

Package: ROvis.plotly (via r-universe)

October 22, 2026

Title Create interactive data visualisations in Rijksoverheid style
using 'plotly'

Version 0.1.0

Description Provides accessible, keyboard-navigable 'plotly'
visualisations in Rijksoverheid style.

License EUPL (>= 1.2)

Depends R (>= 4.5.0)

Imports cli (>= 3.6.2), dplyr (>= 1.1.4), htmltools (>= 0.5.8.1),
htmlwidgets (>= 1.6.4), magrittr (>= 2.0.4), plotly (>=
4.12.0), rlang (>= 1.2.0), ROvis.utils

Suggests covr (>= 3.6.4), pkgdown (>= 2.0.9), quarto (>= 1.4),
testthat (>= 3.0.0), knitr (>= 1.50), rmarkdown (>= 2.30),
ggplot2 (>= 4.0.0), stringr (>= 1.5.2), tibble (>= 3.3.0),
tidyr (>= 1.3.1), zoo (>= 1.8-14)

Remotes rivm-syso/ROvis.utils

VignetteBuilder quarto

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/markdown TRUE

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev make libicu-dev libuv1-dev libssl-dev

Repository <https://rivm-syso.r-universe.dev>

Date/Publication 2026-10-22 08:08:17 UTC

RemoteUrl <https://github.com/rivm-syso/ROvis.plotly>

RemoteRef HEAD

RemoteSha 2b11df4969bff74efa797c56b97a34b367b65d4d

Contents

ro_ply_add_theme	2
ro_ply_cursor	3
ro_ply_dim	4
ro_ply_disable_toolbar	5
ro_ply_focus_linechart	6
ro_ply_keyboard_nav_barchart	7
ro_ply_keyboard_nav_bargraph	9
ro_ply_keyboard_nav_legend	11
ro_ply_keyboard_nav_linechart	12
ro_ply_keyboard_nav_trendline	14
ro_ply_legend_theme	15
ro_ply_standard	16
ro_ply_theme	20
ro_ply_title	24
Index	26

ro_ply_add_theme	<i>Add a Custom Theme to a Plotly Figure</i>
------------------	--

Description

Applies a specified Plotly theme (as a list) to a Plotly figure object.

Usage

```
ro_ply_add_theme(fig, theme)
```

Arguments

- fig A Plotly figure object created by `plotly::plot_ly()` or similar.
- theme A named list of layout parameters to apply as a theme (e.g., font, colors, margins).

Value

A Plotly figure object with the specified theme applied.

See Also

Other plotly: [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
## Not run:
fig <- ro_ply_add_theme(fig, theme = theme_rivm)

## End(Not run)
```

ro_ply_cursor

Set default cursor for a plotly figure

Description

Changes the mouse cursor style for a Plotly figure from the default crosshair to the standard arrow cursor.

Usage

```
ro_ply_cursor(fig)
```

Arguments

fig A Plotly figure object created by `plotly::plot_ly()` or similar.

Value

A Plotly figure object with the cursor style set to "default" (arrow).

See Also

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Other ggplotly: [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#)

Examples

```
## Not run:
fig <- plotly::plot_ly(data = mtcars, x = ~mpg, y = ~hp, type = 'scatter', mode = 'markers')
fig_cursor <- ro_ply_cursor(fig)
fig_cursor

## End(Not run)
```

ro_ply_dim	<i>Set RIVM theme dimensions for a plotly Figure</i>
------------	--

Description

Applies custom width and height to a Plotly figure, ensuring the plot displays at the specified dimensions. The function also sets the plot to autosize for responsive layouts.

Usage

```
ro_ply_dim(fig, width = 776, height = 400)
```

Arguments

fig	A Plotly figure object created by <code>plotly::plot_ly()</code> or similar.
width	Desired plot width in pixels. Default is 776.
height	Desired plot height in pixels. Default is 400.

Value

A Plotly figure object with updated sizing policy and layout.

See Also

Other plotly: `ro_ply_add_theme()`, `ro_ply_cursor()`, `ro_ply_disable_toolbar()`, `ro_ply_focus_linechart()`, `ro_ply_keyboard_nav_barchart()`, `ro_ply_keyboard_nav_bargraph()`, `ro_ply_keyboard_nav_legend()`, `ro_ply_keyboard_nav_linechart()`, `ro_ply_keyboard_nav_trendline()`, `ro_ply_legend_theme()`, `ro_ply_theme()`, `ro_ply_title()`

Examples

```
## Not run:
fig <- plotly::plot_ly(data = mtcars, x = ~mpg, y = ~hp, type = 'scatter', mode = 'markers')
fig_dimmed <- ro_ply_dim(fig, width = 800, height = 500)
fig_dimmed

## End(Not run)
```

`ro_ply_disable_toolbar`*Disable the plotly modebar (toolbar) and legend pop-up*

Description

[toegankelijkheid]

Usage

```
ro_ply_disable_toolbar(fig)
```

Arguments

`fig` A plotly object (from `plotly::plot_ly()` or `plotly::ggplotly()`).

Details

Removes the plotly modebar (toolbar) for a cleaner look and disables the pop-up for the legend, which improves accessibility for keyboard users.

Value

A plotly object without the modebar and with improved legend accessibility.

See Also

Other `ggplotly`: [ro_ply_cursor\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#)

Other `plotly`: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
## Not run:
data <- tidyr::tibble(
  expand.grid(`Age group` = c("A", "B", "C", "D", "E"), `Sex` = c("Man", "Vrouw")),
  `Number of Cases` = sample(1:50, 10, TRUE)
)
library(ggplot2)
library(plotly)
p <- ggplot(data, aes(
  x = `Number of Cases`,
  y = `Age group`,
```

```

    fill = Sex,
    text = paste0(
      "Leeftijdsgroep: <b>", `Age group`, "</b><br>",
      "Geslacht: <b>", Sex, "</b><br>",
      "Aantal cases: <b>", `Number of Cases`, "</b>"
    )
  )) +
  geom_col(position = "dodge") +
  labs(
    x = "Aantal cases",
    y = "Leeftijdsgroep",
    fill = "Geslacht"
  )
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_barchart(fig)
fig <- ro_ply_keyboard_nav_legend(fig)
fig <- ro_ply_disable_toolbar(fig)
fig

## End(Not run)

```

ro_ply_focus_linechart

Remove Focus Outline from Linechart in Plotly (Shiny)

Description

[toegankelijkheid]

Usage

```
ro_ply_focus_linechart()
```

Details

Adds custom CSS to your Shiny app to remove the outline and box-shadow that appear when lines in a Plotly linechart receive focus (via keyboard navigation). This improves the visual appearance for certain use-cases, but be aware that it may affect accessibility for keyboard users.

Value

HTML code (using `tags$head` and `tags$style`) to be included in a Shiny UI.

See Also

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
if (interactive()) {  
  ui <- fluidPage(  
    ro_ply_focus_linechart(),  
    plotlyOutput("myplot")  
  )  
  
  server <- function(input, output, session) {  
    output$myplot <- renderPlotly({  
      plot_ly(x = 1:10, y = rnorm(10), type = "scatter", mode = "lines")  
    })  
  }  
}
```

ro_ply_keyboard_nav_barchart

Improve keyboard accessibility for ggplotly and plotly horizontal bar-charts

Description

[toegankelijkheid]

Usage

```
ro_ply_keyboard_nav_barchart(fig)
```

Arguments

fig	A plotly object representing a horizontal bar chart, such as one created by <code>plotly::plot_ly()</code> or <code>plotly::ggplotly()</code> .
-----	---

Details

Enhances the accessibility of interactive horizontal bar charts created with `plotly` or `ggplotly`. This function injects custom JavaScript to make bar elements focusable and navigable via keyboard, displays accessible tooltips, and ensures screen readers can interpret the chart. The function is intended for use on `plotly` horizontal bar charts (including those created with `ggplotly`).

Keyboard navigation: users can use `Tab` to focus the first bar, and arrow keys (`Up/Down`) to move between bars. Pressing `Esc` removes focus from the chart and returns focus to the next logical element.

Value

A `plotly` object with improved accessibility features for keyboard and screen reader users.

See Also

[ro_ply_keyboard_nav_bargraph](#)

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Other ggplotly: [ro_ply_cursor\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#)

Examples

```
## Not run:
# Example data
data <- tidyr::tibble(
  expand.grid(`Age group` = c("A", "B", "C", "D", "E"),
    `Sex` = c("Man", "Vrouw")
  ),
  `Number of Cases` = sample(1:50, 10, TRUE)
)
```

```
# Basic ggplotly example
library(ggplot2)
library(plotly)
p <- ggplot(data, aes(
  x = `Number of Cases`,
  y = `Age group`,
  fill = Sex,
  text = paste0(
    "Leeftijdsgroep: <b>", `Age group`, "</b><br>",
    "Geslacht: <b>", Sex, "</b><br>",
    "Aantal cases: <b>", `Number of Cases`, "</b>"
  )
)) +
  geom_col(position = "dodge") +
  labs(
    x = "Aantal cases",
    y = "Leeftijdsgroep",
    fill = "Geslacht"
  ) +
  theme_minimal()
```

```
# Convert to plotly and add accessibility
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_barchart(fig)
fig
```

```
# Basic plotly example
library(plotly)
fig <- plot_ly() |>
  add_trace(
    data = data,
```

```

x = ~`Number of Cases`,
y = ~`Age group`,
color = ~`Sex`,
type = "bar",
orientation = "h",
hovertext = ~paste0(
  "Leeftijdsgroep: <b>", `Age group`, "</b><br>",
  "Geslacht: <b>", `Sex`, "</b><br>",
  "Aantal cases: <b>", `Number of Cases`, "</b>"
),
text = ~paste0(
  "Leeftijdsgroep: <b>", `Age group`, "</b><br>",
  "Geslacht: <b>", `Sex`, "</b><br>",
  "Aantal cases: <b>", `Number of Cases`, "</b>"
),
textposition = "none",
hoverinfo = "text"
)
fig <- ro_ply_keyboard_nav_barchart(fig)
fig

## End(Not run)

```

ro_ply_keyboard_nav_bargraph

Improve keyboard accessibility for (stacked) bar graphs in plotly/ggplotly

Description

[toegankelijkheid]

Usage

```
ro_ply_keyboard_nav_bargraph(fig)
```

Arguments

fig A plotly object (from `plotly::plot_ly()` or `plotly::ggplotly()`).

Details

Enhances the accessibility of interactive horizontal bar charts created with plotly or ggplotly. This function injects custom JavaScript to make bar elements focusable and navigable via keyboard, displays accessible tooltips, and ensures screen readers can interpret the chart. The function is intended for use on plotly bar graphs (including those created with ggplotly).

Keyboard navigation: users can use Tab to focus the first bar, and arrow keys (Up/Down/Left/Right) to move between bars. Pressing Esc removes focus from the chart and returns focus to the next logical element.

This function works for:

- Standard (vertical) bar graphs
- Stacked bar graphs
- Grouped/dodged bar graphs
- Proportional stacked bar graphs (bars with proportions)

Not suitable for:

- Horizontal barcharts

Value

A plotly object with improved keyboard accessibility.

See Also

[ro_ply_keyboard_nav_barchart](#)

Other ggplotly: [ro_ply_cursor\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#)

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
## Not run:
library(ggplot2)
library(plotly)
data <- tidyr::tibble(
  Sex = factor(c("Vrouw", "Man"), levels = c("Vrouw", "Man")),
  n = sample(1e6:5e6, 2)
)
p <- ggplot(data, aes(x = Sex, y = n, fill = Sex,
  text = paste0(
    "Geslacht: <b>", Sex, "</b>",
    "</b><br>Aantal cases: <b>",
    format(round(n, 0),
    big.mark = ".", decimal.mark = ",",
    scientific = FALSE), "</b>"
  ))) +
  geom_col(width = 0.6) +
  labs(x = "Geslacht", y = "Aantal cases") +
  theme_minimal()
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_bargraph(fig)
```

```
fig
## End(Not run)
```

ro_ply_keyboard_nav_legend

Improve keyboard accessibility for plotly legends

Description

[toegankelijkheid]

Usage

```
ro_ply_keyboard_nav_legend(fig)
```

Arguments

fig A plotly object (from `plotly::plot_ly()` or `plotly::ggplotly()`).

Details

Makes the legend of a plotly or ggplotly plot accessible via keyboard. Users can use Tab to focus legend items and activate them with Enter or Space. Focus indication is only shown when navigating with the keyboard. ARIA attributes are added for screen readers.

Value

A plotly object with improved legend accessibility.

See Also

Other ggplotly: [ro_ply_cursor\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#)

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
## Not run:
data <- tidyr::tibble(
  expand.grid(`Age group` = c("A", "B", "C", "D", "E"), `Sex` = c("Man", "Vrouw")),
  `Number of Cases` = sample(1:50, 10, TRUE)
)
library(ggplot2)
library(plotly)
p <- ggplot(data, aes(
  x = `Number of Cases`,
  y = `Age group`,
  fill = Sex,
  text = paste0(
    "Leeftijdsgroep: <b>", `Age group`, "</b><br>",
    "Geslacht: <b>", Sex, "</b><br>",
    "Aantal cases: <b>", `Number of Cases`, "</b>"
  )
)) +
  geom_col(position = "dodge") +
  labs(
    x = "Aantal cases",
    y = "Leeftijdsgroep",
    fill = "Geslacht"
  )
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_barchart(fig)
fig <- ro_ply_keyboard_nav_legend(fig)
fig

## End(Not run)
```

ro_ply_keyboard_nav_linechart

Improve keyboard accessibility for line charts in plotly/ggplotly

Description

[toegankelijkheid]

Usage

```
ro_ply_keyboard_nav_linechart(fig)
```

Arguments

fig A plotly object (from `plotly::plot_ly()` or `plotly::ggplotly()`).

Details

Enhances the accessibility of interactive line charts created with `plotly` or `ggplotly`. This function injects custom JavaScript to make bar elements focusable and navigable via keyboard, displays accessible tooltips, and ensures screen readers can interpret the chart. The function is intended for use on `plotly` line charts (including those created with `ggplotly`).

Keyboard navigation: users can use `Tab` to focus the first bar, and arrow keys (`Left/Right/Up/Down`) to move between points on the line. Pressing `Esc` removes focus from the chart and returns focus to the next logical element.

Value

A `plotly` object with improved keyboard accessibility for line charts.

See Also

Other `ggplotly`: `ro_ply_cursor()`, `ro_ply_disable_toolbar()`, `ro_ply_keyboard_nav_barchart()`, `ro_ply_keyboard_nav_bargraph()`, `ro_ply_keyboard_nav_legend()`, `ro_ply_keyboard_nav_trendline()`

Other `plotly`: `ro_ply_add_theme()`, `ro_ply_cursor()`, `ro_ply_dim()`, `ro_ply_disable_toolbar()`, `ro_ply_focus_linechart()`, `ro_ply_keyboard_nav_barchart()`, `ro_ply_keyboard_nav_bargraph()`, `ro_ply_keyboard_nav_legend()`, `ro_ply_keyboard_nav_trendline()`, `ro_ply_legend_theme()`, `ro_ply_theme()`, `ro_ply_title()`

Examples

```
## Not run:
data <- tidyr::tibble(expand.grid(Jaar = 2001:2020, Geslacht = c("Man", "Vrouw")),
  value = sample(1:50, 40, TRUE)) |>
  dplyr::bind_rows(., dplyr::summarize(., value = sum(value), .by = "Jaar") |>
    dplyr::mutate(Geslacht = "Totaal")) |>
  dplyr::mutate(Geslacht = factor(Geslacht, levels = c("Man", "Vrouw", "Totaal")))
library(ggplot2)
library(plotly)
p <- ggplot(data, aes(x = Jaar, y = value, color = Geslacht, group = Geslacht,
  text = paste0(
    "Geslacht: <b>", Geslacht,
    "</b><br>Jaar: <b>", Jaar,
    "</b><br>Aantal cases: <b>", value
  ))) +
  geom_line()
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_linechart(fig)
fig

## End(Not run)
```

ro_ply_keyboard_nav_trendline

Improve keyboard accessibility for scatter-trendline plots in plotly/ggplotly

Description

[toegankelijkheid]

Usage

```
ro_ply_keyboard_nav_trendline(fig)
```

Arguments

fig A plotly object (from `plotly::plot_ly()` or `plotly::ggplotly()`).

Details

Enhances the accessibility of interactive scatter-trendline plots created with plotly or ggplotly. This function injects custom JavaScript to make bar elements focusable and navigable via keyboard, displays accessible tooltips, and ensures screen readers can interpret the chart. The function is intended for use on plotly scatter-trendline plots (including those created with ggplotly). The focusable element are the points.

Keyboard navigation: users can use Tab to focus the first point, and arrow keys (Left/Right/Up/Down) to move between points. Pressing Esc removes focus from the chart and returns focus to the next logical element.

Value

A plotly object with improved keyboard accessibility.

See Also

Other ggplotly: [ro_ply_cursor\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#)

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#), [ro_ply_title\(\)](#)

Examples

```
## Not run:
dates <- seq(as.Date("2021-10-04"), by = "day", length.out = 100)
n <- round(2500 + 700 * sin(seq_along(dates) / 25) + seq_along(dates) *
3 + rnorm(length(dates), 0, 150))

moving_average <- stats::filter(n, rep(1 / 7, 7), sides = 2)
text <- paste0(
  "Dag: <b>", format(dates, "%d-%m-%Y"), "</b><br>",
  "Aantal cases: <b>", format(n,
    big.mark = ".", decimal.mark = ",", scientific = FALSE), "</b><br>",
  "7-daags gemiddelde: <b>",
    format(round(moving_average, 0),
      big.mark = ".",
      decimal.mark = ",", scientific = FALSE), "</b>"
)
data <- tibble::tibble(
  Date_statistics = dates,
  n = n,
  moving_average = as.numeric(moving_average),
  text = text
)
library(ggplot2)
library(plotly)
p <- ggplot(data, aes(x = Date_statistics, y = n, text = text, group = 1)) +
  geom_line() +
  geom_point()
fig <- ggplotly(p, tooltip = "text")
fig <- ro_ply_keyboard_nav_trendline(fig)
fig

## End(Not run)
```

ro_ply_legend_theme *RIVM Legend Styling for plotly figures*

Description

Applies advanced custom styling to the legend of a Plotly figure. This function adds a box around each individual legend item and changes legend symbols to rectangles. The modifications are applied using JavaScript via `htmlwidgets::onRender()`.

Usage

```
ro_ply_legend_theme(fig)
```

Arguments

fig A Plotly figure object created by `plotly::plot_ly()` or similar.

Value

The Plotly figure object with the custom legend styling applied.

See Also

Other plotly: `ro_ply_add_theme()`, `ro_ply_cursor()`, `ro_ply_dim()`, `ro_ply_disable_toolbar()`, `ro_ply_focus_linechart()`, `ro_ply_keyboard_nav_barchart()`, `ro_ply_keyboard_nav_bargraph()`, `ro_ply_keyboard_nav_legend()`, `ro_ply_keyboard_nav_linechart()`, `ro_ply_keyboard_nav_trendline()`, `ro_ply_theme()`, `ro_ply_title()`

Examples

```
## Not run:
fig <- plotly::plot_ly(data = mtcars, x = ~mpg, y = ~hp,
  color = ~factor(cyl), type = 'scatter', mode = 'markers')
fig_styled <- ro_ply_legend_theme(fig)
fig_styled

## End(Not run)
```

ro_ply_standard

Apply styling and accessibility features to plotly figures

Description

This function applies a standardized RIVM theme and accessibility features to a given plotly figure object. It wraps several (internal) ROvis functions.

Usage

```
ro_ply_standard(
  fig,
  type = c("none", "barchart", "bargraph", "linechart", "trendline"),
  barmode = "vgroup",
  base_family = "RijksoverheidSansWebText",
  title_color = ro_color(color_name = "lintblauw"),
  title_size = 17,
  y_axis_title_size = 13,
  x_axis_title_size = 13,
  axis_tick_size = 12,
  axis_title_color = ro_color(color_name = "lintblauw"),
  axis_tick_color = ro_color(color_name = "grijs_7"),
  grid_color = ro_color(color_name = "grijs_3"),
  grid_width = 0.25,
  y_axis_showgrid = TRUE,
  x_axis_showgrid = FALSE,
  x_axis_tick_format = NULL,
```

```

y_axis_tick_format = ",.0f",
y_axis_showline = FALSE,
x_axis_showline = TRUE,
x_line_color = ro_color(color_name = "grijs_7"),
y_line_color = ro_color(color_name = "lintblauw"),
x_line_width = 1.5,
y_line_width = 0,
y_axis_spike = FALSE,
x_axis_spike = TRUE,
cross_hair_color = "#535353",
cross_hair_style = "dash",
cross_hair_width = 1,
colorway = c(ro_color(color_name = "hemelblauw"), ro_color(color_name = "donkergeel"),
  ro_color(color_name = "robijnrood"), ro_color(color_name = "paars_tint90"),
  ro_color(color_name = "mintgroen_tint110"), ro_color(color_name = "oranje"),
  ro_color(color_name = "groen"), ro_color(color_name = "donkerbruin")),
legend_font_size = 13,
legend_color = ro_color(color_name = "grijs_7"),
plot_bgcolor = "#ffffff",
paper_bgcolor = "#ffffff",
tooltip_color = "#000000",
tooltip_border_color = "#000000",
tooltip_background_color = "#ffffff",
tooltip_font_size = 13,
y_data = NULL,
y_label = NULL,
x_label = NULL,
title = NULL,
canvas_width = 776,
left_margin_base = 10,
char_width = 7,
axis_title_size = 13,
thousand_seperator = ".",
decimal_seperator = ",",
width = 776,
height = 400
)

```

Arguments

fig	A plotly figure object created by <code>plotly::plot_ly()</code> to which the styling and accessibility features will be added.
type	Type of chart for which to apply keyboard navigation. Options are "none", "bar-chart", "bargraph", "linechart", or "trendline". Default is "none".
barmode	Bar mode for bar charts. "vgroup" by default.
base_family	Font family for all text (default = "RijksoverheidSansWebText").
title_color	Color for all titles/labels.

<code>title_size</code>	Font size for the main title.
<code>y_axis_title_size</code>	Font size for y-axis title.
<code>x_axis_title_size</code>	Font size for x-axis title.
<code>axis_tick_size</code>	Font size for axis tick labels.
<code>axis_title_color</code>	Color for axis titles.
<code>axis_tick_color</code>	Color for axis tick labels and axis lines.
<code>grid_color</code>	Color for grid lines.
<code>grid_width</code>	Width of grid lines.
<code>y_axis_showgrid</code>	Show grid lines for y-axis.
<code>x_axis_showgrid</code>	Show grid lines for x-axis.
<code>x_axis_tick_format</code>	Format string for x-axis tick labels (e.g., ".2f" for 2 decimal places).
<code>y_axis_tick_format</code>	Format string for y-axis tick labels (e.g., ",.0f" for thousands separator).
<code>y_axis_showline</code>	Show y-axis line.
<code>x_axis_showline</code>	Show x-axis line.
<code>x_line_color</code>	Color for the x-axis line.
<code>y_line_color</code>	Color for the y-axis line.
<code>x_line_width</code>	Width of the x-axis line.
<code>y_line_width</code>	Width of the y-axis line.
<code>y_axis_spike</code>	Show crosshair/spike on y-axis.
<code>x_axis_spike</code>	Show crosshair/spike on x-axis.
<code>cross_hair_color</code>	Color for the crosshair/spike lines.
<code>cross_hair_style</code>	Line style for crosshair (e.g., "dash", "solid").
<code>cross_hair_width</code>	Width of the crosshair lines.
<code>colorway</code>	Colors for data series.
<code>legend_font_size</code>	Font size for the legend.
<code>legend_color</code>	Color for the legend text.
<code>plot_bgcolor</code>	Plot background color.
<code>paper_bgcolor</code>	Figure (paper) background color.

tooltip_color	Color for tooltip text.
tooltip_border_color	Border color for tooltips.
tooltip_background_color	Background color for tooltips.
tooltip_font_size	Font size for tooltip text.
y_data	Used to determine the longest y-axis label for margin calculation.
y_label	Y-axis label to display.
x_label	X-axis label to display.
title	Main title to display at the top left.
canvas_width	Width of the plotting area in pixels.
left_margin_base	Base left margin in pixels.
char_width	Estimated width per character in pixels.
axis_title_size	Font size for axis titles.
thousand_seperator	Separator for thousands in y-axis labels.
decimal_seperator	Separator for decimals in y-axis labels.
width	Desired plot width in pixels. Default is 776.
height	Desired plot height in pixels. Default is 400.

Value

The Plotly figure object with the custom styling applied and accesibility features applied.

Examples

```
## Not run:
library(plotly)
library(R0vis.plotly)

# Create sample data for a grouped bar chart
data <- data.frame(
  AgeGroup = rep(c("0-19", "20-39", "40-59", "60-79", "80+"), 2),
  Sex = rep(c("Male", "Female"), each = 5),
  Cases = c(145, 203, 287, 156, 89, 132, 198, 276, 143, 95)
)

# Create the base Plotly figure with custom colors
fig <- plotly::plot_ly(
  data = data,
  x = ~AgeGroup,
  y = ~Cases,
  color = ~Sex,
```

```

    colors = c(
      "Male" = ro_color("hemelblauw"),
      "Female" = ro_color("robijnrood")
    ),
    type = "bar",
    text = ~paste0(
      "Leeftijdsgroep: <b>", AgeGroup, "</b>",
      "<br>Geslacht: <b>", Sex, "</b>",
      "<br>Aantal cases: <b>",
      format(round(Cases, 0), big.mark = ".", decimal.mark = ",", scientific = FALSE), "</b>"
    ),
    hovertext = ~paste0(
      "Leeftijdsgroep: <b>", Agegroup, "</b>",
      "<br>Geslacht: <b>", Sex, "</b>",
      "<br>Aantal cases: <b>",
      format(round(n, 0), big.mark = ".", decimal.mark = ",", scientific = FALSE), "</b>"
    )
  )
}

# Apply Rijksoverheid / RIVM styling with accessibility features for a bar graph
fig <- ro_ply_standard(
  fig,
  type = "bargraph",          # Adds keyboard navigation
  barmode = "group",          # Group bars by age group
  title = "COVID-19 Cases by Age Group and Sex",
  y_label = "Number of Cases",
  x_label = "Age Group",
)

fig

## End(Not run)

```

ro_ply_theme

RIVM Plotly Theme List

Description

Returns a standardized RIVM theme as a list for use with plotly charts, allowing customization of fonts, colors, grid, axis, legend, and tooltip styling.

Usage

```

ro_ply_theme(
  barmode = "vgroup",
  base_family = "RijksoverheidSansWebText",
  title_color = ro_color(color_name = "lintblauw"),
  title_size = 17,
  y_axis_title_size = 13,
  x_axis_title_size = 13,

```

```

axis_tick_size = 12,
axis_title_color = ro_color(color_name = "lintblauw"),
axis_tick_color = ro_color(color_name = "grijs_7"),
grid_color = ro_color(color_name = "grijs_3"),
grid_width = 0.25,
y_axis_showgrid = TRUE,
x_axis_showgrid = FALSE,
x_axis_tick_format = NULL,
y_axis_tick_format = ",.0f",
y_axis_showline = FALSE,
x_axis_showline = TRUE,
x_line_color = ro_color(color_name = "grijs_7"),
y_line_color = ro_color(color_name = "lintblauw"),
x_line_width = 1.5,
y_line_width = 0,
y_axis_spike = FALSE,
x_axis_spike = TRUE,
cross_hair_color = "#535353",
cross_hair_style = "dash",
cross_hair_width = 1,
colorway = c(ro_color(color_name = "hemelblauw"), ro_color(color_name = "donkergeel"),
  ro_color(color_name = "robynrood"), ro_color(color_name = "paars_tint90"),
  ro_color(color_name = "mintgroen_tint110"), ro_color(color_name = "oranje"),
  ro_color(color_name = "groen"), ro_color(color_name = "donkerbruin")),
legend_font_size = 13,
legend_color = ro_color(color_name = "grijs_7"),
plot_bgcolor = "#ffffff",
paper_bgcolor = "#ffffff",
tooltip_color = "#000000",
tooltip_border_color = "#000000",
tooltip_background_color = "#ffffff",
tooltip_font_size = 13
)

```

Arguments

<code>barmode</code>	Bar mode for bar charts. "vgroup" by default.
<code>base_family</code>	Font family for all text (default = "RijksoverheidSansWebText").
<code>title_color</code>	Color for the main title text.
<code>title_size</code>	Font size for the main title.
<code>y_axis_title_size</code>	Font size for y-axis title.
<code>x_axis_title_size</code>	Font size for x-axis title.
<code>axis_tick_size</code>	Font size for axis tick labels.
<code>axis_title_color</code>	Color for axis titles.

<code>axis_tick_color</code>	Color for axis tick labels and axis lines.
<code>grid_color</code>	Color for grid lines.
<code>grid_width</code>	Width of grid lines.
<code>y_axis_showgrid</code>	Show grid lines for y-axis.
<code>x_axis_showgrid</code>	Show grid lines for x-axis.
<code>x_axis_tick_format</code>	Format string for x-axis tick labels (e.g., <code>".0f"</code> for thousands separator as comma, <code>".0%"</code> for percent, <code>".0f"</code> for no thousand separator, for more information on time-formats see https://d3js.org/d3-time-format)
<code>y_axis_tick_format</code>	Format string for y-axis tick labels (e.g., <code>".0f"</code> for thousands separator as comma, <code>".0%"</code> for percent, <code>".0f"</code> for no thousand separator, for more information on time-formats see https://d3js.org/d3-time-format)
<code>y_axis_showline</code>	Show y-axis line.
<code>x_axis_showline</code>	Show x-axis line.
<code>x_line_color</code>	Color for the x-axis line.
<code>y_line_color</code>	Color for the y-axis line.
<code>x_line_width</code>	Width of the x-axis line.
<code>y_line_width</code>	Width of the y-axis line.
<code>y_axis_spike</code>	Show crosshair/spike on y-axis.
<code>x_axis_spike</code>	Show crosshair/spike on x-axis.
<code>cross_hair_color</code>	Color for crosshair/spike lines.
<code>cross_hair_style</code>	Style for crosshair/spike lines.
<code>cross_hair_width</code>	Width for crosshair/spike lines.
<code>colorway</code>	Colors for data series.
<code>legend_font_size</code>	Font size for the legend.
<code>legend_color</code>	Color for the legend text.
<code>plot_bgcolor</code>	Plot background color.
<code>paper_bgcolor</code>	Figure (paper) background color.
<code>tooltip_color</code>	Font color for tooltips.
<code>tooltip_border_color</code>	Border color for tooltips.
<code>tooltip_background_color</code>	Background color for tooltips.
<code>tooltip_font_size</code>	Font size for tooltips.

Value

A named list with plotly theming options, suitable for passing to `plotly::layout()` or as a theme in a plotly chart.

See Also

Other plotly: `ro_ply_add_theme()`, `ro_ply_cursor()`, `ro_ply_dim()`, `ro_ply_disable_toolbar()`, `ro_ply_focus_linechart()`, `ro_ply_keyboard_nav_barchart()`, `ro_ply_keyboard_nav_bargraph()`, `ro_ply_keyboard_nav_legend()`, `ro_ply_keyboard_nav_linechart()`, `ro_ply_keyboard_nav_trendline()`, `ro_ply_legend_theme()`, `ro_ply_title()`

Examples

```
## Not run:
# Example data: number of cases per age group and sex
stacked_data <- tibble::tibble(
  Agegroup = rep(c("0-9", "10-19", "20-29", "30-39"), each = 3),
  Sex = rep(c("Man", "Vrouw", "Onbekend"), times = 4),
  n = c(10, 14, 2, 15, 17, 1, 23, 18, 0, 9, 11, 3)
)

# Define color palette using your color function
sex_colors <- c(
  "Man" = ro_color("hemelblauw"),
  "Vrouw" = ro_color("robijnrood"),
  "Onbekend" = ro_color("donkergeel")
)

# Use ro_ply_theme with barmode = "stack"
theme_rivm <- ro_ply_theme(barmode = "stack")

# Create the plotly stacked bar chart
fig <- plotly::plot_ly(
  data = stacked_data,
  x = ~Agegroup,
  y = ~n,
  type = "bar",
  color = ~Sex,
  colors = sex_colors,
  text = ~paste0(
    "Leeftijdsgroep: <b>", Agegroup, "</b>",
    "<br>Geslacht: <b>", Sex, "</b>",
    "<br>Aantal cases: <b>",
    format(round(n, 0), big.mark = ".", decimal.mark = ",", scientific = FALSE), "</b>"
  ),
  hoverinfo = "text"
)

fig <- ro_ply_add_theme(fig, theme_rivm)
fig

## End(Not run)
```

ro_ply_title

*Add Custom Titles and Axis Labels to a Plotly Figure***Description**

This function adds a (custom styled) main title, y-axis label, and/or x-axis label as annotations to a Plotly figure, using RIVM theming and Dutch number formatting. It automatically calculates left margin spacing based on the longest y-axis label.

Usage

```
ro_ply_title(
  fig,
  y_data = NULL,
  y_label = NULL,
  x_label = NULL,
  title = NULL,
  base_family = "RijksoverheidSansWebText",
  canvas_width = 776,
  left_margin_base = 10,
  char_width = 7,
  title_size = 17,
  axis_title_size = 13,
  title_color = ro_color(color_name = "lintblauw"),
  thousand_seperator = ".",
  decimal_seperator = ",",
)
```

Arguments

fig	A plotly figure object created by <code>plotly::plot_ly()</code> to which the titles and labels will be added.
y_data	Used to determine the longest y-axis label for margin calculation.
y_label	Y-axis label to display.
x_label	X-axis label to display.
title	Main title to display at the top left.
base_family	Font family to use.
canvas_width	Width of the plotting area in pixels.
left_margin_base	Base left margin in pixels.
char_width	Estimated width per character in pixels.
title_size	Font size for the main title.
axis_title_size	Font size for axis titles.

title_color Color for all titles/labels.
thousand_seperator Separator for thousands in y-axis labels.
decimal_seperator Separator for decimals in y-axis labels.

Value

The plotly figure object with added (and styled) title and/or axis labels as annotations.

See Also

Other plotly: [ro_ply_add_theme\(\)](#), [ro_ply_cursor\(\)](#), [ro_ply_dim\(\)](#), [ro_ply_disable_toolbar\(\)](#), [ro_ply_focus_linechart\(\)](#), [ro_ply_keyboard_nav_barchart\(\)](#), [ro_ply_keyboard_nav_bargraph\(\)](#), [ro_ply_keyboard_nav_legend\(\)](#), [ro_ply_keyboard_nav_linechart\(\)](#), [ro_ply_keyboard_nav_trendline\(\)](#), [ro_ply_legend_theme\(\)](#), [ro_ply_theme\(\)](#)

Examples

```
## Not run:  
# Title only:  
ro_ply_title(fig, y_data = data$'Number of Cases', title = "Fancy title")  
# Y-axis label only:  
ro_ply_title(fig, y_label = "Number of cases")  
# X-axis label only:  
ro_ply_title(fig, x_label = "Groups")  
# Title, Y-axis label, and X-axis label in one go:  
ro_ply_title(fig, y_data = data$'Number of Cases',  
  y_label = "Number of cases", title = "Fancy title", x_label = "groups")  
  
## End(Not run)
```

Index

* ggplotly

- ro_ply_cursor, 3
- ro_ply_disable_toolbar, 5
- ro_ply_keyboard_nav_barchart, 7
- ro_ply_keyboard_nav_bargraph, 9
- ro_ply_keyboard_nav_legend, 11
- ro_ply_keyboard_nav_linechart, 12
- ro_ply_keyboard_nav_trendline, 14

* plotly

- ro_ply_add_theme, 2
- ro_ply_cursor, 3
- ro_ply_dim, 4
- ro_ply_disable_toolbar, 5
- ro_ply_focus_linechart, 6
- ro_ply_keyboard_nav_barchart, 7
- ro_ply_keyboard_nav_bargraph, 9
- ro_ply_keyboard_nav_legend, 11
- ro_ply_keyboard_nav_linechart, 12
- ro_ply_keyboard_nav_trendline, 14
- ro_ply_legend_theme, 15
- ro_ply_theme, 20
- ro_ply_title, 24

- ro_ply_add_theme, 2
- ro_ply_add_theme(), 3–6, 8, 10, 11, 13, 14, 16, 23, 25
- ro_ply_cursor, 3
- ro_ply_cursor(), 2, 4–6, 8, 10, 11, 13, 14, 16, 23, 25
- ro_ply_dim, 4
- ro_ply_dim(), 2, 3, 5, 6, 8, 10, 11, 13, 14, 16, 23, 25
- ro_ply_disable_toolbar, 5
- ro_ply_disable_toolbar(), 2–4, 6, 8, 10, 11, 13, 14, 16, 23, 25
- ro_ply_focus_linechart, 6
- ro_ply_focus_linechart(), 2–5, 8, 10, 11, 13, 14, 16, 23, 25
- ro_ply_keyboard_nav_barchart, 7, 10

- ro_ply_keyboard_nav_barchart(), 2–6, 10, 11, 13, 14, 16, 23, 25
- ro_ply_keyboard_nav_bargraph, 8, 9
- ro_ply_keyboard_nav_bargraph(), 2–6, 8, 11, 13, 14, 16, 23, 25
- ro_ply_keyboard_nav_legend, 11
- ro_ply_keyboard_nav_legend(), 2–6, 8, 10, 13, 14, 16, 23, 25
- ro_ply_keyboard_nav_linechart, 12
- ro_ply_keyboard_nav_linechart(), 2–6, 8, 10, 11, 14, 16, 23, 25
- ro_ply_keyboard_nav_trendline, 14
- ro_ply_keyboard_nav_trendline(), 2–6, 8, 10, 11, 13, 16, 23, 25
- ro_ply_legend_theme, 15
- ro_ply_legend_theme(), 2–6, 8, 10, 11, 13, 14, 23, 25
- ro_ply_standard, 16
- ro_ply_theme, 20
- ro_ply_theme(), 2–6, 8, 10, 11, 13, 14, 16, 25
- ro_ply_title, 24
- ro_ply_title(), 2–6, 8, 10, 11, 13, 14, 16, 23