Package: ggmemo (via r-universe)

June 17, 2026

Title Add Arrows, Labels, and Change Annotations to 'ggplot2' Charts

Version 0.1.0.9000

Description Add callout arrows, highlight data points, and show percent change between rows on 'ggplot2' charts in one line of code. annotate_callout() points at a data row with an arrow and label. annotate_change() draws a color-coded arrow between two rows and labels the delta as percent change, absolute difference, or percentage points. Designed for business charts, quarterly reports, and dashboards. Built on top of the 'ggpp' package.

URL https://lindsay-lintelman.github.io/ggmemo/, https://github.com/lindsay-lintelman/ggmemo

BugReports https://github.com/lindsay-lintelman/ggmemo/issues

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Imports ggplot2, ggpp, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev

Repository https://lindsay-lintelman.r-universe.dev

Date/Publication 2026-06-16 21:34:37 UTC

RemoteUrl https://github.com/lindsay-lintelman/ggmemo

RemoteRef HEAD

RemoteSha fca731f3549f46dd951ae38dab21932c970fe1c4

Contents

---

ggmemo-package *ggmemo: Add Arrows, Labels, and Change Annotations to 'ggplot2' Charts*

Description

Add arrows, labels, and change annotations to ggplot2 charts in one line of code. Two functions:

- `annotate_callout()`: Point at a data row with an arrow and label.
- `annotate_change()`: Show the delta between two rows as percent change, absolute change, or percentage points.

Both return standard ggplot2 layers — add them with +.

Install from GitHub (not on CRAN):

```
pak::pak("lindsay-lintelman/ggmemo")
```

Why ggmemo instead of manual ggplot2 annotation?

Manual annotation requires hardcoding coordinates, computing deltas, formatting labels, and picking colors (~10 lines). ggmemo replaces that with a single function call:

```
# Without ggmemo:
annotate("segment", x = "Q1", xend = "Q4", y = 120, yend = 158,
         arrow = arrow(length = unit(0.15, "inches")),
         colour = "#2E7D32", linewidth = 0.6) +
annotate("label", x = 2.5, y = 139, label = "+31.7
         colour = "#2E7D32", fill = "white", fontface = "bold")

# With ggmemo:
annotate_change(data, from = quarter == "Q1",
                to = quarter == "Q4", value = revenue)
```

Quick reference

```
# Label a data point
annotate_callout(data, where, label, position, nudge, ...)

# Show change between two points
annotate_change(data, from, to, value, format, colors, ...)

format options: "percent" (default), "absolute", "points", "both"
```

Common tasks

Label a peak or milestone	`annotate_callout(df, where = date == "2024-06-01", label = "Peak")`
Show percent change	`annotate_change(df, from = ..., to = ..., value = sales)`
Show absolute difference	`annotate_change(..., format = "absolute")`
Show percentage point change	`annotate_change(..., format = "points")`
Use custom colors	`annotate_change(..., colors = c(up = "#1B9E77", down = "#D95F02", flat = "#999")`
Override label styling	`annotate_callout(..., size = 4, fill = "lightyellow")`

When to use ggmemo

Use ggmemo when you want to annotate a ggplot2 chart with arrows, callout labels, or change annotations without manually computing coordinates, formatting deltas, or positioning text. Common scenarios: quarterly reports, executive dashboards, time-series narration, before/after comparisons.

When NOT to use ggmemo

- Repelling overlapping labels: use the ggrepel package.
- NPC (normalized parent coordinates) positioning: use the ggpp package.
- Interactive annotations: use plotly or ggiraph.
- Theming or styling: use ggthemes, hrbrthemes, or bbplot.

Author(s)

Maintainer: Lindsay Lintelman <lindsay.lintelman@posit.co>

Authors:

- Lindsay Lintelman <lindsay.lintelman@posit.co>

See Also

Useful links:

- https://lindsay-lintelman.github.io/ggmemo/
- https://github.com/lindsay-lintelman/ggmemo
- Report bugs at https://github.com/lindsay-lintelman/ggmemo/issues

Examples

```
library(ggplot2)
library(ggmemo)

# -- Complete template: narrated business chart --
# Data
revenue <- data.frame(
  quarter = factor(c("Q1", "Q2", "Q3", "Q4"),
```

```
                    levels = c("Q1", "Q2", "Q3", "Q4")),
  revenue = c(120, 145, 132, 158)
)

# Annotated chart
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "steelblue", width = 0.6) +
  annotate_callout(
    revenue,
    where = quarter == "Q4",
    label = "Record quarter",
    position = "top-left"
  ) +
  annotate_change(
    revenue,
    from = quarter == "Q1",
    to = quarter == "Q4",
    value = revenue
  ) +
  labs(title = "2024 Quarterly Revenue ($K)", x = NULL, y = NULL) +
  theme_minimal()
```

`annotate_callout` *Add a callout annotation to a ggplot*

Description

Points at a specific data row with an arrow and label. The callout consists of a text label inside a rounded box, connected to the target data point by a line segment with an arrowhead. Built on top of `ggpp::geom_label_s()`.

Usage

```
annotate_callout(data, where, label, position = "top-right", nudge = NULL, ...)
```

Arguments

`data`	A data frame. Should be the same data frame used in the ggplot, or a subset of it. Must contain the columns mapped to x and y in the plot's `aes()`. Note: the automatic nudge heuristic estimates label offset from the data ranges, but it guesses which columns are x and y. For data frames with many numeric columns, passing a two-column subset (e.g., `data[, c("date", "sales")]`) or setting nudge explicitly gives more reliable placement.
`where`	<tidy-eval> A filtering expression that identifies exactly one row of `data`. For example, `year == 2020` or `quarter == "Q4" & region == "West"`. An error is thrown if the expression matches zero or more than one row.
`label`	A single character string for the annotation text.

position	Where to place the label relative to the data point. One of "top-right" (default), "top-left", "bottom-right", or "bottom-left".
nudge	Optional numeric vector of length 2 (c(x, y)) giving explicit nudge amounts in data units. Overrides the automatic nudge heuristic, which estimates 5% of the x and y data ranges. The heuristic works well when data contains only the plotted columns; if data has many numeric columns (like ggplot2::economics), passing a two-column subset or setting nudge explicitly avoids the heuristic picking the wrong column's range.
...	Additional arguments passed to ggpp::geom_label_s(). Use to override defaults like size, colour, fill, alpha, or arrow.

Value

A ggplot2 layer that can be added to a plot with +.

See Also

annotate_change() to label the delta between two data points.

Examples

```
library(ggplot2)

p <- ggplot(economics, aes(x = date, y = unemploy)) +
  geom_line()

# Basic callout
p + annotate_callout(
  economics,
  where = date == as.Date("2009-10-01"),
  label = "Peak unemployment",
  position = "top-right"
)

# With explicit nudge (useful when data has many numeric columns)
p + annotate_callout(
  economics,
  where = date == as.Date("2009-10-01"),
  label = "Peak unemployment",
  nudge = c(365, 500)
)

# Customize label appearance via ... (larger text, yellow background)
p + annotate_callout(
  economics,
  where = date == as.Date("2009-10-01"),
  label = "Peak unemployment",
  nudge = c(365, 500),
  size = 5, fill = "lightyellow"
)
```

```
# Mark both the peak and the trough on the same chart
p +
  annotate_callout(
    economics,
    where = date == as.Date("2009-10-01"),
    label = "Peak",
    nudge = c(365, 500)
  ) +
  annotate_callout(
    economics,
    where = date == as.Date("2000-01-01"),
    label = "Dot-com low",
    position = "bottom-right",
    nudge = c(365, 500)
  )
```

---

annotate_change *Annotate the change between two data points on a ggplot*

Description

Draws a curved arrow between two data rows and labels the midpoint with the computed delta. The label is color-coded: dark green for increases, dark red for decreases, grey for no change. Built on top of ggplot2::annotate().

Usage

```
annotate_change(
  data,
  from,
  to,
  value,
  format = "percent",
  colors = c(up = "#2E7D32", down = "#B22222", flat = "#808080"),
  curvature = -0.2,
  arrow_pad = 0.04,
  expand_y = TRUE,
  ...
)
```

Arguments

data	A data frame. Should be the same data frame used in the ggplot. Must contain the columns mapped to x and y in the plot's aes(), as well as the column specified in value.
from	<tidy-eval> A filtering expression that identifies exactly one row of data for the start of the arrow. For example, quarter == "Q2". An error is thrown if the expression matches zero or more than one row.

to	<tidy-eval> A filtering expression that identifies exactly one row of data for the end of the arrow.
value	<tidy-eval> An unquoted column name indicating which numeric column to compute the change on. For example, value = revenue.
format	How to format the delta label. One of "percent" (default), "absolute", "points", or "both". Percent change from a zero base value falls back to absolute with a warning. Percent change from negative values uses the raw formula and may be confusing; use "absolute" for data that can go negative. Use "points" when the data is already a rate or percentage (e.g., savings rate, market share) — it labels the difference in percentage points (e.g., "+9.8 %pts") instead of computing a misleading percent-of-percent.
colors	Named character vector of length 3 with hex color values for the arrow and label. Names must be "up", "down", and "flat". Defaults to dark green, dark red, and grey.
curvature	Numeric value controlling the curve of the arrow. Positive values curve right, negative values curve left. Defaults to -0.2 for a subtle leftward arc. Set to 0 for a straight arrow.
arrow_pad	Fraction of the y-axis range to lift both arrow endpoints above the data values, creating visible whitespace between the arrow and bars or points. Defaults to 0.04 (4%). Set to 0 for no gap.
expand_y	Logical. If TRUE (default) and curvature is non-zero, adds a scale_y_continuous(expand = ...) layer to prevent the curved arrow from being clipped at the figure edge. The expansion amount scales with abs(curvature). Set to FALSE to suppress this and control the y-axis expansion yourself.
...	Additional arguments passed to the **label** layer (ggplot2::annotate() with geom = "label"). Use to override defaults like size, fontface, or fill. Note: these do not affect the arrow segment. To change the arrow, use colors.

Details

The curved arrow may arc outside the default plot area. To prevent clipping, this function automatically includes a coord_cartesian(clip = "off") layer. If you need a different coordinate system (e.g., coord_flip()), add it **after** annotate_change() so it takes precedence, and set clip = "off" on your coord to keep the arrow visible.

When expand_y = TRUE (the default), the function also adds a scale_y_continuous(expand = ...) layer that pads the y-axis proportionally to abs(curvature). If you set your own scale_y_continuous() **after** annotate_change(), your scale replaces the one from this function.

Value

A list of ggplot2 layers (arrow, label, coord_cartesian(clip = "off"), and optionally scale_y_continuous(expand = ...)) that can be added to a plot with +. The coord layer prevents the curved arrow from being clipped at the plot panel boundary; the scale layer expands the y-axis to accommodate the curve arc.

See Also

annotate_callout() to label a single data point.

Examples

```
library(ggplot2)

revenue <- data.frame(
  quarter = factor(c("Q1", "Q2", "Q3", "Q4"),
                   levels = c("Q1", "Q2", "Q3", "Q4")),
  revenue = c(120, 145, 132, 158)
)

# Percent change (default)
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "grey70", width = 0.6) +
  annotate_change(
    revenue,
    from = quarter == "Q1",
    to = quarter == "Q4",
    value = revenue
  )

# Absolute change
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "grey70", width = 0.6) +
  annotate_change(
    revenue,
    from = quarter == "Q1",
    to = quarter == "Q4",
    value = revenue,
    format = "absolute"
  )

# Percentage points (for data already expressed as rates)
rates <- data.frame(
  year = 2020:2023,
  rate = c(3.5, 8.1, 5.4, 3.7)
)
ggplot(rates, aes(x = year, y = rate)) +
  geom_line() +
  geom_point() +
  annotate_change(rates, from = year == 2020, to = year == 2021,
                  value = rate, format = "points")

# Custom colors (e.g., corporate palette)
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "grey70", width = 0.6) +
  annotate_change(
    revenue,
    from = quarter == "Q1",
    to = quarter == "Q4",
    value = revenue,
    colors = c(up = "#1B9E77", down = "#D95F02", flat = "#7570B3")
  )
```

```
# Date x-axis (time series) — use nudge on the callout for wide data
ggplot(economics, aes(x = date, y = psavert)) +
  geom_line() +
  annotate_change(
    economics,
    from = date == as.Date("2005-07-01"),
    to = date == as.Date("2012-12-01"),
    value = psavert,
    format = "points"
  )

# Showing a decline (red arrow, negative label)
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "grey70", width = 0.6) +
  annotate_change(
    revenue,
    from = quarter == "Q2",
    to = quarter == "Q3",
    value = revenue
  )

# Multiple change annotations (quarter-over-quarter)
ggplot(revenue, aes(x = quarter, y = revenue)) +
  geom_col(fill = "grey70", width = 0.6) +
  annotate_change(revenue, from = quarter == "Q1",
                  to = quarter == "Q2", value = revenue) +
  annotate_change(revenue, from = quarter == "Q2",
                  to = quarter == "Q3", value = revenue) +
  annotate_change(revenue, from = quarter == "Q3",
                  to = quarter == "Q4", value = revenue)

# Year-over-year growth on a line chart
annual <- data.frame(year = 2019:2024,
                     revenue = c(80, 65, 72, 95, 110, 128))
ggplot(annual, aes(x = year, y = revenue)) +
  geom_line() + geom_point() +
  annotate_change(annual, from = year == 2019,
                  to = year == 2024, value = revenue) +
  annotate_callout(annual, where = year == 2020,
                   label = "COVID dip", position = "bottom-right")

# Combined with annotate_callout() on a time series
ggplot(economics, aes(x = date, y = psavert)) +
  geom_line() +
  annotate_callout(
    economics,
    where = date == as.Date("2005-07-01"),
    label = "All-time low",
    nudge = c(365, 1)
  ) +
  annotate_change(
    economics,
    from = date == as.Date("2005-07-01"),
```

```
  to = date == as.Date("2012-12-01"),
  value = psavert,
  format = "points"
)
```

Index