Package: prettyPanelMatch (via r-universe)

May 26, 2026

Title 'ggplot2'-Based Visualization for 'PanelMatch' (Imai, Kim, Wang 2023) Results

Version 0.2.0

Description Provides 'ggplot2'-based plotting functions for 'PanelMatch' (Imai, Kim, Wang (2023) <doi:10.1111/ajps.12685>) results. Tidy-and-plot function pairs for three 'PanelMatch' outputs: treatment effect estimates (PanelEstimate()), placebo test diagnostics (placebo_test()), and covariate balance checks (get_covariate_balance()). Supports multiple models, significance indicators, faceted balance grids, and full 'ggplot2' extensibility.

URL `https://github.com/jacqpark/prettyPanelMatch`

BugReports `https://github.com/jacqpark/prettyPanelMatch/issues`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports ggplot2 (>= 3.4.0), dplyr, rlang

Suggests PanelMatch, testthat (>= 3.0.0)

Repository https://jacqpark.r-universe.dev

Date/Publication 2026-03-27 12:05:19 UTC

RemoteUrl https://github.com/jacqpark/prettypanelmatch

RemoteRef HEAD

RemoteSha d9e33312d9fa4eb166492782503c5d489863bf73

Contents

`gg_covariate_balance` *Create a ggplot2 Covariate Balance Plot*

Description

Produces a faceted covariate balance plot showing standardized mean differences across matching stages. Dependent variables are drawn as black solid lines; covariates as grey lines with distinct linetypes.

Usage

```
gg_covariate_balance(
  data,
  dv_color = "black",
  cov_color = "grey70",
  dv_linetype = "solid",
  cov_linetypes = NULL,
  hline = 0,
  ylim = c(-2, 2),
  xlab = "Time",
  ylab = "SD",
  title = NULL,
  subtitle = NULL,
  show_legend = FALSE,
  strip_text_y_size = 8.5,
  theme_fn = ggplot2::theme_bw
)

## S3 method for class 'ppm_cov_tidy'
autoplot(object, ...)
```

Arguments

`data`	A `ppm_cov_tidy` data frame from `pretty_covariate_balance`.
`dv_color`	Color for DV lines. Default `"black"`.
`cov_color`	Color for covariate lines. Default `"grey70"`.
`dv_linetype`	Linetype for DV lines. Default `"solid"`.
`cov_linetypes`	Character vector of linetypes for covariates. If `NULL` (default), cycles through `"dashed"`, `"twodash"`, `"dotted"`, `"dotdash"`, `"longdash"`.
`hline`	Y-intercept for reference line. Default 0. Set to `NULL` to remove.

`ylim`	Y-axis limits as a length-2 numeric vector. Default `c(-2, 2)`. Set to `NULL` for automatic limits.
`xlab`	X-axis label. Default `"Time"`.
`ylab`	Y-axis label. Default `"SD"`.
`title`	Optional plot title.
`subtitle`	Optional plot subtitle.
`show_legend`	Logical. Show legend? Default `FALSE`.
`strip_text_y_size`	Font size for row strip labels. Default 8.5. Set to `NULL` to use the theme default.
`theme_fn`	A ggplot2 theme function. Default `theme_bw`.
`object`	A `ppm_cov_tidy` data frame.
`...`	Additional arguments passed to `gg_covariate_balance`.

Details

The default layout uses `facet_grid(model ~ stage)`, where models (different DVs / subsamples) form the rows and matching stages form the columns, reproducing the standard PanelMatch covariate-balance diagnostic.

Value

A `ggplot` object that can be further customized with standard ggplot2 syntax.

See Also

`pretty_covariate_balance` to prepare the input data, `ggplot_panel_estimate` for treatment effect plots, `gg_placebo_test` for placebo test plots.

Examples

```
# Toy example (runs without PanelMatch)
toy <- data.frame(
  model    = factor("Model A", ordered = TRUE),
  stage    = factor("Before matching", ordered = TRUE),
  time     = factor(rep(paste0("t-", 3:1), 2),
                    levels = paste0("t-", 3:1), ordered = TRUE),
  variable = factor(rep(c("outcome", "covar1"), each = 3), ordered = TRUE),
  estimate = c(0.3, 0.5, 0.8, -0.1, 0.2, 0.1),
  is_dv    = rep(c(TRUE, FALSE), each = 3),
  stringsAsFactors = FALSE
)
class(toy) <- c("ppm_cov_tidy", "data.frame")
gg_covariate_balance(toy)
```

`gg_placebo_test` *Create a ggplot2 Coefficient Plot from Placebo Test Results*

Description

Produces a customizable ggplot2 coefficient plot for placebo test estimates from the PanelMatch package. This is a convenience wrapper around `ggplot_panel_estimate` with defaults tailored for placebo tests (e.g., y-axis label set to `"Placebo estimate"`).

Usage

```
gg_placebo_test(data, ylab = "Placebo estimate", ...)

## S3 method for class 'ppm_placebo_tidy'
autoplot(object, ...)
```

Arguments

`data`	A `ppm_placebo_tidy` data frame from `pretty_placebo_test`.
`ylab`	Y-axis label. Default `"Placebo estimate"`.
`...`	Additional arguments passed to `gg_placebo_test`.
`object`	A `ppm_placebo_tidy` data frame.

Details

All arguments are passed through to `ggplot_panel_estimate()`, so the full range of customization (shapes, dodging, significance coding, faceting, themes) is available.

Value

A `ggplot` object.

See Also

`pretty_placebo_test` to prepare the input data, `ggplot_panel_estimate` for treatment effect plots.

Examples

```
# Toy example (runs without PanelMatch)
toy <- data.frame(
  term      = factor(paste0("t-", 3:1), levels = paste0("t-", 3:1), ordered = TRUE),
  estimate  = c(0.1, -0.05, 0.02),
  std.error = c(0.15, 0.12, 0.10),
  conf.low  = c(-0.19, -0.29, -0.18),
  conf.high = c(0.39, 0.19, 0.22),
  label     = factor("Model", ordered = TRUE),
  signif    = c("Non-signif", "Non-signif", "Non-signif"),
```

```
  stringsAsFactors = FALSE
)
class(toy) <- c("ppm_placebo_tidy", "data.frame")
gg_placebo_test(toy)
```

`ggplot_panel_estimate` *Create a ggplot2 Coefficient Plot from PanelEstimate Results*

Description

Produces a customizable ggplot2 coefficient plot showing point estimates and confidence intervals across lead periods. Supports multiple models with dodged positions and significance-based shape coding (hollow = not significant, filled = significant). A footnote is added by default.

Usage

```
ggplot_panel_estimate(
  data,
  dodge_width = 0.5,
  point_size = 2.2,
  errorbar_alpha = 0.5,
  errorbar_width = 0,
  shapes = NULL,
  show_signif_shapes = TRUE,
  legend_labels = NULL,
 footnote = "Filled markers denote statistical significance (CI excludes zero).",
  footnote_size = NULL,
  xlab = "Time",
  ylab = "ATT",
  title = NULL,
  subtitle = NULL,
  hline = 0,
  facet_by = NULL,
  theme_fn = ggplot2::theme_minimal
)

## S3 method for class 'ppm_tidy'
autoplot(object, ...)
```

Arguments

`data`	A `ppm_tidy` data frame from `tidy_panel_estimate`.
`dodge_width`	Width of position dodge for multiple models. Default 0.5.
`point_size`	Size of point estimates. Default 2.2.
`errorbar_alpha`	Alpha transparency for error bars. Default 0.5.

`errorbar_width`	Width of error bar caps. Default 0 (no caps).
`shapes`	A character vector of hollow shape names, one per model. Filled counterparts are paired automatically. If `NULL` (default), shapes cycle through: `"circle"`, `"diamond"`, `"triangle"`, `"square"`, `"triangle_down"`. Available names: `"circle"`, `"square"`, `"triangle"`, `"diamond"`, `"triangle_down"`. Numeric codes (0-14) are also accepted for advanced users.
`show_signif_shapes`	Logical. If `TRUE` (default), uses different shapes for significant vs. non-significant estimates. If `FALSE`, uses uniform shapes per model.
`legend_labels`	Optional character vector to override legend labels (one per model, in input order).
`footnote`	Character string for the significance footnote. Set to `NULL` to suppress. Default explains hollow vs. filled convention.
`footnote_size`	Font size for the footnote. Default `NULL`, which matches the axis title size from the active theme.
`xlab`	X-axis label. Default `"Time"`.
`ylab`	Y-axis label. Default `"ATT"`.
`title`	Optional plot title.
`subtitle`	Optional plot subtitle.
`hline`	Intercept for reference line. Default 0. Set to `NULL` to remove.
`facet_by`	Optional variable name to facet by (e.g., `"label"`). Default `NULL` (no faceting).
`theme_fn`	A ggplot2 theme function. Default `theme_minimal`.
`object`	A `ppm_tidy` data frame.
`...`	Additional arguments passed to `ggplot_panel_estimate`.

Details

The returned object is a standard `ggplot` object, so you can add any ggplot2 layers, scales, or themes on top of it.

Value

A `ggplot` object that can be further customized with standard ggplot2 syntax.

See Also

`tidy_panel_estimate` to prepare the input data, `gg_placebo_test` for placebo test plots, `gg_covariate_balance` for covariate balance plots.

Examples

```
# Toy example (runs without PanelMatch)
toy <- data.frame(
  term      = factor(paste0("t+", 0:2), levels = paste0("t+", 0:2), ordered = TRUE),
  estimate  = c(0.5, 0.8, 1.2),
  std.error = c(0.2, 0.3, 0.25),
```

```
  conf.low  = c(0.1, 0.2, 0.7),
  conf.high = c(0.9, 1.4, 1.7),
  label     = factor("Model", ordered = TRUE),
  signif    = c("Non-signif", "Signif", "Signif"),
  stringsAsFactors = FALSE
)
class(toy) <- c("ppm_tidy", "data.frame")
ggplot_panel_estimate(toy)
```

`pretty_covariate_balance` *Tidy Covariate Balance Matrices into a Data Frame*

Description

Converts `get_covariate_balance()` matrices from the PanelMatch package into a single tidy data frame suitable for ggplot2 plotting.

Usage

```
pretty_covariate_balance(
  ...,
  stage_labels = c("Before matching", "Matched, pre-refinement", "Post-refinement"),
  dv = NULL
)
```

Arguments

`...`	Named arguments where each value is a list of matrices from `get_covariate_balance()`, one per matching stage. Names become model labels (row facet labels). A single matrix can be passed directly instead of wrapping in a list.
`stage_labels`	Character vector naming the matching stages, in order. Must be at least as long as the longest list of matrices provided. Default: `c("Before matching", "Matched, pre-refinement", "Post-refinement")`.
`dv`	Character vector of variable names that are dependent variables. These are styled differently (black solid lines) in the plot. Variables not listed here are treated as covariates (grey, varied linetypes).

Details

Each named argument should be a list of 1–3 matrices (one per matching stage, e.g., before matching, after matching pre-refinement, post-refinement). The argument names become model/row facet labels.

Value

A data frame (with class `"ppm_cov_tidy"`) containing:

model Model label (ordered factor, row facet)

stage Matching stage label (ordered factor, column facet)

time Pre-treatment period label (ordered factor, e.g., "t-3")

variable Covariate or DV name (ordered factor)

estimate Standardized mean difference

is_dv Logical; `TRUE` for dependent variables

See Also

`gg_covariate_balance` to plot the result, `tidy_panel_estimate` for treatment effect estimates, `pretty_placebo_test` for placebo test results.

Examples

```
# Create a mock covariate balance matrix (rows = time, cols = variables)
mat <- matrix(
  c(0.3, -0.1, 0.5, 0.2, 0.8, 0.1),
  nrow = 3,
  dimnames = list(NULL, c("outcome", "covar1"))
)
pretty_covariate_balance(
  "My Model" = mat,
  stage_labels = "Before matching",
  dv = "outcome"
)
```

`pretty_placebo_test` *Tidy placebo_test Results into a Data Frame*

Description

Converts one or more `placebo_test()` results from the PanelMatch package into a single tidy data frame suitable for ggplot2 plotting.

Usage

```
pretty_placebo_test(..., labels = NULL, confidence_level = 0.95)
```

Arguments

`...`	One or more `placebo_test()` result objects. Can also be a single named list of results. Named arguments become model labels automatically.
`labels`	A character vector of labels, one per result. If `NULL` and inputs are named, the names are used. Otherwise defaults to `"Model"` (single) or `"Model 1"`, `"Model 2"`, etc.
`confidence_level`	Confidence level for constructing intervals. Default 0.95.

Value

A data frame (with class `"ppm_placebo_tidy"`) containing:

term Lag period label (e.g., "t-3", "t-2")

estimate Point estimate

std.error Standard error

conf.low Lower confidence bound

conf.high Upper confidence bound

label Model label (ordered factor preserving input order)

signif Whether the CI excludes zero ("Signif" or "Non-signif")

See Also

`gg_placebo_test` to plot the result, `tidy_panel_estimate` for treatment effect estimates, `pretty_covariate_balance` for covariate balance.

Examples

```
# Create a mock placebo_test result
pt <- list(
  estimates = c("t-3" = 0.10, "t-2" = -0.05, "t-1" = 0.02),
  standard.errors = c("t-3" = 0.15, "t-2" = 0.12, "t-1" = 0.10)
)
pretty_placebo_test(pt, labels = "My Model")

# Custom confidence level
pretty_placebo_test(pt, confidence_level = 0.90)
```

`tidy_panel_estimate` *Tidy PanelEstimate Summaries into a Data Frame*

Description

Converts one or more `summary(PanelEstimate(...))` objects into a single tidy data frame suitable for ggplot2 plotting.

Usage

```
tidy_panel_estimate(..., labels = NULL)
```

Arguments

`...`	One or more PanelEstimate summary objects (or raw PanelEstimate objects, which will be summarized automatically). Can also be a single named list of summaries.
`labels`	A character vector of labels, one per summary. Used to distinguish models in the plot legend. If `NULL` and inputs are named (either as named arguments or a named list), the names are used. Otherwise defaults to `"Model 1"`, `"Model 2"`, etc.

Value

A data frame (with class `"ppm_tidy"`) containing columns:

term Lead period label (e.g., "t+0", "t+1", ...)

estimate Point estimate (ATT)

std.error Standard error

conf.low Lower confidence bound

conf.high Upper confidence bound

label Model label (ordered factor preserving input order)

signif Whether the CI excludes zero ("Signif" or "Non-signif")

See Also

`ggplot_panel_estimate` to plot the result, `pretty_placebo_test` for placebo test results, `pretty_covariate_balance` for covariate balance.

Examples

```
# Create a mock PanelEstimate summary (matrix with 4 columns)
pe_sum <- matrix(
  c(0.5, 0.2, 0.1, 0.9,
    0.8, 0.3, 0.2, 1.4,
    1.2, 0.25, 0.7, 1.7),
```

```
  nrow = 3, byrow = TRUE,
  dimnames = list(NULL, c("Estimate", "Std.Error", "lower", "upper"))
)
tidy_panel_estimate(pe_sum, labels = "My Model")

# Multiple models with named arguments
pe_sum2 <- matrix(
  c(0.3, 0.15, 0.0, 0.6,
    0.6, 0.20, 0.2, 1.0,
    0.9, 0.18, 0.5, 1.3),
  nrow = 3, byrow = TRUE,
  dimnames = list(NULL, c("Estimate", "Std.Error", "lower", "upper"))
)
tidy_panel_estimate("Model A" = pe_sum, "Model B" = pe_sum2)
```

Index