# Package 'ggrank'

September 18, 2026

**Title** Visualise Changes in Rankings with 'ggplot2'

**Version** 0.1.0

**Description** Calculates, inspects, tabulates, and visualises changes in rankings across two to four ordered states. Creates 'ggplot2'-based rank-transition charts that retain categories entering or leaving a selected top-rank boundary. Supports ranks calculated from numeric values as well as authoritative ranks supplied without values.

**License** MIT + file LICENSE

**URL** https://thinkdenominator.github.io/ggrank/, https://github.com/ThinkDenominator/ggrank

**BugReports** https://github.com/ThinkDenominator/ggrank/issues

**Encoding** UTF-8

**Language** en-GB

**Depends** R (>= 4.1.0)

**LazyData** true

**Imports** dplyr, ggplot2, rlang, scales

**Suggests** knitr, pkgdown, rmarkdown, rstudioapi, shiny, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Rubeshkumar Polani [aut, cre] (ORCID: <https://orcid.org/0000-0002-0418-7592>), Think Denominator [cph]

**Maintainer** Rubeshkumar Polani <rubesh@thinkdenominator.com>

**Repository** CRAN

**Date/Publication** 2026-09-18 11:40:02 UTC

# Contents



---

`ggrank` *Draw a rank-transition chart*

---

## Description

Creates a `ggplot2` chart that combines ranked tables with connecting lines. Categories entering or exiting `top_n` remain visible outside the boundary.

## Usage

```
ggrank(
  data,
  category,
  period,
  value = NULL,
  rank = NULL,
  label = NULL,
  group = NULL,
  periods = NULL,
  top_n = 10,
  direction = c("descending", "ascending"),
  ties = c("min", "dense", "first"),
  show_transitions = c("boundary", "top_only", "all"),
  colour_by = c("auto", "group", "movement", "none"),
  palette = NULL,
  legend_title = NULL,
  legend_labels = NULL,
  show_legend = TRUE,
  category_header = "Category",
  value_header = "Value",
  label_wrap = 28,
  category_width = 2.4,
  value_width = 1.55,
  state_gap = 1.15,
  base_size = 11,
```

```
  title = NULL,
  subtitle = NULL,
  check_rank = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | A data frame with one row per category and period. |
| `category`, `period` | Unquoted columns identifying the category and ordered state. |
| `value` | Optional unquoted numeric ranking-value column. It may be omitted when an authoritative `rank` column is supplied. |
| `rank` | Optional unquoted column containing precomputed ranks. |
| `label` | Optional unquoted display-label column. By default `value` is formatted using `format()`. |
| `group` | Optional unquoted category-group column. |
| `periods` | Optional vector selecting and ordering two to four states. |
| `top_n` | Rank threshold to retain in each state. All categories tied at the boundary are included, so the result may contain more than `top_n` categories. |
| `direction` | Whether large (`"descending"`) or small (`"ascending"`) values rank first. |
| `ties` | Ranking method: `"min"` (the default competition ranking), `"dense"`, or `"first"` (unique alphabetical ranks). |
| `show_transitions` | `"boundary"` retains categories appearing in the top N in any selected state; `"top_only"` includes top-N observations only; `"all"` includes every category. |
| `colour_by` | Colour categories by `"group"`, `"movement"`, or `"none"`. The default `"auto"` uses groups when supplied and movement otherwise. |
| `palette` | Optional named colour vector. |
| `legend_title` | Optional legend title. |
| `legend_labels` | Optional named labels corresponding to every displayed group or movement value. |
| `show_legend` | Show the colour legend. |
| `category_header`, `value_header` | Headers shown over the two box columns. |
| `label_wrap` | Approximate number of characters per category-label line. |
| `category_width`, `value_width` | Relative widths of the category and value boxes. Increase `value_width` for long confidence-interval labels. |
| `state_gap` | Horizontal space reserved for connectors between states. |
| `base_size` | Base text size passed to `theme_ggrank()`. |
| `title`, `subtitle` | Optional plot title and subtitle. |
| `check_rank` | When `TRUE`, supplied ranks are checked for disagreements with equal values, shared ranks across different values, and the requested ranking direction. Potential disagreements warn rather than fail because authoritative ranks may use external tie-breakers or additional data. |

## Details

With `colour_by = "auto"`, supplying `group` uses group colours; otherwise movement colours are used. Movement classification prioritises entry to and exit from the selected top-N boundary, followed by positive (`riser`), negative (`faller`), or zero (`stable`) rank change. Palette and legend-label names must match the displayed group or movement values exactly. For charts with three or four periods, movement colour summarises the net change between the first and last displayed period; use `ggrank_change()` to inspect each adjacent transition.

## Value

A `ggplot` object.

## Examples

```
ggrank(ggrank_causes, cause, year, rate,
  rank = rank, label = display_value, group = cause_group,
  periods = c(1990, 2021), top_n = 10
)
```

---

# ggrank_app *Launch the ggrank graphical interface*

---

## Description

Opens a local Shiny application for creating a rank-transition chart, inspecting its comparison table, and visualising its largest rank changes. Users can start with the synthetic teaching data or upload a CSV file. The guided workflow supports either numeric values that need ranking or an existing rank column with no marks, rates, scores, or values.

## Usage

```
ggrank_app(..., launch.browser = NULL)
```

## Arguments

| | |
|---|---|
| `...` | Additional arguments passed to `shiny::runApp()`. |
| `launch.browser` | Logical or a function passed to `shiny::runApp()`. The default uses the RStudio Viewer when available and otherwise opens a browser during an interactive session. |

## Details

The app is a companion to the code-first workflow. It shows and downloads reusable R code for the selected analysis.

## Value

Invisibly returns the value from `shiny::runApp()`.

## Examples

```
if (interactive()) {
  ggrank_app()
}
```

---

`ggrank_causes` *Synthetic GBD-inspired causes*

---

## Description

A synthetic teaching dataset with cause rankings and uncertainty intervals. Values do not represent published Global Burden of Disease estimates. This dataset was generated specifically for package examples and contains no downloaded IHME or Global Burden of Disease data. The package is independent and is not affiliated with or endorsed by IHME.

## Usage

```
ggrank_causes
```

## Format

A data frame with 36 rows and 8 variables: `year`, `cause`, `cause_group`, `rank`, `rate`, `lower`, `upper`, and `display_value`.

## Source

Synthetic data created for ggrank.

---

`ggrank_change` *Visualise the largest rank changes*

---

## Description

Creates a diverging bar chart answering "Who moved the most?" Positive values rose towards rank one; negative values fell away from rank one. Raw data are processed by `ggrank_table()`, preserving one ranking engine.

## Usage

```
ggrank_change(
  data,
  category = NULL,
  period = NULL,
  value = NULL,
  rank = NULL,
  label = NULL,
  group = NULL,
  periods = NULL,
  top = 15,
  top_n = 10,
  direction = c("descending", "ascending"),
  ties = c("min", "dense", "first"),
  check_rank = TRUE,
  comparison = c("latest", "all"),
  from = NULL,
  to = NULL,
  show_stable = FALSE,
  change_label = c("change", "ranks", "none"),
  label_wrap = 30,
  palette = c(riser = "#0072B2", faller = "#D55E00", stable = "#667085"),
  legend_title = NULL,
  legend_labels = NULL,
  show_legend = TRUE,
  base_size = 11,
  title = NULL,
  subtitle = NULL
)
```

## Arguments

| | |
|---|---|
| `data` | Raw data or a rank-change table returned by `ggrank_table()`. |
| `category, period` | Unquoted columns used with raw `data`. |
| `value` | Optional numeric value column. It may be omitted when `rank` is supplied. |
| `rank, label, group` | Optional unquoted columns used with raw `data`. |
| `periods` | Optional vector selecting and ordering two to four periods. |
| `top` | Maximum categories displayed per comparison, selected by `abs(rank_change)`. This is not `top` risers plus `top` fallers. |
| `top_n` | Top-rank boundary used to classify entrants and exits with raw data. It does not control the number of bars; `top` does. |
| `direction, ties, check_rank` | Ranking options passed to `ggrank_table()`. |
| `comparison` | Display the latest consecutive comparison (default) or all comparisons. Ignored when `from` and `to` are supplied. |

| | |
|---|---|
| from, to | Optional explicit comparison. Both must be supplied and the pair must already exist in the rank-change table. |
| show_stable | Include unchanged categories. The default excludes them. |
| change_label | Show signed change ("change"), detailed ranks such as "7 -> 3 (+4)" ("ranks"), or no bar-end label ("none"). |
| label_wrap | Approximate characters per category-label line. |
| palette | Named colours for riser, faller, and stable. |
| legend_title | Optional legend title. |
| legend_labels | Optional named labels for displayed movements. |
| show_legend | Show the movement legend. |
| base_size | Base text size. |
| title, subtitle | Optional text overriding informative defaults. Use subtitle = "" to suppress the default subtitle. |

## Value

A ggplot object whose data retain the underlying table status.

## Examples

```
ggrank_change(ggrank_products, product, year, sales, top = 5)

changes <- ggrank_table(
  ggrank_products, product, year, sales,
  top_n = 5, show_transitions = "all"
)
ggrank_change(changes, top = 5, comparison = "all")
```

---

# ggrank_data *Prepare and inspect ranked data*

---

## Description

Calculates ranks within each state before any top-N display filtering is applied. This helper is optional: ggrank() performs the same ranking automatically. Use it when you want to inspect, teach, export, or reuse the calculated ranks.

## Usage

```
ggrank_data(
  data,
  category,
  period,
  value = NULL,
  rank = NULL,
```

```
  label = NULL,
  group = NULL,
  periods = NULL,
  direction = c("descending", "ascending"),
  ties = c("min", "dense", "first"),
  check_rank = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | A data frame with one row per category and period. |
| `category, period` | Unquoted columns identifying the category and state. |
| `value` | Optional unquoted numeric ranking-value column. It may be omitted when an authoritative `rank` column is supplied. |
| `rank` | Optional unquoted column containing authoritative precomputed ranks. When supplied, these ranks are preserved. |
| `label` | Optional unquoted display-label column. |
| `group` | Optional unquoted category-group column. |
| `periods` | Optional vector selecting and ordering states. Unlike `ggrank()`, this preparation helper is not limited to four states. |
| `direction` | Whether large (`"descending"`) or small (`"ascending"`) values rank first. |
| `ties` | Ranking method: `"min"` (competition ranking), `"dense"`, or `"first"` (unique ranks resolved alphabetically by category). |
| `check_rank` | When `TRUE`, supplied ranks are checked for potential disagreements with the values and ranking direction. These checks warn rather than fail because authoritative ranks may use external information. |

## Details

Ranking uses the exact numeric `value`; formatting supplied through `label` never changes the rank. By default, equal values receive the same competition rank (`1, 2, 3, 3, 5`). Tied categories receive separate alphabetical display positions so their plot boxes do not overlap.

## Value

A data frame containing `category`, `period`, `value`, `rank`, `display_position`, `label`, and `group`.

## Examples

```
ggrank_data(ggrank_products, product, year, sales)

tied <- data.frame(
  year = rep(c(2020, 2025), each = 4),
  organism = rep(LETTERS[1:4], 2),
  rate = c(5, 4, 3, 3, 6, 4, 4, 2)
)
```

```
ggrank_data(tied, organism, year, rate)

ranks_only <- data.frame(
  year = rep(c(2024, 2025), each = 3),
  student = rep(c("A", "B", "C"), 2),
  rank = c(1, 2, 3, 2, 1, 3)
)
ggrank_data(ranks_only, student, year, rank = rank)
```

---

## ggrank_products *Synthetic product rankings*

---

### Description

A general-purpose teaching dataset of product sales across three years. Every product name, ranking, and sales value is synthetic and was generated specifically for package teaching examples.

### Usage

```
ggrank_products
```

### Format

A data frame with 24 rows and 4 variables: `year`, `product`, `category`, and `sales`.

### Source

Synthetic data created for ggrank.

---

## ggrank_table *Build a readable rank-transition table*

---

### Description

Produces the analytical companion to `ggrank()`. Each row compares a category between two adjacent selected states. A two-state comparison has one row per category; three or four states produce successive transition rows such as 1990 to 2010 and 2010 to 2021.

## Usage

```
ggrank_table(
  data,
  category,
  period,
  value = NULL,
  rank = NULL,
  label = NULL,
  group = NULL,
  periods = NULL,
  top_n = 10,
  direction = c("descending", "ascending"),
  ties = c("min", "dense", "first"),
  show_transitions = c("boundary", "top_only", "all"),
  check_rank = TRUE
)
```

## Arguments

| | |
|---|---|
| `data` | A data frame with one row per category and period. |
| `category, period` | Unquoted columns identifying the category and ordered state. |
| `value` | Optional unquoted numeric ranking-value column. It may be omitted when an authoritative `rank` column is supplied. |
| `rank` | Optional unquoted column containing precomputed ranks. |
| `label` | Optional unquoted display-label column. By default `value` is formatted using `format()`. |
| `group` | Optional unquoted category-group column. |
| `periods` | Optional vector selecting and ordering two to four states. |
| `top_n` | Rank threshold to retain in each state. All categories tied at the boundary are included, so the result may contain more than `top_n` categories. |
| `direction` | Whether large (`"descending"`) or small (`"ascending"`) values rank first. |
| `ties` | Ranking method: `"min"` (the default competition ranking), `"dense"`, or `"first"` (unique alphabetical ranks). |
| `show_transitions` | `"boundary"` retains categories appearing in the top N in any selected state; `"top_only"` includes top-N observations only; `"all"` includes every category. |
| `check_rank` | When `TRUE`, supplied ranks are checked for disagreements with equal values, shared ranks across different values, and the requested ranking direction. Potential disagreements warn rather than fail because authoritative ranks may use external tie-breakers or additional data. |

## Details

Rank change is calculated as `rank_from - rank_to`: positive values rose towards rank one, negative values fell, and zero is stable. `status` is `"entrant"` when a category crosses from outside to

inside top_n, and "exit" for the reverse. These boundary statuses take precedence over "riser" and "faller". "new" and "absent" indicate that a category exists on only one side; "missing" identifies a supplied non-finite value.

## Value

A data frame with category, transition states, ranks, rank change, values, value change, labels, group, missing-value indicators, and movement status. Positive rank_change means that a category rose in the ranking.

## Examples

```
ggrank_table(
  ggrank_products, product, year, sales,
  periods = c(2022, 2024), top_n = 5
)
```

---

# theme_ggrank *A minimal theme for rank-transition charts*

---

## Description

A minimal theme for rank-transition charts

## Usage

```
theme_ggrank(base_size = 11, base_family = "")
```

## Arguments

| | |
|---|---|
| base_size | Base font size. |
| base_family | Base font family. |

## Value

A complete ggplot2 theme.

## Examples

```
theme_ggrank()
```

# Index