wars People*

Description

A dataset containing information about people (characters) in the Star Wars universe, fetched from the Star Wars API (SWAPI).

Usage

```
starwars_people
```

Format

A data frame with 82 rows and 11 variables:

name `character`. The name of the character.

height `integer`. The height of the character in centimeters.

mass `double`. The mass of the character in kilograms.

hair_color `character`. The hair color of the character.

skin_color `character`. The skin color of the character.

eye_color `character`. The eye color of the character.

birth_year `double`. The birth year of the character relative to the Battle of Yavin (BBY). A higher number means earlier in the timeline.

gender `character`. The gender of the character.

films `list` of `character` vectors. A list of film titles this character appeared in.

starships `list` of `character` vectors. A list of starship names this character has piloted.

vehicles `list` of `character` vectors. A list of vehicle names this character has piloted.

Source

https://swapi.info/api/people

`starwars_planets` *Star Wars Planets*

Description

A dataset containing information about planets in the Star Wars universe, fetched from the Star Wars API (SWAPI).

Usage

```
starwars_planets
```

Format

A data frame with 60 rows and 11 variables:

name `character`. The name of the planet.

rotation_period `integer`. The number of standard hours it takes for the planet to complete a single rotation on its axis.

orbital_period `integer`. The number of standard days it takes for the planet to orbit once around the local star.

diameter `integer`. The diameter of the planet in kilometers.

climate `character`. The climate(s) of the planet.

gravity `character`. The gravity of the planet relative to standard gravity.

terrain `character`. The terrain type(s) found on the planet.

surface_water `double`. The percentage of the planet surface covered by water or other bodies of water.

population `double`. The average population of sentient beings inhabiting the planet.

residents `list` of `character` vectors. A list of character names who are residents of the planet.

films `list` of `character` vectors. A list of film titles in which this planet appears.

Source

https://swapi.info/api/planets

`starwars_species` *Star Wars Species*

Description

A dataset containing information about species in the Star Wars universe, fetched from the Star Wars API (SWAPI).

Usage

```
starwars_species
```

Format

A data frame with 37 rows and 12 variables:

name `character`. The name of the species.

classification `character`. The classification of the species (e.g. mammal, reptile).

designation `character`. The designation of the species (e.g. sentient, reptilian).

average_height `double`. The average height of the species in centimeters.

average_lifespan `double`. The average lifespan of the species in years.

skin_colors `character`. A comma-separated string of common skin colors for the species.

hair_colors `character`. A comma-separated string of common hair colors for the species.

eye_colors `character`. A comma-separated string of common eye colors for the species.

language `character`. The language commonly spoken by the species.

homeworld `character`. The name of the planet this species originates from.

people `list` of `character` vectors. A list of character names that are members of this species.

films `list` of `character` vectors. A list of film titles in which this species appears.

Source

https://swapi.info/api/species

`starwars_starships` *Star Wars Starships*

Description

A dataset containing information about starships in the Star Wars universe, fetched from the Star Wars API (SWAPI).

Usage

```
starwars_starships
```

Format

A data frame with 36 rows and 15 variables:

name `character`. The name of the starship.

model `character`. The model or official name of the starship.

manufacturer `character`. The manufacturer of the starship.

cost_in_credits `double`. The cost of the starship new, in galactic credits.

length `double`. The length of the starship in meters.

max_atmosphering_speed `double`. The maximum speed of the starship in atmosphere. `NA` if the starship is incapable of atmospheric flight.

crew `double`. The number of personnel needed to run or pilot the starship.

passengers `double`. The number of non-essential people this starship can transport.

cargo_capacity `double`. The maximum cargo load in kilograms.

consumables `character`. The maximum length of time the starship can provide consumables for its entire crew without resupplying.

hyperdrive_rating `double`. The class of the starship's hyperdrive.

MGLT `double`. The maximum number of Megalights this starship can travel in a standard hour.

starship_class `character`. The class of the starship.

pilots `list` of `character` vectors. A list of character names who have piloted this starship.

films `list` of `character` vectors. A list of film titles in which this starship appears.

Source

https://swapi.info/api/starships

Index