

# Package ‘metaconfoundr’

May 8, 2026

**Title** Visualize 'Confounder' Control in Meta-Analyses

**Version** 0.1.2

**Description** Visualize 'confounder' control in meta-analysis.

'metaconfoundr' is an approach to evaluating bias in studies used in meta-analyses based on the causal inference framework. Study groups create a causal diagram displaying their assumptions about the scientific question. From this, they develop a list of important 'confounders'. Then, they evaluate whether studies controlled for these variables well. 'metaconfoundr' is a toolkit to facilitate this process and visualize the results as heat maps, traffic light plots, and more.

**License** MIT + file LICENSE

**URL** <https://github.com/malcolmbarrett/metaconfoundr>

**BugReports** <https://github.com/malcolmbarrett/metaconfoundr/issues>

**Depends** R (>= 3.5.0)

**Imports** dplyr (>= 1.0.0), forcats, ggplot2 (>= 3.4.0), magrittr, purrr, rlang (>= 0.1.2), shiny, stringr, tibble, tidyr (>= 1.0.0), tidyselect

**Suggests** covr, knitr, metafor, patchwork, readr, rio, rmarkdown, roxygen2, spelling, testthat (>= 3.0.0), vdiff

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Malcolm Barrett [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-0299-5825>>),  
Julie M. Petersen [aut] (ORCID:

<<https://orcid.org/0000-0001-7845-4545>>),  
 Ludovic Trinquart [aut] (ORCID:  
 <<https://orcid.org/0000-0002-3028-4900>>)

**Maintainer** Malcolm Barrett <malcolmbarrett@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-01-17 19:00:09 UTC

## Contents

count_non_confounders . . . . .	2
facet_constructs . . . . .	3
geom_cochrane . . . . .	3
ipi_data . . . . .	5
label_robins . . . . .	6
launch_metaconfoundr_app . . . . .	6
mc_detect_layout . . . . .	7
mc_heatmap . . . . .	8
metaconfoundr() . . . . .	9
scale_fill_cochrane . . . . .	10
score_control . . . . .	11
summarize_control_quality . . . . .	11
theme_mc . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

count\_non\_confounders *Count and plot non-confounders*

---

## Description

Count and plot non-confounders

## Usage

```
count_non_confounders(.df)
```

```
plot_non_confounders(.df, ..., geom = ggplot2::geom_col, sort = TRUE)
```

## Arguments

.df	A data frame, usually the result of metaconfoundr()
...	Arguments passed to geom
geom	The ggplot2 geom to use
sort	Logical. Should the results be sorted?

**Value**

a tibble or ggplot

**Examples**

```
ipi %>%  
  metaconfoundr() %>%  
  plot_non_confounders(size = 3, geom = ggplot2::geom_point)
```

---

facet_constructs	<i>Facet by constructs</i>
------------------	----------------------------

---

**Description**

A helper function to facet by constructs in `mc_heatmap()` and `mc_trafficlight()`

**Usage**

```
facet_constructs(...)
```

**Arguments**

... Arguments passed to `ggplot2::facet_grid()`

**Value**

a facet component

**See Also**

Other plots: `geom_cochrane()`, `mc_heatmap()`, `scale_fill_cochrane()`, `theme_mc()`

---

geom_cochrane	<i>Add Cochrane-style symbols to heatmaps and traffic light plots</i>
---------------	---

---

**Description**

Add Cochrane-style symbols to heatmaps and traffic light plots

**Usage**

```
geom_cochrane(
  mapping = ggplot2::aes(shape = control_quality),
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

**Arguments**

mapping	Set of aesthetic mappings created by <a href="#">aes()</a> or <a href="#">aes_()</a> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <a href="#">ggplot()</a> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <a href="#">fortify()</a> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code> ).
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to <a href="#">layer()</a> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired <code>geom/stat</code> .
na.rm	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <a href="#">borders()</a> .

**Value**

a geom

**See Also**

Other plots: [facet\\_constructs\(\)](#), [mc\\_heatmap\(\)](#), [scale\\_fill\\_cochrane\(\)](#), [theme\\_mc\(\)](#)

---

ipi\_data

*IPI data*

---

**Description**

These data represent 14 analyses (retrospective cohorts and sibling-matched designs) to evaluate the association between short interpregnancy interval (<6 months versus 18-23 months) and risk of preterm birth (<37 weeks gestation) and the adequacy of confounder control (Petersen et al.). Adequacy of confounder control was determined overall for each study as well as by variable and construct (groupings of conceptually related variables). These studies are a subset of studies originally identified in a systematic review by Ahrens et al. to summarize associations between short interpregnancy interval and a variety of perinatal outcomes in high-resource settings.

**Usage**

ipi

ipi\_wide

ipi\_metaanalysis

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 407 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 37 rows and 14 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 11 rows and 9 columns.

**Source**

Ahrens et al. (2019)

**References**

Ahrens KA, Nelson H, Stidd RL, Moskosky S, Hutcheon JA. Short interpregnancy intervals and adverse perinatal outcomes in high-resource settings: An updated systematic review. *Paediatr Perinat Epidemiol.* 2019;33(1):O25-O47.

Petersen JM, Barrett M, Ahrens K, Murray EJ, Hogue C, Mumford S, Bryant Mantha A, Fox MP, Gadupudi S, Trinquart L. Confounder Matrix: A Tool to Assess Confounding Bias in Systematic Reviews of Observational Studies. 2020

---

`label_robins`*Label values using ROBINS approach*

---

**Description**

`label_robins()` is a helper function to modify metaconfoundr labels to use ROBINS-like labels: low risk, some concerns, high risk.

**Usage**

```
label_robins()
```

**Value**

a character vector of ROBINS labels

**Examples**

```
mc_heatmap(metaconfoundr(ipi)) +  
  ggplot2::scale_fill_ordinal(labels = label_robins())
```

```
mc_heatmap(metaconfoundr(ipi)) +  
  scale_fill_cochrane(labels = label_robins())
```

---

`launch_metaconfoundr_app`*Launch metaconfoundr Shiny app*

---

**Description**

`launch_metaconfoundr_app()` launches a Shiny app to create visualizations of confounding control in meta-analyses

**Usage**

```
launch_metaconfoundr_app()
```

**Value**

A Shiny app

---

mc\_detect\_layout      *Tidy metaconfoundr data layouts*

---

### Description

mc\_longer() and mc\_wider() are helper functions to put `metaconfoundr()` for long and wide data sets, respectively. results into a tidy format. mc\_detect\_layout() chooses between the two automatically based on the number of variables in the data frame. mc\_study\_values() helps standardize evaluations of control quality.

### Usage

```
mc_detect_layout(...)
```

```
mc_longer(
  study = contains("construct"),
  construct = contains("construct"),
  variable = matches("variable|factor"),
  control_quality = contains("control_quality"),
  is_confounder = contains("confounder"),
  study_values = mc_study_values()
)
```

```
mc_study_values(inadequate = 0, some_concerns = 1, adequate = 2)
```

```
mc_wider(
  construct = contains("construct"),
  variable = matches("variable|factor"),
  is_confounder = contains("confounder"),
  study = everything(),
  study_values = mc_study_values()
)
```

### Arguments

...	Additional arguments passed to mc_wider() or mc_longer()
study	The column with the name of the studies
construct	The domain or construct column
variable	The column that describes the confounding variables
control_quality	The column that describes the confounding control quality
is_confounder	The column that describes if a variable is a confounder
study_values	What are the levels of control_quality? Use mc_study_values() to set up.
inadequate	Which value signifies inadequate control?
some_concerns	Which value signifies control with some concerns?
adequate	Which value signifies adequate control?

**Value**

a function that tidies the data

---

mc_heatmap	<i>Plot a heatmap or traffic light plot of metaconfoundr() summaries</i>
------------	--

---

**Description**

mc\_heatmap() and mc\_trafficlight() visualize the results of metaconfoundr(), summarizing the quality of confounder control in each study.

**Usage**

```
mc_heatmap(
  .df,
  legend_title = "control quality",
  sort = FALSE,
  by_group = FALSE,
  score = c("adequate", "sum", "controlled"),
  non_confounders = FALSE
)
```

```
mc_trafficlight(
  .df,
  size = 8,
  legend_title = "control quality",
  sort = FALSE,
  by_group = FALSE,
  score = c("adequate", "sum", "controlled"),
  non_confounders = FALSE
)
```

**Arguments**

.df	A data frame, usually the result of metaconfoundr()
legend_title	The legend title
sort	Logical. Sort by confounder score? Calculated by <a href="#">score_control()</a>
by_group	Logical. If sorted, sort within domain?
score	The approach used to calculate the score. adequate tests if the study controlled at a strictly adequate level. sum treats control_quality as an ordinal integer, summing it's values such that a higher score has better control overall. controlled tests if any control, including some concerns control, is present.
non_confounders	Logical. Include non-confounders? Default is FALSE.
size	The size of the points in the traffic light plot

**Value**

a ggplot

**See Also**

Other plots: [facet\\_constructs\(\)](#), [geom\\_cochrane\(\)](#), [scale\\_fill\\_cochrane\(\)](#), [theme\\_mc\(\)](#)

**Examples**

```
ipi %>%
  metaconfoundr() %>%
  dplyr::mutate(variable = stringr::str_wrap(variable, 10)) %>%
  mc_heatmap() +
  theme_mc() +
  facet_constructs() +
  ggplot2::guides(x = ggplot2::guide_axis(n.dodge = 2))
```

```
ipi %>%
  metaconfoundr() %>%
  mc_trafficlight() +
  geom_cochrane() +
  facet_constructs() +
  scale_fill_cochrane() +
  theme_mc() +
  ggplot2::guides(x = ggplot2::guide_axis(n.dodge = 2))
```

---

metaconfoundr()

*Prepare a meta-analysis data set for metaconfoundr*

---

**Description**

`metaconfoundr()` standardizes data frames with information on how well a set of studies control for a set of variables. In this approach, a set of domain experts agree on the variables that are required to properly control for confounding for a scientific question. Then, for a given confounder, the studies are described as being adequately controlled, inadequately controlled, or controlled with some concerns. `metaconfoundr()` is intended to standardize data for use in [mc\\_heatmap\(\)](#) and [mc\\_trafficlight\(\)](#). See the vignette on data preparation for more information on how to set up your evaluation.

**Usage**

```
metaconfoundr(.df, data_format = mc_detect_layout())
```

**Arguments**

`.df` A data frame. See the vignette on data preparation for more details.

`data_format` The format of the data. Detected automatically by default, but explicit options include [mc\\_longer\(\)](#) and [mc\\_wider\(\)](#)

**Value**

a tibble

**Examples**

```
metaconfoundr(ipi)

metaconfoundr(ipi_wide)

ipi_wide2 <- ipi_wide %>%
  dplyr::rename(scope = construct)

metaconfoundr(ipi_wide2, mc_wider(construct = "scope"))
```

---

scale\_fill\_cochrane *Add Cochrane-style palettes to ggplots*

---

**Description**

Add Cochrane-style palettes to ggplots

**Usage**

```
scale_fill_cochrane(...)

scale_color_cochrane(...)

scale_shape_cochrane(...)
```

**Arguments**

... Arguments passed to the underline scale function

**Value**

scales for ggplot

**See Also**

Other plots: [facet\\_constructs\(\)](#), [geom\\_cochrane\(\)](#), [mc\\_heatmap\(\)](#), [theme\\_mc\(\)](#)

---

score_control	<i>Add a score of confounding control</i>
---------------	---

---

**Description**

score\_control() adds a variable, score, that summarizes how well a study controls for a domain or construct. Used to sort heatmaps and traffic light plots.

**Usage**

```
score_control(.df, score = c("adequate", "sum", "controlled"))
```

**Arguments**

.df	A data frame, usually the result of <code>metaconfoundr()</code>
score	The approach used to calculate the score. adequate tests if the study controlled at a strictly adequate level. sum treats control_quality as an ordinal integer, summing it's values such that a higher score has better control overall. controlled tests if any control, including some concerns control, is present.

**Value**

a tibble

**Examples**

```
library(dplyr)

ipi %>%
  metaconfoundr() %>%
  filter(is_confounder == "Y") %>%
  score_control("controlled") %>%
  arrange(desc(score))
```

---

summarize_control_quality	<i>Summarize the control quality of studies</i>
---------------------------	---

---

**Description**

summarize\_control\_quality() allows you to summarize how well studies control for variables within one or more domains, and how well those domains are controlled for overall. Each logical statement is a domain and can be named.

**Usage**

```
summarize_control_quality(.df, ..., domains = TRUE)
```

**Arguments**

.df	A data frame, usually the result of <code>metaconfoundr()</code>
...	Boolean arguments to declare adequate control logic
domains	Logical. Include the domains in the output? If FALSE, only returns overall control quality.

**Value**

A tibble

**Examples**

```
summary_df <- summarize_control_quality(
  metaconfoundr(ipi),
  Sociodemographics = `Maternal age` & `Race/ethnicity` & `Marital status`,
  Socioeconomics = `SES category` | Insurance & Education,
  "Reproductive Hx" = `Prior pregnancy outcome`
)

summary_df

summary_df %>%
  mc_trafficlight() +
  theme_mc() +
  facet_constructs() +
  geom_cochrane() +
  scale_fill_cochrane()
```

---

theme\_mc

*A minimal theme for metaconfoundr plots*

---

**Description**

A minimal theme for metaconfoundr plots

**Usage**

```
theme_mc(base_size = 14)
```

**Arguments**

base_size	base font size, given in pts.
-----------	-------------------------------

**Value**

a ggplot theme

**See Also**

Other plots: [facet\\_constructs\(\)](#), [geom\\_cochrane\(\)](#), [mc\\_heatmap\(\)](#), [scale\\_fill\\_cochrane\(\)](#)

# Index

- \* **datasets**
  - ipi\_data, [5](#)
- \* **plots**
  - facet\_constructs, [3](#)
  - geom\_cochrane, [3](#)
  - mc\_heatmap, [8](#)
  - scale\_fill\_cochrane, [10](#)
  - theme\_mc, [12](#)
- aes(), [4](#)
- aes\_(), [4](#)
- borders(), [4](#)
- count\_non\_confounders, [2](#)
- facet\_constructs, [3](#), [5](#), [9](#), [10](#), [13](#)
- fortify(), [4](#)
- geom\_cochrane, [3](#), [3](#), [9](#), [10](#), [13](#)
- ggplot(), [4](#)
- ggplot2::facet\_grid(), [3](#)
- ipi (ipi\_data), [5](#)
- ipi\_data, [5](#)
- ipi\_metaanalysis (ipi\_data), [5](#)
- ipi\_wide (ipi\_data), [5](#)
- label\_robins, [6](#)
- launch\_metaconfoundr\_app, [6](#)
- layer(), [4](#)
- mc\_detect\_layout, [7](#)
- mc\_heatmap, [3](#), [5](#), [8](#), [10](#), [13](#)
- mc\_heatmap(), [3](#), [9](#)
- mc\_longer (mc\_detect\_layout), [7](#)
- mc\_longer(), [9](#)
- mc\_study\_values (mc\_detect\_layout), [7](#)
- mc\_trafficlight (mc\_heatmap), [8](#)
- mc\_trafficlight(), [3](#), [9](#)
- mc\_wider (mc\_detect\_layout), [7](#)
- mc\_wider(), [9](#)
- metaconfoundr (metaconfoundr()), [9](#)
- metaconfoundr(), [7](#), [9](#), [11](#), [12](#)
- plot\_non\_confounders
  - (count\_non\_confounders), [2](#)
- scale\_color\_cochrane
  - (scale\_fill\_cochrane), [10](#)
- scale\_fill\_cochrane, [3](#), [5](#), [9](#), [10](#), [13](#)
- scale\_shape\_cochrane
  - (scale\_fill\_cochrane), [10](#)
- score\_control, [11](#)
- score\_control(), [8](#)
- summarize\_control\_quality, [11](#)
- theme\_mc, [3](#), [5](#), [9](#), [10](#), [12](#)