

# Package ‘SveltePlots’

July 21, 2025

**Type** Package

**Version** 0.1.0

**Title** A Wrapper for a Svelte Custom Web Component

**Author** Pascal Schmidt [aut, cre]

**Maintainer** Pascal Schmidt <pascal.sfu@gmail.com>

**Description** An interactive charting library built on 'Svelte' and 'D3' to easily produce SVG charts in R. Designed to simplify 'shiny' development by eliminating the need for `renderUI()`, `insertUI()`, `removeUI()`, and 'shiny' proxy functions, using 'Svelte's reactive state system instead.

**URL** <https://github.com/Pascal-Schmidt/SveltePlots>,  
<https://pascal-schmidt.github.io/SveltePlots/>

**BugReports** <https://github.com/Pascal-Schmidt/SveltePlots/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Suggests** testthat (>= 3.0.0), lubridate, rmarkdown, quarto, knitr,  
covr, ggplot2

**Config/testthat/edition** 3

**Imports** data.table, htmlwidgets, magrittr, dplyr, rlang, RColorBrewer,  
padr, purrr, stringr, tibble, tidyr

**Depends** R (>= 4.1.0)

**Language** en-US

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-06-10 09:30:10 UTC

## Contents

|                      |           |
|----------------------|-----------|
| CO2                  | 2         |
| confidence_intervals | 3         |
| dau                  | 4         |
| economics            | 4         |
| fruit                | 5         |
| gapminder            | 6         |
| penguins             | 7         |
| purchases            | 8         |
| quests               | 8         |
| segments             | 9         |
| sp                   | 10        |
| spaes                | 11        |
| sp_add_arrows        | 12        |
| sp_add_segments      | 14        |
| sp_add_series        | 17        |
| sp_add_text          | 20        |
| sp_facet             | 22        |
| sp_title             | 23        |
| sp_tooltip           | 24        |
| sp_x_axis            | 25        |
| sp_y_axis            | 28        |
| SveltePlots-shiny    | 29        |
| walmart_sales_weekly | 30        |
| <b>Index</b>         | <b>31</b> |

---

CO2

*Plant Growth Dataset*

---

### Description

This dataset contains information about the growth of plants under different treatments and concentrations over time.

### Usage

CO2

### Format

A tibble with 84 rows and 6 variables:

**Plant** Factor. The identifier for the plant.

**Type** Factor. The type of plant, here represented as "Quebec".

**Treatment** Factor. The treatment applied to the plant, here represented as "nonchilled".

**conc** Numeric. The concentration level of the treatment.

**uptake** Numeric. The uptake measurement of the plant.

**date** Date. The date of the observation.

### Examples

```
data(CO2)
head(CO2)
```

---

confidence\_intervals *Confidence Intervals Data*

---

### Description

This dataset contains time series data with actual values and corresponding confidence intervals.

### Usage

```
confidence_intervals
```

### Format

A tibble with 368 rows and 7 variables:

**.model\_id** Integer. The model ID.

**.model\_desc** Character. Description of the model.

**.key** Factor. The key indicating actual or forecast data.

**.index** Date. The date of the observation.

**.value** Numeric. The actual value.

**.conf\_lo** Numeric. The lower bound of the confidence interval.

**.conf\_hi** Numeric. The upper bound of the confidence interval.

### Examples

```
data(confidence_intervals)
head(confidence_intervals)
```

---

|     |                                |
|-----|--------------------------------|
| dau | <i>Daily Active Users Data</i> |
|-----|--------------------------------|

---

**Description**

This dataset contains daily active user (DAU) counts for a specific period, along with slot information.

**Usage**

```
dau
```

**Format**

A tibble with 19 rows and 3 variables:

**date** The date, in Date format.

**DAU** Daily active users count, a numeric value.

**Slot** Slot number associated with the DAU count, a numeric value.

**Examples**

```
data(dau)
head(dau)
```

---

|           |                       |
|-----------|-----------------------|
| economics | <i>Economics Data</i> |
|-----------|-----------------------|

---

**Description**

This dataset contains economic data, including personal consumption expenditures, population, personal savings rate, median duration of unemployment, and the number of unemployed individuals, recorded monthly from July 1967.

**Usage**

```
economics
```

**Format**

A tibble with 574 rows and 6 variables:

**date** Date. The date of the observation.

**pce** Numeric. Personal consumption expenditures, in billions of dollars.

**pop** Numeric. Total population, in thousands.

**psavert** Numeric. Personal savings rate, as a percentage.

**uempmed** Numeric. Median duration of unemployment, in weeks.

**unemploy** Numeric. Number of unemployed individuals, in thousands.

**Source**

This dataset is sourced from the Federal Reserve Economic Data (FRED) database. <https://fred.stlouisfed.org/>

**Examples**

```
data(economics)
head(economics)
```

---

fruit

*Example Dataset: Fruit Data*

---

**Description**

This dataset contains information about different types of fruits in various baskets over multiple years. It includes details such as the fruit type, the value, and the color of the fruit.

**Usage**

```
fruit
```

**Format**

A tibble with 16 rows and 6 variables:

**id** Integer. A unique identifier for each record.

**year** Character. The year the data was recorded.

**basket** Integer. The basket number containing the fruits.

**fruit** Character. The type of fruit.

**value** Integer. The value associated with the fruit.

**color** Character. The color of the fruit.

**Examples**

```
data(fruit)
head(fruit)
```

---

gapminder

*Gapminder Dataset*

---

### Description

This dataset provides country-level data on life expectancy, GDP per capita, and population. It is included in the ‘gapminder’ package. For detailed information, please refer to the [gapminder documentation](https://cran.r-project.org/package=gapminder).

### Usage

```
gapminder
```

### Format

A data frame with 1704 rows and 6 variables.

**country** Factor with country names.

**continent** Factor with continent names.

**year** Integer.

**lifeExp** Numeric.

**pop** Integer.

**gdpPercap** Numeric.

### Source

This dataset is sourced from the ‘gapminder’ package.

### See Also

[gapminder documentation](#)

### Examples

```
data(gapminder)
head(gapminder)
```

---

penguins

*Penguins Dataset*

---

## Description

This dataset provides measurements of penguins. It is included in the ‘palmerpenguins’ package. For detailed information, please refer to the [palmerpenguins documentation](https://allisonhorst.github.io/palmerpenguins/).

## Usage

```
penguins
```

## Format

A data frame with 344 rows and 8 variables.

**species** Factor with levels Adelie, Chinstrap, Gentoo.

**island** Factor with levels Biscoe, Dream, Torgersen.

**bill\_length\_mm** Numeric.

**bill\_depth\_mm** Numeric.

**flipper\_length\_mm** Numeric.

**body\_mass\_g** Numeric.

**sex** Factor with levels female, male.

**year** Integer.

## Source

This dataset is sourced from the ‘palmerpenguins’ package.

## See Also

[palmerpenguins documentation](#)

## Examples

```
data(penguins)
head(penguins)
```

---

purchases

*Revenue Data*

---

### Description

This dataset contains revenue data over a period of time with rolling revenue calculations.

### Usage

```
purchases
```

### Format

A data frame with 10 rows and 4 variables:

**date** Date. The timestamp of the revenue data.

**age** Factor. The age range category.

**revenue** Numeric. The revenue for the given date.

**revenue\_roll** Numeric. The rolling revenue calculation.

### Examples

```
data(purchases)
head(purchases)
```

---

quests

*Example Dataset: Retention Data*

---

### Description

This dataset contains retention data for different custom categories and progression stages. It includes the number of observations at two different points (n.x and n.y) and the retention rate.

### Usage

```
quests
```

### Format

A tibble with 38 rows and 5 variables:

**custom\_01** Character. A custom category identifier.

**progression\_2** Factor. The progression stage.

**n.x** Numeric. The number of observations at the first point.

**n.y** Numeric. The number of observations at the second point.

**retention** Numeric. The retention rate, calculated as n.x divided by n.y.



## Examples

```
data(quests)
head(quests)
```

---

segments

*Segments*

---

## Description

This dataset contains information about various events including promotions, gacha events, and experiments. Each event has details such as start and end dates, additional descriptions, and graphical representation attributes.

## Usage

```
segments
```

## Format

A tibble with 658 rows and 8 variables:

**event\_type** Type of the event, a character string.

**start\_date** Start date of the event, in Date format.

**end\_date** End date of the event, in Date format.

**extra\_details** Additional details about the event, a character string.

**colors** Color associated with the event, a character string.

**key** A numeric identifier for the event.

**y\_start** Starting y-coordinate for graphical representation, a numeric value.

**y\_end** Ending y-coordinate for graphical representation, a numeric value.

## Examples

```
data(segments)
head(segments)
```

## Description

This function allows for the creation of various types of plots including scatter plots, time-series plots, line charts, bar charts, density plots, histograms, pie charts, and boxplots. It provides flexibility through several arguments that control the appearance and behavior of the plots. The function is part of the SveltePlots package which leverages the power of Svelte and D3 for rendering.

## Usage

```
sp(  
  data,  
  mapping,  
  type,  
  mode = "grouped",  
  size = 2,  
  alpha = 1,  
  tooltip = TRUE,  
  include_legend = TRUE,  
  colors = NULL,  
  height = 500,  
  combine_same_groups = TRUE,  
  breaks = "Sturges",  
  facet_var = NULL  
)
```

## Arguments

|                             |  |
|-----------------------------|--|
| <code>data</code>           | A data frame containing the data to be plotted.  |
| <code>mapping</code>        | A list specifying the mapping of data to aesthetics, similar to <code>ggplot2</code> 's <code>aes()</code> function. This includes specifying <code>x</code> , <code>y</code> , and <code>group</code> attributes if needed.   |
| <code>type</code>           | Character string specifying the type of plot to create. Accepted values are <code>points</code> (for scatter plots), <code>line</code> (for line charts), <code>bar</code> (for bar charts), <code>histogram</code> , <code>density</code> , <code>pie</code> , and <code>boxplot</code> . |
| <code>mode</code>           | Specifies the mode for bar plots. Accepted values are <code>grouped</code> , <code>stacked</code> , and <code>percent</code> , determining how bars are arranged. Default is <code>grouped</code> .  |
| <code>size</code>           | Numeric value specifying the size of the points or lines.  |
| <code>alpha</code>          | Numeric value specifying the opacity of the points or lines, on a scale from 0 to 1.   |
| <code>tooltip</code>        | Logical indicating whether tooltips should be shown on hover.  |
| <code>include_legend</code> | Logical indicating whether a legend should be included in the plot.  |
| <code>colors</code>         | A vector of colors to be used for the different groups in the plot. If <code>NULL</code> , default colors are used.  |

|                     |  |
|---------------------|--|
| height              | Numeric value specifying the height of the plot in pixels.   |
| combine_same_groups | Logical indicating whether multiple series with the same group should be combined into one legend category.                    |
| breaks              | Controls the number of bins for histograms. Can be a vector or a method compatible with the hist() function's breaks argument. |
| facet_var           | A character vector splitting the data for faceting charts  |

**Value**

A SveltePlots plot object which can be rendered in a web page or an R Markdown document.

**Examples**

```
library(SveltePlots)

data("penguins")
sp(
  data = penguins,
  mapping = spaes(x = flipper_length_mm, y = bill_length_mm, group = species),
  type = "points"
)
```

---

spaes

*Create Aesthetic Mappings for SveltePlots*


---

**Description**

'spaes' creates a list of aesthetic mappings for use with SveltePlot charts. Similar to 'aes' in 'ggplot2' and 'hcaes' in 'highcharter', it defines how data should be mapped to visual properties such as position, size, and color. 'spaes' mappings are used when creating or adding series to plots, ensuring a consistent and expressive interface for defining the appearance of data in charts.

**Usage**

```
spaes(x, y, ...)
```

**Arguments**

|     |  |
|-----|--|
| x   | The x aesthetic mapping. This could be a column name from the data frame or any expression that evaluates to a numeric or categorical value, corresponding to the x-axis position of the elements in the plot. |
| y   | The y aesthetic mapping. Similarly, this is used for mapping data to the y-axis position in the chart. It can be a direct column reference or an expression.   |
| ... | Additional aesthetic mappings. These could include mappings for group, and y_min and y_max for confidence bands.   |

**Value**

An object of class 'spaes', which is a list of unevaluated expressions that define how data is mapped to the visual properties of a chart.

**Examples**

```
library(SveltePlots)

sp(mtcars, spaes(x = disp, y = mpg, group = vs), type = "points")
```

---

|               |   |
|---------------|---|
| sp_add_arrows | <i>Add Arrows to a SveltePlot Chart</i> |
|---------------|---|

---

**Description**

Adds arrows to a SveltePlot chart to illustrate directions, trends, or to point out specific data points. This function enhances the chart's ability to convey insights to the viewer.

**Usage**

```
sp_add_arrows(
  sp,
  x_start,
  x_end,
  y_start,
  y_end,
  arrow_head_type = NULL,
  size = NULL,
  color = "black",
  curvature = 1e-05,
  direction = "upward",
  arrow_head = NULL
)
```

**Arguments**

|                 |  |
|-----------------|--|
| sp              | A SveltePlot htmlwidget object to which arrows will be added.  |
| x_start         | Numeric vector specifying the starting x-coordinates of the arrows.  |
| x_end           | Numeric vector specifying the ending x-coordinates of the arrows.  |
| y_start         | Numeric vector specifying the starting y-coordinates of the arrows.  |
| y_end           | Numeric vector specifying the ending y-coordinates of the arrows.  |
| arrow_head_type | Character vector specifying the type of arrow head. Can be customized to suit different visualization needs. |

|            |   |
|------------|---|
| size       | Numeric vector specifying the size of the arrows.   |
| color      | Character vector specifying the color of the arrows. Default is "black".  |
| curvature  | Numeric vector specifying the curvature of the arrows. This is useful for creating curved arrows that can more naturally point between two points on the chart. |
| direction  | Character vector specifying the direction of the arrow. Valid options are "upward" or "downward". Default is "upward".  |
| arrow_head | Also not sure what it does.   |

### Value

An object of class `htmlwidget` representing the plot with arrows added.

### Examples

```
library(SveltePlots)
data("purchases")

sp(
  data = purchases, type = "line",
  mapping = spaes(x = date, y = revenue_roll, group = age),
  colors = c("red", "green", "blue"),
  combine_same_groups = TRUE
) |>
  sp_add_series(
    data = purchases,
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    alpha = 0.4,
    tooltip = FALSE,
    include_legend = FALSE
  ) |>
  sp_add_series(
    data = purchases[purchases$revenue == max(purchases$revenue), ],
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    size = 5,
    tooltip = FALSE
  ) |>
  sp_add_segments(
    x_start = "2000-01-12", x_end = "2000-01-17",
    y_start = "auto", y_end = "auto",
    type = "rect",
    opacity = 0.2,
    background_color = "black",
    text_color = "white",
    show_legend = TRUE,
    legend_text = "Highest Revenue Day",
    tooltip = "Revenue: <strong>$13179</strong>"
  ) |>
  sp_add_arrows(
    x_start = c("2000-03-01", "2000-03-01"), x_end = c("2000-01-15", "2000-01-15"),
```

```

    y_start = c(8000, 12000), y_end = c(10000, 13000),
    arrow_head = c(0, 0),
    size = c(200, 200),
    curvature = c(0.2, 0.4),
    direction = c("downward", "downward"),
    color = c("black", "black"),
    arrow_head_type = c("triangle", "triangle")
  ) |>
  sp_add_text(
    x = c("2000-02-01", "2000-02-20"),
    y = c(12500, 8500),
    text = c(
      "This was the highest revenue day",
      "Window of Some Event Happening"
    )
  )
)

```

---

 sp\_add\_segments

*Add Segments or Rectangles to a SveltePlot Chart*


---

### Description

Adds segments or rectangles to highlight specific areas or differences within a SveltePlot chart. This function can be used to draw attention to certain data points, ranges, or to compare groups.

### Usage

```

sp_add_segments(
  sp,
  x_start,
  x_end,
  y_start = "even",
  y_end = "even",
  type = "lines",
  linetype = "solid",
  line_width = 1,
  opacity = 0.2,
  show_legend = TRUE,
  background_color = NULL,
  legend_text = " ",
  tooltip = "",
  font_size = 12,
  text_color = "black",
  x_position = NULL,
  y_position = NULL,
  outline_width = 1,
  outline_color = "black",
  key = NULL
)

```

**Arguments**

|                  |  |
|------------------|--|
| sp               | A SveltePlot htmlwidget object to which segments or rectangles will be added.  |
| x_start          | Vector of starting x positions for segments or rectangles. If the x-axis is numeric, this should be a numeric vector; if the x-axis is date or time, this should be a character vector representing dates.                       |
| x_end            | Vector of ending x positions for segments or rectangles, similar in type to x_start.   |
| y_start          | Vector of starting y positions for segments or rectangles. Can be numeric or "auto" to span the entire y-axis range.   |
| y_end            | Vector of ending y positions for segments or rectangles, similar in type to y_start.   |
| type             | Character vector specifying the type of annotation to add: "lines" for line segments or "rect" for rectangles. Default is "lines".   |
| linetype         | Character vector specifying the appearance of the line if type is "lines". Supported values include "blank", "solid", "dashed", "dotted", "dotdash", "longdash", and "twodash". Custom linetypes can also be defined as strings. |
| line_width       | Numeric vector specifying the width of lines if type is "lines". Default is 1.   |
| opacity          | Numeric vector between 0 and 1 specifying the opacity of the lines or rectangles. Default is 0.5.  |
| show_legend      | Logical indicating whether to include these segments or rectangles in the chart's legend. Default is TRUE.   |
| background_color | Character vector specifying the color(s) for the lines or rectangles. If NULL, a default color scheme is used.   |
| legend_text      | Character vector specifying custom text for legend entries. Default is NULL, and no legend will be shown.  |
| tooltip          | Character vector specifying tooltip text to be displayed on hover. Each segment or rectangle can have its own tooltip text.  |
| font_size        | Numeric vector specifying the font size of the tooltip text. Default is 12.  |
| text_color       | Character vector specifying the color of the tooltip text. Default is "black".   |
| x_position       | Vector of x positions for the tooltips. If NULL, defaults to x_start.  |
| y_position       | Vector of y positions for the tooltips. Default is NULL.   |
| outline_width    | Numeric vector specifying the width of the outline of the rectangles. Default is 1.  |
| outline_color    | Character vector specifying the color of the outline of the rectangles. Default is "black".  |
| key              | Character vector specifying keys for the segments or rectangles. Default assigns the keys from 1 to the number of rows in the data set. For more information see <a href="#">Each keyed block</a>                                |

**Value**

An object of class htmlwidget representing the plot with segments added.

**Examples**

```

library(SveltePlots)
data("segments")
data("dau")
data("purchases")

sp(
  data = purchases, type = "line",
  mapping = spaes(x = date, y = revenue_roll, group = age),
  colors = c("red", "green", "blue"),
  combine_same_groups = TRUE,
  height = 500
) |>
  sp_add_series(
    data = purchases,
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    alpha = 0.4,
    tooltip = FALSE,
    include_legend = FALSE
  ) |>
  sp_add_series(
    data = purchases[purchases$revenue == max(purchases$revenue), ],
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    size = 5,
    tooltip = FALSE
  ) |>
  sp_add_segments(
    x_start = "2000-01-12", x_end = "2000-01-17",
    y_start = "auto", y_end = "auto",
    type = "rect",
    opacity = 0.2,
    background_color = "black",
    text_color = "white",
    show_legend = TRUE,
    legend_text = "Highest Revenue Day",
    tooltip = "Revenue: <strong>$13179</strong>"
  ) |>
  sp_add_arrows(
    x_start = c("2000-03-01", "2000-03-01"), x_end = c("2000-01-15", "2000-01-15"),
    y_start = c(8000, 12000), y_end = c(10000, 13000),
    arrow_head = c(0, 0),
    size = c(200, 200),
    curvature = c(0.2, 0.4),
    direction = c("downward", "downward"),
    color = c("black", "black"),
    arrow_head_type = c("triangle", "triangle")
  ) |>
  sp_add_text(
    x = c("2000-02-01", "2000-02-20"),
    y = c(12500, 8500),

```



```
    text = c(
      "This was the highest revenue day",
      "Window of Some Event Happening"
    )
  )
)
```

```
sp <- sp(
  data = dau,
  type = "line",
  spaes(x = date, y = DAU),
  tooltip = FALSE
) |>
sp_add_series(
  data = dau,
  mapping = spaes(x = date, y = DAU),
  type = "points",
  size = 4,
  tooltip = TRUE,
) |>
sp_add_segments(
  x_start = segments$start_date,
  x_end = segments$end_date,
  y_start = "even",
  y_end = "even",
  type = "rect",
  opacity = 0.2,
  background_color = segments$colors,
  text_color = "white",
  show_legend = TRUE,
  legend_text = segments$event_type,
  tooltip = unlist(segments$extra_details),
  key = segments$key
) |>
sp_title("DAU", font_size = 24) |>
sp_x_axis(rotation_axis_ticks = -30)
```

sp

**Description**

This function adds additional series to an existing SveltePlot chart. It supports adding lines or points with customizable aesthetics such as color, size, and opacity. This is particularly useful for layering multiple data sets on a single plot for comparison or highlighting relationships.

**Usage**

```
sp_add_series(
  sp,
  data,
  mapping,
  type,
  alpha = 1,
  size = 2,
  colors = NULL,
  tooltip = TRUE,
  include_legend = TRUE,
  second_axis = FALSE
)
```

**Arguments**

|                |   |
|----------------|---|
| sp             | A SveltePlot htmlwidget object to which the series will be added. This is typically the output from a previous call to <code>sp</code> or <code>sp_add_series</code> .                |
| data           | A data frame containing the data to be added as a series to the chart.  |
| mapping        | A list of aesthetic mappings created by <code>spaes</code> . Each call to <code>sp_add_series</code> requires its own set of mappings to correctly display the data.                  |
| type           | A character string specifying the type of series to add. Valid options are "points" for scatter plots, "line" for line charts, or "bands" for confidence intervals or other purposes. |
| alpha          | A numeric value between 0 and 1 specifying the opacity of the series. Default is 1 (fully opaque).  |
| size           | A positive numeric value determining the size of the points or thickness of the line. Default is 2.   |
| colors         | A character vector of colors to use for the series. If NULL (the default), a default color scheme is applied.   |
| tooltip        | A logical value indicating whether tooltips should be shown on hover. Default is TRUE.  |
| include_legend | A logical value indicating whether a legend entry should be added for the series. Default is TRUE.  |
| second_axis    | A logical value indicating if the series should be plotted on a secondary y-axis on the right side. Values will be scaled by default to the domain of the first y-axis.               |

**Value**

An object of class `htmlwidget` representing the plot with a series added.

**Examples**

```

library(SveltePlots)
library(dplyr)
library(lubridate)

data("economics")
data("confidence_intervals")
data("purchases")

sp(
  data = economics,
  type = "line",
  mapping = spaes(x = date, y = unemploy),
  colors = "red"
) %>%
  sp_add_series(
    data = economics,
    mapping = spaes(x = date, y = pce),
    type = "line",
    colors = "green"
  ) %>%
  sp_add_series(
    data = economics,
    mapping = spaes(x = date, y = psavert),
    type = "line",
    colors = "blue"
  )

data("gapminder")

gapminder <- gapminder %>%
  dplyr::mutate(
    country = as.character(country),
    year = lubridate::ymd(paste0(year, "-01-01"))
  )

sp <- SveltePlots::sp(
  data = gapminder %>%
    dplyr::group_by(year, continent) %>%
    dplyr::summarise(
      lifeExp = mean(lifeExp)
    ) %>%
    dplyr::ungroup(),
  mapping = spaes(x = year, y = lifeExp, group = continent),
  type = "line",
  combine_same_groups = FALSE
) %>%
  sp_add_series(
    data = gapminder %>%
      dplyr::filter(country == "Germany"),
    mapping = spaes(x = year, y = lifeExp, group = country),
    type = "line",
  )

```

```

    colors = "gold"
  ) %>%
  sp_add_series(
    gapminder %>%
      dplyr::filter(country == "Chile"),
    mapping = spaes(x = year, y = lifeExp, group = country),
    type = "line",
    colors = "silver"
  ) %>%
  sp_add_series(
    gapminder %>%
      dplyr::filter(country == "Chile"),
    mapping = spaes(x = year, y = lifeExp, group = country),
    type = "points",
    size = 3,
    tooltip = FALSE
  )

sp(
  data = purchases,
  mapping = spaes(x = date, y = revenue_roll, group = age),
  type = "line",
  colors = c("red", "green", "blue"),
  combine_same_groups = FALSE
) %>%
  sp_add_series(
    data = purchases,
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    alpha = 0.4,
    tooltip = FALSE,
  ) %>%
  sp_add_series(
    data = purchases[purchases$revenue == max(purchases$revenue), ],
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    size = 5,
    tooltip = FALSE
  )

```

---

sp\_add\_text

*Add Text Annotations to a SveltePlot Chart*


---

### Description

Adds text annotations at specified positions on a SveltePlot chart. This function can be used to label specific parts of a chart, display values, or add any other textual information.

**Usage**

```
sp_add_text(
  sp,
  x,
  y,
  text,
  color = "black",
  font_size = 12,
  text_anchor = NULL,
  style = NULL
)
```

**Arguments**

|             |  |
|-------------|--|
| sp          | A SveltePlot htmlwidget object to which text annotations will be added.  |
| x           | Vector of x positions for the text annotations.  |
| y           | Vector of y positions for the text annotations.  |
| text        | Character vector of the text to be displayed as annotations.   |
| color       | Character vector specifying the color(s) of the annotation text. Default is "black".   |
| font_size   | Numeric vector specifying the size of the text. Default is 12.   |
| text_anchor | Character vector specifying the text alignment relative to its (x, y) position. Valid options are "start", "middle", or "end". |
| style       | Optional CSS style string to apply to the text.  |

**Value**

An object of class htmlwidget representing the plot with text annotations.

**Examples**

```
library(SveltePlots)
data("purchases")

sp(
  data = purchases, type = "line",
  mapping = spaes(x = date, y = revenue_roll, group = age),
  colors = c("red", "green", "blue"),
  combine_same_groups = TRUE
) |>
  sp_add_series(
    data = purchases,
    mapping = spaes(x = date, y = revenue, group = age),
    type = "points",
    alpha = 0.4,
    tooltip = FALSE,
    include_legend = FALSE
  ) |>
  sp_add_series(
```

```

data = purchases[purchases$revenue == max(purchases$revenue), ],
mapping = spaes(x = date, y = revenue, group = age),
type = "points",
size = 5,
tooltip = FALSE
) |>
sp_add_segments(
  x_start = "2000-01-12", x_end = "2000-01-17",
  y_start = "auto", y_end = "auto",
  type = "rect",
  opacity = 0.2,
  background_color = "black",
  text_color = "white",
  show_legend = TRUE,
  legend_text = "Highest Revenue Day",
  tooltip = "Revenue: <strong>$13179</strong>"
) |>
sp_add_arrows(
  x_start = c("2000-03-01", "2000-03-01"), x_end = c("2000-01-15", "2000-01-15"),
  y_start = c(8000, 12000), y_end = c(10000, 13000),
  arrow_head = c(0, 0),
  size = c(200, 200),
  curvature = c(0.2, 0.4),
  direction = c("downward", "downward"),
  color = c("black", "black"),
  arrow_head_type = c("triangle", "triangle")
) |>
sp_add_text(
  x = c("2000-02-01", "2000-02-20"),
  y = c(12500, 8500),
  text = c(
    "This was the highest revenue day",
    "Window of Some Event Happening"
  )
)
)

```

---

sp\_facet

*Create Multiple Charts Based on a Faceting Variable*


---

## Description

This function creates multiple charts based on a specified faceting variable.

## Usage

```
sp_facet(sp, ncol = NULL, nrow = NULL, scales = "fixed")
```

**Arguments**

|        |   |
|--------|---|
| sp     | A SveltePlot htmlwidget object.   |
| ncol   | The number of columns in the facet grid. Default is NULL, which auto-calculates based on the number of rows.        |
| nrow   | The number of rows in the facet grid. Default is NULL, which auto-calculates based on the number of columns.        |
| scales | A character string specifying whether scales are shared across all facets. Options are "fixed" (default) or "free". |

**Details**

This function splits the data by the specified facet variable and creates multiple charts (facets) accordingly. The function can automatically determine the number of rows and columns in the facet grid if not specified.

**Value**

An object of class `htmlwidget` representing the faceted plot.

**Examples**

```
library(SveltePlots)
data("penguins")
sp <- sp(
  data = penguins,
  mapping = spaes(x = flipper_length_mm, y = bill_length_mm, group = species),
  type = "points",
  facet_var = "sex"
) |>
  SveltePlots::sp_facet(ncol = 2, scales = "free")
sp
```

---

sp\_title

*Set title for SveltePlots charts*

---

**Description**

This function sets the title properties for SveltePlots charts, including text alignment, color, font size, font weight, and padding.

**Usage**

```
sp_title(
  sp,
  title = NULL,
  text_align = "left",
  color = "black",
```

```

    font_size = 16,
    font_weight = "bold",
    custom_css = ""
  )

```

### Arguments

|             |   |
|-------------|---|
| sp          | The SveltePlots object to modify.   |
| title       | The title text to be displayed.   |
| text_align  | Text alignment of the title ("left", "center", "right"; default: "left"). |
| color       | Color of the title text (default: "black").                               |
| font_size   | Font size of the title text (default: 16).                                |
| font_weight | Font weight of the title text (default: "bold").                          |
| custom_css  | Some custom css for the title.  |

### Value

An object of class `htmlwidget` representing the plot with an added title.

---

|            |  |
|------------|--|
| sp_tooltip | <i>Create a tooltip for SveltePlots charts</i> |
|------------|--|

---

### Description

This function creates a customizable tooltip for SveltePlots charts. Tooltips provide additional information when hovering over data points.

### Usage

```

sp_tooltip(
  sp,
  type = NULL,
  format = NULL,
  background_color = "white",
  opacity = 0.8,
  text_color = "black",
  border_color = NULL,
  border_width = 1,
  font_size = 12,
  font_family = "Arial, sans-serif",
  padding = 5,
  position = NULL,
  show_delay = 0,
  animation = FALSE,
  animation_params = NULL,
  justify_content = "space-between",
  cross_hair = FALSE
)

```



**Arguments**

|                  |   |
|------------------|---|
| sp               | The SveltePlots object to attach the tooltip to.  |
| type             | The type of tooltip, "shared" or "single". Default depends on the x-axis and is shared for date and factors and single for numeric.   |
| format           | The format of tooltip content.  |
| background_color | Background color of the tooltip (default: "white").   |
| opacity          | Numeric value between 0 and 1 specifying the opacity of the tooltip (default: 0.8).   |
| text_color       | Text color of the tooltip (default: "black").   |
| border_color     | Border color of the tooltip (default: "#cccccc").   |
| border_width     | Border width of the tooltip (default: 1).   |
| font_size        | Font size of the tooltip text (default: 12).  |
| font_family      | Font family of the tooltip text (default: "Arial, sans-serif").   |
| padding          | Padding around the tooltip content (default: 5).  |
| position         | Position of the tooltip relative to the data point ("top", "bottom", "left", "right").  |
| show_delay       | Delay in milliseconds before showing the tooltip (default: 0).  |
| animation        | Whether to animate the tooltip (default: FALSE).  |
| animation_params | A list containing animation parameters for multiple series when type = "shared":<br><b>duration</b> The duration of the animation in milliseconds. Default is 0.<br><b>delay</b> The delay before the animation starts in milliseconds. Default is 0. |
| justify_content  | How to justify the content inside the tooltip (default: "space-between").   |
| cross_hair       | Whether to enable crosshair (default: FALSE).   |

**Value**

A SveltePlots object with an attached tooltip.

---

sp\_x\_axis

---

*Modify X-Axis of a SveltePlot Chart*


---

**Description**

Modifies labels, scales, and appearance of the x-axis in a SveltePlot chart. This function allows for customization of the x-axis, including label formatting, scale type (linear or logarithmic), tick marks, and more.

**Usage**

```

sp_x_axis(
  sp,
  title = NULL,
  format = NULL,
  scale = "linear",
  ticks = 6,
  label = NULL,
  font_size_label = 14,
  font_size_ticks = 12,
  rotation_axis_ticks = 0,
  show_bar_labels = FALSE,
  position = "top",
  color_ticks = "black",
  color_label = "black",
  dx = 0,
  dy = 0,
  text_anchor = NULL,
  dominant_baseline = NULL,
  trigger = NULL,
  linetype = "solid"
)

```

**Arguments**

|                     |   |
|---------------------|---|
| sp                  | A SveltePlot htmlwidget object.   |
| title               | Character string specifying the title of the x-axis. Default is NULL.   |
| format              | Character string specifying the format of the x-axis labels. This should correspond to valid D3 format strings. Default is NULL. Documentation for the formats are here <a href="https://d3js.org/d3-format">https://d3js.org/d3-format</a> . |
| scale               | Character string indicating the scale type of the axis. Can be either "linear" or "log" for logarithmic. Default is "linear".   |
| ticks               | Numeric value indicating the suggested number of tick marks. D3 will ultimately decide the exact number of ticks based on this suggestion. Default is 6.  |
| label               | Character string for the label of the x-axis. Default is NULL.  |
| font_size_label     | Numeric value specifying the font size of the x-axis label. Default is 14.  |
| font_size_ticks     | Numeric value specifying the font size of the tick labels on the x-axis. Default is 12.   |
| rotation_axis_ticks | Numeric value indicating the rotation angle (in degrees) of the x-axis tick labels. Default is 0.   |
| show_bar_labels     | Logical indicating whether to show labels on bars for bar charts. Default is FALSE.   |

|                   |   |
|-------------------|---|
| position          | Character string specifying the position of bar labels. Can be either "top" or "middle". Default is "top".  |
| color_ticks       | Character string specifying the color of the tick labels on the x-axis. Default is "black".   |
| color_label       | Character string specifying the color of the x-axis label. Default is "black".  |
| dx                | Numeric shift along the x-axis for the x-axis label positioning. Default is 0.  |
| dy                | Numeric shift along the y-axis for the x-axis label positioning. Default is 0.  |
| text_anchor       | Character string specifying the text-anchor attribute for the x-axis labels and ticks. Can be "start", "middle", or "end". Default is NULL.       |
| dominant_baseline | Character string specifying the dominant-baseline attribute for the x-axis labels and ticks. Default is NULL.                                     |
| trigger           | Character string specifying the trigger type for tooltips. Can be "axis" or "single". Default is NULL and trigger is chosen based on x-axis type. |
| linetype          | Character string specifying the line type for the axis trigger. Can be "solid", "dashed", etc. Default is "solid".                                |

### Value

An object of class `htmlwidget` representing the plot with modified x-axis.

### Examples

```
library(SveltePlots)
sp(
  data = economics,
  mapping = spaes(x = date, y = unemploy),
  type = "line",
  tooltip = TRUE,
  colors = "red"
) |>
  sp_add_series(
    data = economics,
    mapping = spaes(x = date, y = pce),
    type = "line",
    tooltip = TRUE,
    colors = "green"
  ) |>
  sp_add_series(
    data = economics,
    mapping = spaes(x = date, y = psavert),
    type = "line",
    tooltip = FALSE,
    colors = "blue"
  ) |>
  sp_x_axis(
    format = "%b %Y",
    ticks = 4,
    label = "Date",
```

```

    font_size_label = 14,
    font_size_ticks = 12,
    rotation_axis_ticks = -30
  ) |>
  sp_y_axis(
    font_size_label = 14,
    font_size_ticks = 12
  )

```

---

sp\_y\_axis

*Modify Y-Axis of a SveltePlot Chart*


---

### Description

Similar to [sp\\_x\\_axis](#), but for the y-axis. This function customizes the appearance and scaling of the y-axis, including label formatting, tick marks, and label and tick colors.

### Usage

```

sp_y_axis(
  sp,
  format = NULL,
  scale = "linear",
  ticks = 6,
  label = NULL,
  font_size_label = 14,
  font_size_ticks = 12,
  color_ticks = "black",
  color_label = "black"
)

```

### Arguments

|                 |   |
|-----------------|---|
| sp              | A SveltePlot htmlwidget object.   |
| format          | Character string specifying the format of the x-axis labels. This should correspond to valid D3 format strings. Default is NULL. Documentation for the formats are here <a href="https://d3js.org/d3-format">https://d3js.org/d3-format</a> . |
| scale           | Character string indicating the scale type of the axis. Can be either "linear" or "log" for logarithmic. Default is "linear".   |
| ticks           | Numeric value indicating the suggested number of tick marks. D3 will ultimately decide the exact number of ticks based on this suggestion. Default is 6.  |
| label           | Character string for the label of the x-axis. Default is NULL.  |
| font_size_label | Numeric value specifying the font size of the x-axis label. Default is 14.  |

|                 |   |
|-----------------|---|
| font_size_ticks | Numeric value specifying the font size of the tick labels on the x-axis. Default is 12.     |
| color_ticks     | Character string specifying the color of the tick labels on the x-axis. Default is "black". |
| color_label     | Character string specifying the color of the x-axis label. Default is "black".              |

**Value**

An object of class `htmlwidget` representing the plot with modified y-axis.

**Examples**

```
## Not run:
library(SveltePlots)
# Assume `economics` data and prior `sp` and `sp_x_axis` calls
sp(...) |>
  sp_y_axis(label = "Unemployment", font_size_label = 18, font_size_ticks = 14)

## End(Not run)
```

---

SveltePlots-shiny

*Shiny bindings for SveltePlots*


---

**Description**

Output and render functions for using SveltePlots within Shiny applications and interactive Rmd documents.

**Usage**

```
SveltePlotsOutput(outputId, width = "100%", height = "400px")

renderSveltePlots(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

|               |  |
|---------------|--|
| outputId      | output variable to read from   |
| width, height | Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.         |
| expr          | An expression that generates a <code>AgeGroupFacet</code>  |
| env           | The environment in which to evaluate <code>expr</code> .   |
| quoted        | Is <code>expr</code> a quoted expression (with <code>quote()</code> )? This is useful if you want to save an expression in a variable. |

**Value**

No return value. Called for side effects in a 'shiny' app context.

No return value. Called for side effects in a 'shiny' app context.

---

walmart\_sales\_weekly *Walmart Weekly Sales Data*

---

**Description**

Weekly Sales Data

**Usage**

walmart\_sales\_weekly

**Format**

A tibble with 1,001 rows and 17 variables:

**id** Unique identifier for the combination of Store and Dept, a factor

**Store** Store number, a numeric value

**Dept** Department number, a numeric value

**Date** Date of the observation, in Date format

**Weekly\_Sales** Sales for the given department in the given store, a numeric value

**IsHoliday** Indicator of whether the week is a special holiday week, a logical value

**Type** Type of store, a character string

**Size** Size of the store, a numeric value

**Temperature** Temperature during the week, in degrees Fahrenheit, a numeric value

**Fuel\_Price** Cost of fuel in the region, a numeric value

**Markdown1** Markdown 1, a numeric value

**Markdown2** Markdown 2, a numeric value

**Markdown3** Markdown 3, a numeric value

**Markdown4** Markdown 4, a numeric value

**Markdown5** Markdown 5, a numeric value

**CPI** Consumer Price Index, a numeric value

**Unemployment** Unemployment rate, a numeric value

**Details**

A dataset containing weekly sales data for a retail company, including information on holidays, markdowns, and economic indicators.

**Examples**

```
data(walmart_sales_weekly)
head(walmart_sales_weekly)
```

# Index

## \* datasets

- CO2, [2](#)
  - confidence\_intervals, [3](#)
  - dau, [4](#)
  - economics, [4](#)
  - fruit, [5](#)
  - gapminder, [6](#)
  - penguins, [7](#)
  - purchases, [8](#)
  - quests, [8](#)
  - segments, [9](#)
  - walmart\_sales\_weekly, [30](#)
- 
- CO2, [2](#)
  - confidence\_intervals, [3](#)
- 
- dau, [4](#)
- 
- economics, [4](#)
- 
- fruit, [5](#)
- 
- gapminder, [6](#)
- 
- penguins, [7](#)
  - purchases, [8](#)
- 
- quests, [8](#)
- 
- renderSveltePlots (SveltePlots-shiny),  
[29](#)
- 
- segments, [9](#)
  - sp, [10](#), [18](#)
  - sp\_add\_arrows, [12](#)
  - sp\_add\_segments, [14](#)
  - sp\_add\_series, [17](#), [18](#)
  - sp\_add\_text, [20](#)
  - sp\_facet, [22](#)
  - sp\_title, [23](#)
  - sp\_tooltip, [24](#)
  - sp\_x\_axis, [25](#), [28](#)
  - sp\_y\_axis, [28](#)
  - spaes, [11](#), [18](#)
  - SveltePlots-shiny, [29](#)
  - SveltePlotsOutput (SveltePlots-shiny),  
[29](#)
  - walmart\_sales\_weekly, [30](#)