

Package: ROvis.ggplot2 (via r-universe)

October 19, 2026

Title Create data visualisations in Rijksoverheid style using ggplot2

Version 0.1.0

Description A tool to uniformly visualise graphs in 'ggplot2' using
standardized Rijksoverheid (Dutch National Government) styling.
Part of the ROvis umbrella package.

License EUPL (>= 1.2)

Imports ggplot2 (>= 4.0.3), grid (>= 4.5.1), scales (>= 1.4.0),
ROvis.utils, rlang, cli (>= 3.6.6)

Suggests covr (>= 3.6.4), gt (>= 1.3.0), knitr (>= 1.51), pkgdown (>= 2.0.9), quarto (>= 1.4), dplyr (>= 1.2.1), testthat (>= 3.0.0),
tidyverse (>= 2.0.0), vdiff (>= 1.0.9)

Depends R (>= 4.5.0)

Remotes rivm-syso/ROvis.utils

VignetteBuilder quarto

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(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 2026-10-13 11:58:29 UTC

RemoteUrl <https://github.com/rivm-syso/ROvis.ggplot2>

RemoteRef HEAD

RemoteSha d0baca41c7d9979cec5cc3fd87dee1ab079df172

Contents

ro_gg_axis	2
ro_gg_group_ticks	3
ro_gg_plot	4

ro_gg_remove_axis_gap	6
ro_gg_save	7
ro_gg_theme	8
ro_gg_y_title	11

Index	13
--------------	-----------

ro_gg_axis	<i>Expand Axis Limits for Plots</i>
------------	-------------------------------------

Description

Calculates axis limits for a plot that are aligned with pretty breaks, ensuring that all data points are within the range and the maximum value is comfortably included, in accordance with RIVM styling. Optionally, the lower limit can be set to zero.

Usage

```
ro_gg_axis(x, start_at_zero = TRUE)
```

Arguments

- x Numeric vector of values to be plotted.
- start_at_zero Logical; if TRUE (default), the lower limit of the axis will start at 0.

Value

A numeric vector of length 2, giving the minimum and maximum axis limits.

See Also

Other ggplot2: [ro_gg_group_ticks\(\)](#), [ro_gg_plot\(\)](#), [ro_gg_remove_axis_gap\(\)](#), [ro_gg_save\(\)](#), [ro_gg_theme\(\)](#), [ro_gg_y_title\(\)](#)

Examples

```
# Example with ggplot2
## Not run:
library(ggplot2)
data <- data.frame(x = 1:11, y = c(11:17, 5:2))

ggplot(data, aes(x, y)) +
  geom_point() +
  scale_x_continuous(
    breaks = pretty,
    limits = ro_gg_axis
  ) +
  scale_y_continuous(
    breaks = pretty,
    limits = ro_gg_axis
```

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

ro_gg_group_ticks	<i>Add RIVM-style tick marks between bar/column groups</i>
-------------------	--

Description

[Experimental] Replaces the default per-category axis ticks of a bar/column chart with tick marks placed *between* each group of bars, following RIVM design guidelines. `ggplot2` centers axis ticks under/next to each category by default; `ro_gg_group_ticks()` hides those and draws short line segments at the boundary between every pair of adjacent categories instead, plus the two outer edges.

Since the tick marks are drawn just outside the panel area, always use this function in combination with `coord_cartesian(clip = "off")`.

Usage

```
ro_gg_group_ticks(plot, flip = FALSE)
```

Arguments

<code>plot</code>	A <code>ggplot</code> object with a discrete category axis (e.g. from <code>geom_col()</code> / <code>geom_bar()</code>).
<code>flip</code>	Logical. Set to <code>TRUE</code> if the category axis is <code>y</code> rather than <code>x</code> (the same orientation you'd pass to <code>theme_ggrivm(flip = TRUE)</code>). Default = <code>FALSE</code> .

Value

A `ggplot` object

See Also

Other `ggplot2`: [ro_gg_axis\(\)](#), [ro_gg_plot\(\)](#), [ro_gg_remove_axis_gap\(\)](#), [ro_gg_save\(\)](#), [ro_gg_theme\(\)](#), [ro_gg_y_title\(\)](#)

Examples

```
## Not run:
library(ggplot2)

dat <- data.frame(agegroup = c("0-19", "20-39", "40-59"), n = c(1200, 4800, 6100))
p <- ggplot(dat, aes(agegroup, n)) +
  geom_col() +
  theme_ggrivm()

ro_gg_group_ticks(p) +
  coord_cartesian(clip = "off")

## End(Not run)
```

ro_gg_plot	<i>Apply the RO ggplot2 theme and auto-fix common RO style requirements</i>
------------	---

Description

[Experimental] `ro_gg_plot()` is a concept/draft function that explores applying the RO visual style in a single step, instead of requiring the separate calls to `ro_gg_theme()`, `ro_gg_remove_axis_gap()`, `ro_gg_axis()`, a manual thousands-separator, `ro_gg_group_ticks()`, and `ro_gg_y_title()` described in the vignette("ggplot2-in-ro-style") vignette.

Unlike `ro_gg_theme()`, which returns a `theme()` object that only carries cosmetic settings, `ro_gg_plot()` takes the *whole plot* as input and returns a modified plot. This is required because several RO style rules (axis limits/expansion, number formatting, y-axis title position) depend on the data, the scales, and the geoms already present on the plot, none of which a plain `theme()` object can see. Call `ro_gg_plot()` last, after all layers, scales, and labels have been added.

Usage

```
ro_gg_plot(
  plot,
  flip = FALSE,
  base_size = 12,
  base_family = "RijksoverheidSansWebText",
  lang = c("nl", "en"),
  auto_set_y_title = TRUE
)
```

Arguments

<code>plot</code>	A ggplot object, with all layers, scales, and labels already added.
<code>flip</code>	Logical. Passed on to <code>ro_gg_theme()</code> . Default = FALSE.
<code>base_size</code>	Integer. Passed on to <code>ro_gg_theme()</code> . Default = 12.
<code>base_family</code>	Character. Passed on to <code>ro_gg_theme()</code> . Default = "RijksoverheidSansWebText".
<code>lang</code>	Character. "nl" (default) uses a dot as thousands separator (e.g. "1.000"); "en" uses a comma (e.g. "1,000").
<code>auto_set_y_title</code>	Logical. If TRUE (default), relocates the y-axis title via <code>ro_gg_y_title()</code> , as described above. Set to FALSE to leave the y-axis title alone (<code>ro_gg_plot()</code> then only applies the cosmetic theme, legend, and axis changes) so you can position it yourself, e.g. with your own <code>ggplot2::annotation_custom()</code> call, or via <code>theme(axis.title.y = ...)</code> if the default placement isn't needed at all.

Value

A modified ggplot object.

What this does automatically

- Applies the `ro_gg_theme()` cosmetic theme.
- Removes legend titles (RO style never shows one), for fill, colour/color, shape, linetype, alpha, and size.
- For bar/column charts (any layer using `geom_col()`/`geom_bar()`), sets the group width to 80% of each category slot, removes any padding between bars within the same group (switching `position_dodge2()` to `position_dodge()` if needed), and removes bar outlines. Tick marks are then moved from under each category to the boundary between groups via `ro_gg_group_ticks()`. Works for both the default and flipped (`flip = TRUE`) orientation.
- For bar/column and line charts (any layer using `geom_col()`/`geom_bar()`/`geom_line()`), and only if the user has not already added a scale for the continuous axis, applies `ro_gg_remove_axis_gap()` expansion, `ro_gg_axis()` as the axis limits (so the continuous axis starts at 0), and a thousands separator on the labels. Other chart types (e.g. scatter plots) only get the thousands separator.
- Relocates the y-axis title to the RO position via `ro_gg_y_title()`, when a y-axis title is present. Works for both the default and flipped (`flip = TRUE`) orientation.

Known limitations

- The bar group width (80%), zero padding within a group, and tick-mark styling (color/width/length) are fixed and not user-configurable; if you need different values, apply `ro_gg_group_ticks()` and the geom parameters yourself instead of `ro_gg_plot()`.
- Detecting whether a scale was already set by the user relies on `plot$scales$scales`, which is a bit buggy.
- Facetted plots are not specifically handled; `ro_gg_y_title()` and `ro_gg_group_ticks()` only look at the first panel.

See Also

- `ro_gg_theme()`, `ro_gg_remove_axis_gap()`, `ro_gg_axis()`, `ro_gg_group_ticks()`, and `ro_gg_y_title()`, which `ro_gg_plot()` combines.

Other ggplot2: `ro_gg_axis()`, `ro_gg_group_ticks()`, `ro_gg_remove_axis_gap()`, `ro_gg_save()`, `ro_gg_theme()`, `ro_gg_y_title()`

Examples

```
## Not run:
library(ggplot2)

dat <- data.frame(
  agegroup = c("0-19", "20-39", "40-59"),
  n = c(1200, 4800, 6100)
)

p <- ggplot(dat, aes(x = agegroup, y = n)) +
  geom_col() +
  labs(title = "Cases by age group", y = "Number of cases", x = "Age group")
```

```
# build the plot fully with `+` first, then call ro_gg_plot() on the finished object
ro_gg_plot(p)

## End(Not run)
```

ro_gg_remove_axis_gap *Remove gap between data and axes*

Description

Simplifies usage of the expansion function of ggplot2. Expansion vectors are used to add some space between the data and the axes. For certain types of graphs you need to adjust the default ggplot expansion values to comply with the RIVM style. For example, if you create a bar plot with `ro_gg_theme()`, there is a gap between the bars and the x-axis. You can remove this gap with the `ro_gg_remove_axis_gap()` function. This function returns the correct expansion vector that you need to use in the scale functions (`scale_y_...` or `scale_x_...`) of your ggplot object. You also need to adjust the expansion if you create a line graph where the y-axis starts at 0.

Usage

```
ro_gg_remove_axis_gap()
```

Value

expansion vector

Adjust x or y scale?

If you create a vertical (default) bar graph, you want to remove the gap between the bars and the x-axis, thus you need to use `ro_gg_remove_axis_gap()` in ggplot's `scale_y_...`. For horizontal bar graphs, it is the other way around. You want to remove the gap between the bars and the y-axis, thus you need to use `ro_gg_remove_axis_gap()` in ggplot's `scale_x_...` function. For line graphs where the y-axis starts at 0 you need to use `ro_gg_remove_axis_gap()` in ggplot's `scale_y_...` function.

See Also

Other ggplot2: [ro_gg_axis\(\)](#), [ro_gg_group_ticks\(\)](#), [ro_gg_plot\(\)](#), [ro_gg_save\(\)](#), [ro_gg_theme\(\)](#), [ro_gg_y_title\(\)](#)

Examples

```
#bar graph with gap removal
my_plot <- mtcars |>
dplyr::mutate(gear = as.factor(gear))|>
dplyr::group_by(gear) |>
dplyr::summarise(hp = mean(hp)) |>
ggplot2::ggplot(ggplot2::aes(x = gear, y = hp)) +
```

```
ggplot2::geom_bar(stat = "identity") +
ro_gg_theme() +
ggplot2::scale_y_continuous(
  expand = ro_gg_remove_axis_gap()
)

#horizontal bar graph with gap removal
my_plot2 <- mtcars |>
dplyr::mutate(gear = as.factor(gear)) |>
dplyr::group_by(gear) |>
dplyr::summarise(hp = mean(hp)) |>
ggplot2::ggplot(ggplot2::aes(x = hp, y = gear)) +
ggplot2::geom_bar(stat = "identity") +
ro_gg_theme(flip = TRUE) +
ggplot2::scale_x_continuous(
  expand = ro_gg_remove_axis_gap()
)

# line graph that starts at 0 with gap removal
my_plot3 <- mtcars |>
dplyr::group_by(gear) |>
dplyr::summarise(hp = mean(hp)) |>
ggplot2::ggplot(ggplot2::aes(x = gear, y = hp)) +
ggplot2::geom_line() +
ggplot2::expand_limits(y = 0) +
ro_gg_theme() +
ggplot2::scale_y_continuous(
  expand = ro_gg_remove_axis_gap()
)
```

ro_gg_save*Save ggplot objects with an offline configuration*

Description

This function can help you save your ggplot objects for offline reports. It changes the font size of your title and text and sets a default set of parameters for offline publication. Originally made for Staat van Infectieziektes.

Usage

```
ro_gg_save(
  filename,
  plot,
  fontsize = 10.75,
  dpi = 600,
  width = 12,
  height = 8,
```

```

    units = "cm",
    ...
  )

```

Arguments

filename	Filename to save object as. Should include preferred extension (e.g "name.png")
plot	ggplot object to save.
fontsize	Fontsize for title and text. Default = 10.75.
dpi	Plot resolution. Default = 600.
width, height	Plot size in units from the units argument. Defaults are 12 and 8
units	Possibilities: "cm", "mm", "px" or "in".
...	Other arguments passed on to the ggsave function.

See Also

`ggplot2::ggsave()`

Other ggplot2: `ro_gg_axis()`, `ro_gg_group_ticks()`, `ro_gg_plot()`, `ro_gg_remove_axis_gap()`, `ro_gg_theme()`, `ro_gg_y_title()`

Examples

```

## Not run:
my_plot <- ggplot2::ggplot(data = datasets::mtcars, mapping = ggplot2::aes(x = wt, y = mpg)) +
  ggplot2::geom_point() +
  ggplot2::labs(title = "mtcars plot") +
  ro_gg_theme()

ro_gg_save(my_plot, filename = "my_plot.png")

## End(Not run)

```

ro_gg_theme

Add RIVM graph theme to ggplot object

Description

`ro_gg_theme()` is a custom ggplot2 theme designed to apply the RIVM style to your visualizations. It functions similarly to built-in ggplot2 themes like `theme_minimal()`. Wherever technically feasible, this theme adheres to the RIVM guidelines for data visualization. We use versioning to enable the user to inspect new theme changes and compare them side-by-side. **version 1.0** is the first published version and `ro_gg_theme(version = 1.0)` corresponds to this version. We also offer the version 0.0, which was the 'work in progress' version of the theme when the official guidelines were not published yet.

Usage

```
ro_gg_theme(
  base_size = 12,
  base_family = "RijksoverheidSansWebText",
  flip = FALSE,
  version = "1.0"
)
```

Arguments

base_size	Integer. Default = 13. If you select version 1.0, base_size is transformed to $\text{base_size} / 13 * 12$ (default = 12).
base_family	Character. Default = "RijksoverheidSansWebText". Accepts system fonts. If RijksoverheidSansWebText is not installed, Verdana is used, then Arial, then the first font found on your system, whichever is installed first. Tip: you can check your available systemfonts with <code>systemfonts::system_fonts()\$family</code> .
flip	Logical. When set to FALSE (default), graphs have a thick x-axis and horizontal grid lines. For certain types of graphs, such as those with horizontal bars, it may be necessary to 'flip' these elements. Setting flip to TRUE changes the style to include a thick y-axis and vertical grid lines. Note: flip is only available from version 1.0. If you set it to TRUE for version 0.0, it will be ignored.
version	Character. The theme version. Default = 1.0.

Value

ggplot theme object

Dependencies

- [ro_color](#)

RIVM style characteristics (version 1.0)

The following design elements from the RIVM style guide are incorporated into `ro_gg_theme()`:

- Background and Layout:
 - The background color for all plot elements is white.
 - The y-axis is hidden, while the x-axis is shown.
 - Major x-axis grid lines are hidden, major y-axis grid lines are shown, minor grid lines are hidden.
 - Titles (main, subtitle, caption, and facet) are aligned to the left of the plot area.
 - The x-axis title is aligned to the right of the plot area.
 - The legend is positioned at the bottom and aligned to the left of the plot area.
 - Margins around the plot have been increased to ensure all text elements have enough space.
- Colors:

- Titles (main, subtitle, legend, facet): `ro_color("donkerblauw")`
- Caption: `"#505050"`
- Text and axis lines: `ro_color("grijs_7")` and default `base_line_size`.
- Grid lines: `ro_color("grijs_3")`
- Text:
 - All text elements are proportionally sized relative to `base_size`:
 - Main title: `ro_color("donkerblauw")`, bold and $1.42 \times \text{base_size}$.
 - Subtitle: `ro_color("donkerblauw")` and $1.17 \times \text{base_size}$.
 - Other titles: `ro_color(donkerblauw)` and $1.08 \times \text{base_size}$.
 - Caption: `"#505050"` and default `base_size`.
 - Legend text: `ro_color("grijs_7")` and $1.08 \times \text{base_size}$.
 - Other text: `ro_color("grijs_7")` and default `base_size`.
- Grid and Axis Formatting:
 - All line elements are proportionally sized relative to `base_line_size` ($\text{base_size} / 22$):
 - Axis lines: `ro_color("grijs_7")` and default `base_line_size`.
 - Grid lines: `ro_color("grijs_3")` and $0.5 \times \text{base_line_size}$.
- Other:
 - Legend keys are designed to be twice as wide as they are high.
 - Space between facets are twice the default size.

What are the differences between `ro_gg_theme` and the official RIVM style?

- Font family: The official RIVM style uses the Rijksoverheid Sans font. This theme does not add the RO fonts to your device, but will use them if available. If the RO font is not available, Verdana is used, then Arial, then the first font found on your system, whichever is installed first (see [ro_check_if_font_available](#)).
- Y-axis title alignment: The official RIVM style rotates the y-axis title horizontally and aligns it to the top of the plot area, which ggplot2 currently doesn't support. As a result, y-axis alignment is left to the user.
- X-axis ticks: The official RIVM style positions x-axis ticks between the bars of a barplot. Achieving this in ggplot2 would require manually calculating the exact tick positions, which is too specific and inconsistent to implement as part of a general theme.

Theme customization approach

There are two approaches for creating a custom ggplot2 theme: (1) modifying an existing theme using `'existing theme + theme(...)'` and (2) fully replacing it with `'existing theme %+replace% theme()'`.

For `ro_gg_theme()`, we chose the first approach (`'existing theme + theme(...)'`). This method offers greater flexibility to users, making it easier to apply additional customization on top of the `ro_gg_theme()` defaults.

See Also

- To view RIVM colors, use ROvis.utils function [ro_show_colors](#).
- To access the hex color code of a specific color, use the ROvis.utils function [ro_color](#).
- To view more usage examples, refer to the [ROvis website](#).

Other ggplot2: [ro_gg_axis\(\)](#), [ro_gg_group_ticks\(\)](#), [ro_gg_plot\(\)](#), [ro_gg_remove_axis_gap\(\)](#), [ro_gg_save\(\)](#), [ro_gg_y_title\(\)](#)

Examples

```
## Not run:
#without theme
my_plot <- ggplot2::ggplot(data = datasets::mtcars, mapping = ggplot2::aes(x = wt, y = mpg)) +
  ggplot2::geom_point() +
  ggplot2::labs(title = "mtcars plot")

#with theme version 0.0
my_plot +
  ROvis.utils::ro_gg_theme(version = "0.0")

#with theme version 1.0 (default)
my_plot +
  ROvis.utils::ro_gg_theme()

# different y-axis alignment choices
# top-aligned
my_plot +
  ROvis.utils::ro_gg_theme() +
  ggplot2::theme(axis.title.y = element_text(hjust = 1))
# annotated text as y-axis
my_plot +
  ROvis.utils::ro_gg_theme() +
  ggplot2::theme(axis.title.y = element_blank()) +
  ggplot2::annotate("text", x = 0, y = 35.5, label = "mpg", color = ro_color("lintblauw"))

## End(Not run)
```

ro_gg_y_title

*Relocate y-axis title of ggplot object to RIVM style position***Description**

[Experimental] Relocates the y-axis title of an existing ggplot object (with `theme_ggrivm(version = "1.0")` applied) to the right location, following RIVM design guidelines. The function calculates the y-coordinate of the highest grid line and positions the y-axis title slightly above this value for optimal appearance. Note: Since the title may (partly) fall outside the plot area, always use this function in combination with ggplot's `coord_cartesian(clip = "off")`. Please note: This is an experimental function. Its features and behavior may change in future updates.

Usage

```
ro_gg_y_title(plot)
```

Arguments

plot	A ggplot object with a defined y-axis title and the custom theme <code>ro_gg_theme(version = "1.0")</code> applied.
------	---

Value

A ggplot object

See Also

Other ggplot2: [ro_gg_axis\(\)](#), [ro_gg_group_ticks\(\)](#), [ro_gg_plot\(\)](#), [ro_gg_remove_axis_gap\(\)](#), [ro_gg_save\(\)](#), [ro_gg_theme\(\)](#)

Examples

```
## Not run:
my_plot <- ggplot2::ggplot(data = datasets::mtcars, mapping = ggplot2::aes(x = wt, y = mpg)) +
  ggplot2::geom_point() +
  ggplot2::labs(title = "mtcars plot") +
  R0vis.utils::ro_gg_theme(version = "1.0")
my_plot

#relocate y-axis title to RIVM style position
R0vis.utils::ro_gg_y_title(my_plot) +
  ggplot2::coord_cartesian(clip = "off")

## End(Not run)
```

Index

* **ggplot2**

- ro_gg_axis, [2](#)
- ro_gg_group_ticks, [3](#)
- ro_gg_plot, [4](#)
- ro_gg_remove_axis_gap, [6](#)
- ro_gg_save, [7](#)
- ro_gg_theme, [8](#)
- ro_gg_y_title, [11](#)

* **ggplotly**

- ro_gg_theme, [8](#)

ggplot2::annotation_custom(), [4](#)

ggplot2::ggsave(), [8](#)

ro_check_if_font_available, [10](#)

ro_color, [9](#), [11](#)

ro_gg_axis, [2](#)

ro_gg_axis(), [3–6](#), [8](#), [11](#), [12](#)

ro_gg_group_ticks, [3](#)

ro_gg_group_ticks(), [2](#), [4–6](#), [8](#), [11](#), [12](#)

ro_gg_plot, [4](#)

ro_gg_plot(), [2](#), [3](#), [6](#), [8](#), [11](#), [12](#)

ro_gg_remove_axis_gap, [6](#)

ro_gg_remove_axis_gap(), [2–5](#), [8](#), [11](#), [12](#)

ro_gg_save, [7](#)

ro_gg_save(), [2](#), [3](#), [5](#), [6](#), [11](#), [12](#)

ro_gg_theme, [8](#)

ro_gg_theme(), [2–6](#), [8](#), [12](#)

ro_gg_y_title, [11](#)

ro_gg_y_title(), [2–6](#), [8](#), [11](#)

ro_show_colors, [11](#)