Package 'ggtime'

February 9, 2026

Title Grammar of Graphics and Plot Helpers for Time Series Visualization

Version 0.2.0

Description Extends the capabilities of 'ggplot2' by providing grammatical elements and plot helpers designed for visualizing temporal patterns. The package implements a grammar of temporal graphics, which leverages calendar structures to highlight changes over time. The package also provides plot helper functions to quickly produce commonly used time series graphics, including time plots, season plots, and seasonal sub-series plots.

License GPL (>= 3)

URL https://pkg.mitchelloharawild.com/ggtime/, https://github.com/mitchelloharawild/ggtime

Imports ggplot2, grid, gtable, lifecycle, rlang, scales, tsibble, fabletools, dplyr, lubridate (>= 1.7.1), tidyr, vctrs, cli

Suggests testthat (>= 3.0.0), tsibbledata, feasts, fable, ggrepel, roxygen2, ggdist, distributional, vdiffr, svglite, fontquiver, sysfonts, showtext

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

BugReports https://github.com/mitchelloharawild/ggtime/issues

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre] (ORCID: <https://orcid.org/0000-0001-6729-7695>), Cynthia A. Huang [aut] (ORCID: <https://orcid.org/0000-0002-9218-987X>), Matthew Kay [aut] (ORCID: <https://orcid.org/0000-0001-9446-0419>), Rob Hyndman [aut] (ORCID: <https://orcid.org/0000-0002-2140-5352>), Earo Wang [ctb] (ORCID: <https://orcid.org/0000-0001-6448-5260>)

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2026-02-09 08:10:02 UTC

Contents

`autoplot.dcmp_ts` *Decomposition plots*

Description

Produces a faceted plot of the components used to build the response variable of the dable. Useful for visualising how the components contribute in a decomposition or model.

Usage

```
## S3 method for class 'dcmp_ts'
autoplot(object, .vars = NULL, scale_bars = TRUE, level = c(80, 95), ...)
```

Arguments

`object`	A dable.
`.vars`	The column of the dable used to plot. By default, this will be the response variable of the decomposition.
`scale_bars`	If `TRUE`, each facet will include a scale bar which represents the same units across each facet.
`level`	If the decomposition contains distributions, which levels should be used to display intervals?
`...`	Further arguments passed to `ggplot2::geom_line()`, which can be used to specify fixed aesthetics such as `colour = "red"` or `size = 3`.

Value

A ggplot object showing a set of time plots of the decomposition.

Examples

```
library(fabletools)
library(feasts)
tsibbledata::aus_production %>%
  model(STL(Beer)) %>%
  components() %>%
  autoplot()
```

---

autoplot.fbl_ts *Plot a set of forecasts*

Description

Produces a forecast plot from a fable. As the original data is not included in the fable object, it will need to be specified via the `data` argument. The `data` argument can be used to specify a shorter period of data, which is useful to focus on the more recent observations.

Usage

```
## S3 method for class 'fbl_ts'
autoplot(object, data = NULL, level = c(80, 95), show_gap = TRUE, ...)

## S3 method for class 'fbl_ts'
autolayer(
  object,
  data = NULL,
  level = c(80, 95),
  point_forecast = list(mean = mean),
  show_gap = TRUE,
  ...
)
```

Arguments

`object`	A fable.
`data`	A tsibble with the same key structure as the fable.
`level`	The confidence level(s) for the plotted intervals.
`show_gap`	Setting this to `FALSE` will connect the most recent value in `data` with the forecasts.
`...`	Further arguments passed used to specify fixed aesthetics for the forecasts such as `colour = "red"` or `linewidth = 3`.
`point_forecast`	The point forecast measure to be displayed in the plot.

Examples

```
library(fable)
library(tsibbledata)

fc <- aus_production %>%
  model(ets = ETS(log(Beer) ~ error("M") + trend("Ad") + season("A"))) %>%
  forecast(h = "3 years")

fc %>%
  autoplot(aus_production)


aus_production %>%
  autoplot(Beer) +
  autolayer(fc)
```

`autoplot.tbl_cf` *Auto- and Cross- Covariance and -Correlation plots*

Description

Produces an appropriate plot for the result of `feasts::ACF()`, `feasts::PACF()`, or `feasts::CCF()`.

Usage

```
## S3 method for class 'tbl_cf'
autoplot(object, level = 95, ...)
```

Arguments

`object`	A tbl_cf object (the result `feasts::ACF()`, `feasts::PACF()`, or `feasts::CCF()`).
`level`	The level of confidence for the blue dashed lines.
`...`	Unused.

Value

A ggplot object showing the correlations.

autoplot.tbl_ts *Plot time series from a tsibble*

Description

Produces a time series plot of one or more variables from a tsibble. If the tsibble contains a multiple keys, separate time series will be identified by colour.

Usage

```
## S3 method for class 'tbl_ts'
autoplot(object, .vars = NULL, ...)

## S3 method for class 'tbl_ts'
autolayer(object, .vars = NULL, ...)
```

Arguments

object	A tsibble.
.vars	A bare expression containing data you wish to plot. Multiple variables can be plotted using ggplot2::vars().
...	Further arguments passed to ggplot2::geom_line(), which can be used to specify fixed aesthetics such as colour = "red" or size = 3.

Value

A ggplot object showing a time plot of a time series.

Examples

```
library(dplyr)
tsibbledata::gafa_stock %>%
 autoplot(vars(Close, log(Close)))
```

gg_arma *Plot characteristic ARMA roots*

Description

Produces a plot of the inverse AR and MA roots of an ARIMA model. Inverse roots outside the unit circle are shown in red.

Usage

```
gg_arma(data)
```

Arguments

`data` A mable containing models with AR and/or MA roots.

Details

Only models which compute ARMA roots can be visualised with this function. That is to say, the `glance()` of the model contains `ar_roots` and `ma_roots`.

Value

A ggplot object the characteristic roots from ARMA components.

Examples

```
if (requireNamespace("fable", quietly = TRUE)) {
library(fable)
library(tsibble)
library(dplyr)

tsibbledata::aus_retail %>%
  filter(
    State == "Victoria",
    Industry == "Cafes, restaurants and catering services"
  ) %>%
  model(ARIMA(Turnover ~ pdq(0,1,1) + PDQ(0,1,1))) %>%
  gg_arma()
}
```

`gg_irf` *Plot impulse response functions*

Description

Produces a plot of impulse responses from an impulse response function.

Usage

```
gg_irf(data, y = all_of(measured_vars(data)))
```

Arguments

`data` A tsibble with impulse responses

`y` The impulse response variables to plot (defaults to all measured variables).

Value

A ggplot object of the impulse responses.

`gg_lag` *Lag plots*

Description

A lag plot shows the time series against lags of itself. It is often coloured the seasonal period to identify how each season correlates with others.

Usage

```
gg_lag(
  data,
  y = NULL,
  period = NULL,
  lags = 1:9,
  geom = c("path", "point"),
  arrow = FALSE,
  ...
)
```

Arguments

`data` A tidy time series object (tsibble)

`y` The variable to plot (a bare expression). If NULL, it will automatically selected from the data.

`period` The seasonal period to display. If NULL (default), the largest frequency in the data is used. If numeric, it represents the frequency times the interval between observations. If a string (e.g., "1y" for 1 year, "3m" for 3 months, "1d" for 1 day, "1h" for 1 hour, "1min" for 1 minute, "1s" for 1 second), it's converted to a Period class object from the lubridate package. Note that the data must have at least one observation per seasonal period, and the period cannot be smaller than the observation interval.

`lags` A vector of lags to display as facets.

`geom` The geometry used to display the data.

`arrow` Arrow specification to show the direction in the lag path. If TRUE, an appropriate default arrow will be used. Alternatively, a user controllable arrow created with `grid::arrow()` can be used.

`...` Additional arguments passed to the geom.

Value

A ggplot object showing a lag plot of a time series.

Examples

```
library(tsibble)
library(dplyr)
tsibbledata::aus_retail %>%
  filter(
    State == "Victoria",
    Industry == "Cafes, restaurants and catering services"
  ) %>%
  gg_lag(Turnover)
```

gg_season *Seasonal plot*

Description

Produces a time series seasonal plot. A seasonal plot is similar to a regular time series plot, except the x-axis shows data from within each season. This plot type allows the underlying seasonal pattern to be seen more clearly, and is especially useful in identifying years in which the pattern changes.

Usage

```
gg_season(
  data,
  y = NULL,
  period = NULL,
  facet_period = NULL,
  max_col = Inf,
  max_col_discrete = 7,
  pal = (scales::hue_pal())(9),
  polar = FALSE,
  labels = c("none", "left", "right", "both"),
  labels_repel = FALSE,
  labels_left_nudge = 0,
  labels_right_nudge = 0,
  ...
)
```

Arguments

`data` A tidy time series object (tsibble)

`y` The variable to plot (a bare expression). If NULL, it will automatically selected from the data.

`period` The seasonal period to display. If NULL (default), the largest frequency in the data is used. If numeric, it represents the frequency times the interval between observations. If a string (e.g., "1y" for 1 year, "3m" for 3 months, "1d" for 1 day, "1h" for 1 hour, "1min" for 1 minute, "1s" for 1 second), it's converted to a

Period class object from the lubridate package. Note that the data must have at least one observation per seasonal period, and the period cannot be smaller than the observation interval.

`facet_period` A secondary seasonal period to facet by (typically smaller than period).

`max_col` The maximum number of colours to display on the plot. If the number of seasonal periods in the data is larger than `max_col`, the plot will not include a colour. Use `max_col = 0` to never colour the lines, or Inf to always colour the lines. If labels are used, then max_col will be ignored.

`max_col_discrete`
The maximum number of colours to show using a discrete colour scale.

`pal` A colour palette to be used.

`polar` If TRUE, the season plot will be shown on polar coordinates.

`labels` Position of the labels for seasonal period identifier.

`labels_repel` If TRUE, the seasonal period identifying labels will be repelled with the ggrepel package.

`labels_left_nudge`, `labels_right_nudge`
Allows seasonal period identifying labels to be nudged to the left or right from their default position.

`...` Additional arguments passed to geom_line()

Value

A ggplot object showing a seasonal plot of a time series.

References

Hyndman and Athanasopoulos (2019) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. https://OTexts.com/fpp3/

Examples

```
library(tsibble)
library(dplyr)
tsibbledata::aus_retail %>%
  filter(
    State == "Victoria",
    Industry == "Cafes, restaurants and catering services"
  ) %>%
  gg_season(Turnover)
```

gg_subseries *Seasonal subseries plots*

Description

A seasonal subseries plot facets the time series by each season in the seasonal period. These facets form smaller time series plots consisting of data only from that season. If you had several years of monthly data, the resulting plot would show a separate time series plot for each month. The first subseries plot would consist of only data from January. This case is given as an example below.

Usage

```
gg_subseries(data, y = NULL, period = NULL, ...)
```

Arguments

`data` A tidy time series object (tsibble)

`y` The variable to plot (a bare expression). If NULL, it will automatically selected from the data.

`period` The seasonal period to display. If NULL (default), the largest frequency in the data is used. If numeric, it represents the frequency times the interval between observations. If a string (e.g., "1y" for 1 year, "3m" for 3 months, "1d" for 1 day, "1h" for 1 hour, "1min" for 1 minute, "1s" for 1 second), it's converted to a Period class object from the lubridate package. Note that the data must have at least one observation per seasonal period, and the period cannot be smaller than the observation interval.

`...` Additional arguments passed to geom_line()

Details

The horizontal lines are used to represent the mean of each facet, allowing easy identification of seasonal differences between seasons. This plot is particularly useful in identifying changes in the seasonal pattern over time.

similar to a seasonal plot (`gg_season()`), and

Value

A ggplot object showing a seasonal subseries plot of a time series.

References

Hyndman and Athanasopoulos (2019) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. https://OTexts.com/fpp3/

Examples

```
library(tsibble)
library(dplyr)
tsibbledata::aus_retail %>%
  filter(
    State == "Victoria",
    Industry == "Cafes, restaurants and catering services"
  ) %>%
  gg_subseries(Turnover)
```

---

gg_tsdisplay *Ensemble of time series displays*

Description

Plots a time series along with its ACF along with an customisable third graphic of either a PACF, histogram, lagged scatterplot or spectral density.

Usage

```
gg_tsdisplay(
  data,
  y = NULL,
  plot_type = c("auto", "partial", "season", "histogram", "scatter", "spectrum"),
  lag_max = NULL
)
```

Arguments

data	A tidy time series object (tsibble)
y	The variable to plot (a bare expression). If NULL, it will automatically selected from the data.
plot_type	type of plot to include in lower right corner. By default ("auto") a season plot will be shown for seasonal data, a spectrum plot will be shown for non-seasonal data without missing values, and a PACF will be shown otherwise.
lag_max	maximum lag at which to calculate the acf. Default is 10*log10(N/m) where N is the number of observations and m the number of series. Will be automatically limited to one less than the number of observations in the series.

Value

A list of ggplot objects showing useful plots of a time series.

Author(s)

Rob J Hyndman & Mitchell O'Hara-Wild

References

Hyndman and Athanasopoulos (2019) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. https://OTexts.com/fpp3/

See Also

plot.ts, feasts::ACF(), spec.ar

Examples

```
library(tsibble)
library(dplyr)
tsibbledata::aus_retail %>%
  filter(
    State == "Victoria",
    Industry == "Cafes, restaurants and catering services"
  ) %>%
  gg_tsdisplay(Turnover)
```

gg_tsresiduals *Ensemble of time series residual diagnostic plots*

Description

Plots the residuals using a time series plot, ACF and histogram.

Usage

```
gg_tsresiduals(data, type = "innovation", plot_type = "histogram", ...)
```

Arguments

data	A mable containing one model with residuals.
type	The type of residuals to compute. If type="response", residuals on the back-transformed data will be computed.
plot_type	type of plot to include in lower right corner. By default ("auto") a season plot will be shown for seasonal data, a spectrum plot will be shown for non-seasonal data without missing values, and a PACF will be shown otherwise.
...	Additional arguments passed to gg_tsdisplay().

Value

A list of ggplot objects showing a useful plots of a time series model's residuals.

References

Hyndman and Athanasopoulos (2019) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. https://OTexts.com/fpp3/

See Also

gg_tsdisplay()

Examples

```
if (requireNamespace("fable", quietly = TRUE)) {
library(fable)

tsibbledata::aus_production %>%
  model(ETS(Beer)) %>%
  gg_tsresiduals()
}
```

Index