Package 'autoMR'

March 29, 2026

Type Package

Title Automated Mendelian Randomization Workflows and Visualizations

Version 0.1.1

Description Provides tools to summarize, analyze, and visualize results
from Mendelian randomization studies using summarized genetic
association data. The package includes functions for generating forest
plots and scatter plots at the single-nucleotide polymorphism
and Mendelian randomization method levels, and for fitting multiple
estimators in a unified workflow, including inverse-variance weighted
estimation, Mendelian randomization Egger regression,
the weighted median estimator, the robust adjusted
profile score, Mendelian randomization pleiotropy residual sum
and outlier, Mendelian randomization with the genotype
recoding invariance property, and a Bayesian horseshoe
method. Related methods are described by Burgess (2013)
<doi:10.1002/gepi.21758>, Bowden (2015) <doi:10.1093/ije/dyv080>,
Bowden (2016) <doi:10.1002/gepi.21965>, Zhao (2020)
<doi:10.1214/19-AOS1866>, Verbanck (2018)
<doi:10.1038/s41588-018-0099-7>, Dudbridge (2025)
<doi:10.1371/journal.pgen.1011967>, and Grant and Burgess (2024)
<doi:10.1016/j.ajhg.2023.12.002>. Related open-source software
includes 'TwoSampleMR' <https://github.com/MRCIEU/TwoSampleMR>,
'mr.raps' <https://github.com/qingyuanzhao/mr.raps>,
'MR-PRESSO' <https://github.com/rondolab/MR-PRESSO>, and
'MR-Horse' <https://github.com/aj-grant/mrhorse>.

Depends R (>= 4.2)

Imports ggplot2 (>= 4.0.1), dplyr (>= 1.1.4), tidyr (>= 1.3.1),
MendelianRandomization (>= 0.10.0), R2jags (>= 0.8-9), coda (>=
0.19-4.1), nortest (>= 1.0-4), grDevices, graphics, stats,
utils, methods

URL https://github.com/KelinZhong/autoMR

BugReports https://github.com/KelinZhong/autoMR/issues

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Kelin Zhong [aut, cre],
    Chia-Ling Kuo [aut]

Maintainer Kelin Zhong <kelin.zhong@uconn.edu>

Contents

---

export_forest_plots _Export MR Forest Plots to Disk_

Description

Saves plots stored in a GWASForestPlots or MRForestPlots object to a directory using ggplot2::ggsave().
You can export all plots or filter to a specific outcome, a specific exposure, or a specific outcome-exposure pair.

Usage

```
export_forest_plots(
  object,
  save_dir = tempdir(),
  file_type = "pdf",
  width = 8,
```

```
  height = 6,
  dpi = 300,
  outcome = NULL,
  exposure = NULL
)

## S4 method for signature 'GWASForestPlots'
export_forest_plots(
  object,
  save_dir = tempdir(),
  file_type = "pdf",
  width = 8,
  height = 6,
  dpi = 300,
  outcome = NULL,
  exposure = NULL
)

## S4 method for signature 'MRForestPlots'
export_forest_plots(
  object,
  save_dir = tempdir(),
  file_type = "pdf",
  width = 8,
  height = 6,
  dpi = 300,
  outcome = NULL,
  exposure = NULL
)
```

Arguments

`object`	A `GWASForestPlots` or `MRForestPlots` object returned by `GWAS_forest()` or `MR_forest()`.
`save_dir`	Character string; directory to write files into. Must already exist. Defaults to `tempdir()`.
`file_type`	Character string; output format passed to `ggplot2::ggsave()`. One of `"pdf"`, `"png"`, `"jpeg"`, or `"tiff"`. Default is `"pdf"`.
`width`	Numeric; plot width in inches. Default is 8.
`height`	Numeric; plot height in inches. Default is 6.
`dpi`	Integer; resolution in dots per inch (ignored for PDF). Default is `300`.
`outcome`	Optional character string; if supplied, only plots whose outcome matches this value are exported.
`exposure`	Optional character string; if supplied, only plots whose exposure matches this value are exported.

Details

Filtering logic:

- **Both** outcome **and** exposure **supplied** — exports the single plot for that exact pair.
- outcome **only** — exports all exposures for that outcome.
- exposure **only** — exports all outcomes for that exposure.
- **Neither supplied** — exports every plot in the object.

File names are prefixed with Instrument_forest_ for GWASForestPlots objects and Method_forest_ for MRForestPlots objects, followed by <Outcome>_<Exposure>.<file_type> with spaces replaced by underscores.

Value

Invisibly returns object so calls can be chained.

Methods (by class)

- export_forest_plots(GWASForestPlots): Method for GWASForestPlots.
- export_forest_plots(MRForestPlots): Method for MRForestPlots.

Examples

```
data("fi_49item")
input1   <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(MR_input_data = input1)

gwas_plots <- GWAS_forest(MR_input_data = input1, report_form = "Beta")
mr_plots   <- MR_forest(summary_df = outcome1, effect = "Beta")

# Inspect what is stored; for fi_49item both objects print:
#   [1] fi_49item :: Zn

# Retrieve the exact outcome/exposure labels stored in the object
gwas_out <- gwas_plots@outcomes[1]    # "fi_49item"
gwas_exp <- gwas_plots@exposures[1]   # "Zn"
mr_out   <- mr_plots@outcomes[1]      # "fi_49item"
mr_exp   <- mr_plots@exposures[1]     # "Zn"

# Export all instrument-level plots as PDF (commented — writes to disk)
export_forest_plots(gwas_plots, save_dir = tempdir())

# Export one outcome only
export_forest_plots(gwas_plots, save_dir = tempdir(), outcome = gwas_out)

# Export one exposure only
export_forest_plots(mr_plots, save_dir = tempdir(), exposure = mr_exp)

# Export one specific pair
export_forest_plots(mr_plots, save_dir = tempdir(), outcome = mr_out, exposure = mr_exp)
```

---

export_scatter_plots   *Export MR Scatter Plots to Disk*

Description

Saves plots stored in an `MRScatterPlots` object to a directory. You can export all plots or filter to a specific outcome, a specific exposure, or a specific outcome-exposure pair.

Usage

```
export_scatter_plots(
  object,
  save_dir = tempdir(),
  file_type = "pdf",
  width = 8,
  height = 6,
  outcome = NULL,
  exposure = NULL
)

## S4 method for signature 'MRScatterPlots'
export_scatter_plots(
  object,
  save_dir = tempdir(),
  file_type = "pdf",
  width = 8,
  height = 6,
  outcome = NULL,
  exposure = NULL
)
```

Arguments

object	An `MRScatterPlots` object returned by `plot_mr_scatter()`.
save_dir	Character string; directory to write files into. Must exist. Defaults to `tempdir()`.
file_type	Character string; output format passed to the corresponding `grDevices` device. One of `"pdf"`, `"png"`, `"jpeg"`, or `"tiff"`. Default is `"pdf"`.
width	Numeric; plot width in inches. Default is 8.
height	Numeric; plot height in inches. Default is 6.
outcome	Optional character string; if supplied, only plots whose outcome matches this value are exported.
exposure	Optional character string; if supplied, only plots whose exposure matches this value are exported.

Details

Filtering logic:

- **Both** outcome **and** exposure **supplied** — exports the single plot for that exact pair.

- outcome **only** — exports all exposures for that outcome.

- exposure **only** — exports all outcomes for that exposure.

- **Neither supplied** — exports every plot in the object.

File names follow the pattern Scatter_<Outcome>_<Exposure>.<file_type> with spaces replaced by underscores.

Value

Invisibly returns object so calls can be chained.

Methods (by class)

- export_scatter_plots(MRScatterPlots): Method for MRScatterPlots.

Examples

```
data("fi_49item")
input1   <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(MR_input_data = input1)
plots    <- plot_mr_scatter(MR_input_data = input1, summary_df = outcome1)

# Inspect what is stored; for fi_49item this prints:
#   [1] fi_49item :: Zn
#' # Retrieve the exact outcome/exposure labels stored in the object
out_name <- plots@outcomes[1]   # "fi_49item"
exp_name <- plots@exposures[1]  # "Zn"

# Export all plots to tempdir() (commented — writes to disk)
export_scatter_plots(plots, save_dir = tempdir())

# Export one outcome only
export_scatter_plots(plots, save_dir = tempdir(), outcome = out_name)

# Export one exposure only
export_scatter_plots(plots, save_dir = tempdir(), exposure = exp_name)

# Export one specific pair
export_scatter_plots(plots, save_dir = tempdir(), outcome = out_name, exposure = exp_name)
```

---

fi_49item *Frailty Index Dataset (49-items)*

Description

A dataset containing variables used to compute a 49-item frailty index (FI). The exact variables/columns depend on your project definitions.

Format

A data frame.

---

format_mr_input *Format vectors into a Mendelian Randomization input data frame*

Description

Format vectors into a Mendelian Randomization input data frame

Usage

```
format_mr_input(
  Instrument,
  beta_exposure,
  se_exposure,
  beta_outcome,
  se_outcome,
  Outcome,
  Exposure,
  ALLELE1 = NULL,
  ALLELE0 = NULL,
  A1FREQ = NULL
)
```

Arguments

Instrument	Character vector of instrument/SNP identifiers.
beta_exposure	Numeric vector of exposure effects.
se_exposure	Numeric vector of exposure standard errors.
beta_outcome	Numeric vector of outcome effects.
se_outcome	Numeric vector of outcome standard errors.
Outcome	Character string or vector for outcome names (Mandatory).
Exposure	Character string or vector for exposure names (Mandatory).
ALLELE1	Optional character vector for non-effect alleles.
ALLELE0	Optional character vector for effect alleles.
A1FREQ	Optional numeric vector for effect allele frequencies.

Value

A data frame formatted for MR analysis.

Examples

```
data("fi_49item")

input_test1 <- format_mr_input(
  Instrument = fi_49item$Instrument,
  beta_exposure = fi_49item$beta_exposure,
  se_exposure = fi_49item$se_exposure,
  beta_outcome = fi_49item$beta_outcome,
  se_outcome = fi_49item$se_outcome,
  Outcome = fi_49item$Outcome,
  Exposure = fi_49item$Exposure
)
head(input_test1)

input_test2 <- format_mr_input(
  Instrument = fi_49item$Instrument,
  beta_exposure = fi_49item$beta_exposure,
  se_exposure = fi_49item$se_exposure,
  beta_outcome = fi_49item$beta_outcome,
  se_outcome = fi_49item$se_outcome,
  Outcome = fi_49item$Outcome,
  Exposure = fi_49item$Exposure,
  ALLELE1 = fi_49item$NEA_outcome,
  ALLELE0 = fi_49item$EA_outcome,
  A1FREQ = fi_49item$A1FREQ_outcome
)
head(input_test2)
```

---

fried_frailty	*Fried Frailty Phenotype Dataset*

Description

A dataset containing variables used to compute Fried frailty phenotype measures. The exact variables/columns depend on your project definitions.

Format

A data frame.

GWASForestPlots-class   *S4 class to store instrument-level MR forest plots*

Description

A `GWASForestPlots` object is returned by `GWAS_forest()`. It holds one ggplot object per outcome-exposure pair — each plot shows per-instrument (SNP-level) causal estimates alongside a pooled IVW estimate — together with the metadata needed to filter and export subsets.

Slots

`plots` Named list of `ggplot` objects. Names follow the pattern `"Outcome::Exposure"`.

`outcomes` Character vector of outcome labels, parallel to `plots`.

`exposures` Character vector of exposure labels, parallel to `plots`.

Methods

`export_forest_plots(object, save_dir, file_type, width, height, dpi, outcome, exposure)`
    Saves plots to disk. Optionally filter by `outcome`, `exposure`, or both to export a subset.

GWAS_forest   *Generate Instrument-level Forest Plots for Mendelian Randomization*

Description

Creates a forest plot for each instrument (SNP) within the MR analysis, including a pooled IVW estimate at the top for comparison.

Usage

```
GWAS_forest(
  MR_input_data,
  report_form,
  save_plot = FALSE,
  file_type = "jpeg",
  save_dir = tempdir(),
  xlim_custom = NULL,
  dot_size = 2,
  axis_text_size = 10,
  axis_title_size = 12,
  digits = 2,
  plot_width = 8,
  plot_height = 6,
  label_text_size = 3
)
```

Arguments

`MR_input_data`	Harmonised MR input data frame. Must contain Outcome, Exposure, Instrument, beta_exposure, se_exposure, beta_outcome, and se_outcome columns.
`report_form`	Character string or vector indicating the standard output scale for each outcome (e.g., "Beta", "OR", "HR"). Defaults to "Beta".
`save_plot`	Deprecated. Plots are no longer saved inside the function. Use `export_forest_plots()` on the returned object instead. Ignored with a warning if supplied as `TRUE`.
`file_type`	Deprecated. Passed to `export_forest_plots()` instead. Ignored with a warning if supplied.
`save_dir`	Deprecated. Passed to `export_forest_plots()` instead. Ignored with a warning if supplied.
`xlim_custom`	Optional numeric vector of length 2 providing custom limits for the x-axis. If NULL, limits are determined by the data.
`dot_size`	Numeric value specifying the size of the points. Default is 2.
`axis_text_size`	Numeric value specifying the font size for axis labels. Default is 10.
`axis_title_size`	Numeric value specifying the font size for axis titles. Default is 12.
`digits`	Integer specifying the number of decimal places for labels. Default is 2.
`plot_width`	Numeric value specifying the width of exported plot file. Default is 8.
`plot_height`	Numeric value specifying the height of exported plot file. Default is 6.
`label_text_size`	Numeric value specifying the size of estimate labels (Beta/OR/HR and 95 percent CI) shown on the plot. Default is 3.

Value

A `GWASForestPlots` object containing one `ggplot` per outcome-exposure pair, with instrument-level (SNP) causal estimates and a pooled IVW estimate. Use `export_forest_plots()` to write plots to disk with optional filtering.

Examples

```
data("fi_49item")
input1 <- harmonize_mr_data(df = fi_49item)

gwas_plots <- GWAS_forest(
  MR_input_data   = input1,
  report_form     = "Beta",
  xlim_custom     = NULL,
  dot_size        = 2,
  axis_text_size  = 10,
  axis_title_size = 12,
  digits          = 2,
  plot_width      = 8,
  plot_height     = 6,
  label_text_size = 3
```

```
)

# Retrieve the exact outcome/exposure labels stored in the object
out_name <- gwas_plots@outcomes[1]   # "fi_49item"
exp_name <- gwas_plots@exposures[1]  # "Zn"

# Export all instrument-level plots as PDF (commented – writes to disk)
export_forest_plots(gwas_plots, save_dir = tempdir(), file_type = "pdf")

# Export plots for one outcome only
export_forest_plots(gwas_plots, save_dir = tempdir(), outcome = out_name)

# Export plots for one exposure only
export_forest_plots(gwas_plots, save_dir = tempdir(), exposure = exp_name)

# Export one specific outcome-exposure pair
export_forest_plots(gwas_plots, save_dir = tempdir(), outcome = out_name, exposure = exp_name)
```

---

harmonize_mr_data *Harmonize exposure and outcome SNP data*

Description

Harmonizes alleles and ensures consistent effect direction between exposure and outcome datasets.
Standardizes exposure effects to be positive.

Usage

```
harmonize_mr_data(df, Outcome = NULL, Exposure = NULL)
```

Arguments

df	Data frame containing required columns (beta_outcome, se_outcome, beta_exposure, se_exposure) and metadata (Outcome, Exposure).
Outcome	Optional; Character string to set as the Outcome name.
Exposure	Optional; Character string to set as the Exposure name.

Value

A harmonized data frame with 'Instrument' and allele columns.

Examples

```
input_test3 <- harmonize_mr_data(df = fi_49item)
head(input_test3)
```

MRForestPlots-class   *S4 class to store method-level MR forest plots*

Description

An `MRForestPlots` object is returned by `MR_forest()`. It holds one `ggplot` object per outcome-exposure pair — each plot compares causal estimates across MR methods (IVW, RAPS, Egger, etc.) — together with the metadata needed to filter and export subsets.

Slots

`plots` Named list of `ggplot` objects. Names follow the pattern `"Outcome::Exposure"`.

`outcomes` Character vector of outcome labels, parallel to `plots`.

`exposures` Character vector of exposure labels, parallel to `plots`.

Methods

`export_forest_plots(object, save_dir, file_type, width, height, dpi, outcome, exposure)`
    Saves plots to disk. Optionally filter by `outcome`, `exposure`, or both to export a subset.

MRScatterPlots-class   *S4 class to store MR scatter plots*

Description

An `MRScatterPlots` object is returned by `plot_mr_scatter()`. It holds one plot-parameter list per outcome-exposure pair, together with the metadata needed to filter and export subsets of plots.

Slots

`plots` Named list of plot-parameter lists. Names follow the pattern `"Outcome::Exposure"`.

`outcomes` Character vector of outcome labels, parallel to `plots`.

`exposures` Character vector of exposure labels, parallel to `plots`.

Methods

`export_scatter_plots(object, save_dir, file_type, width, height, outcome, exposure)`
    Saves plots to disk. Optionally filter by `outcome`, `exposure`, or both to export a subset.

MR_forest   *Generate Forest Plots across Multiple MR Methods*

Description

Creates a forest plot comparing causal estimates across different MR methods (e.g., IVW, RAPS, Egger).

Usage

```
MR_forest(
  summary_df,
  effect,
  save_plot = FALSE,
  file_type = "jpeg",
  save_dir = tempdir(),
  custom_xlim = NULL,
  dot_size = 3,
  axis_text_size = 10,
  axis_title_size = 12,
  pval_text_size = 3,
  plot_width = 8,
  plot_height = 6,
  clamp_nonpositive = FALSE
)
```

Arguments

summary_df	MR results data frame, typically the output from run_mr_analysis().
effect	Character string or vector indicating the effect scale ("Beta", "OR", or "HR").
save_plot	Deprecated. Plots are no longer saved inside the function. Use export_forest_plots() on the returned object instead. Ignored with a warning if supplied as TRUE.
file_type	Deprecated. Passed to export_forest_plots() instead. Ignored with a warning if supplied.
save_dir	Deprecated. Passed to export_forest_plots() instead. Ignored with a warning if supplied.
custom_xlim	Optional numeric vector of length 2 for x-axis limits.
dot_size	Numeric value specifying the point size. Default is 3.
axis_text_size	Numeric value for axis font size.
axis_title_size	Numeric value for title font size.
pval_text_size	Numeric value for p-value label size.
plot_width	Width of exported plot.
plot_height	Height of exported plot.

clamp_nonpositive

> Logical; whether non-positive estimates should be clamped to a small positive value before log-transformation.

Value

An `MRForestPlots` object containing one `ggplot` per outcome-exposure pair, with causal estimates compared across MR methods. Use export_forest_plots() to write plots to disk with optional filtering.

Examples

```
data("fi_49item")
input1   <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(
  MR_input_data     = input1,
  outcome.form      = NULL,
  use_ivw           = TRUE,
  use_raps          = TRUE,
  use_median        = TRUE,
  use_egger         = TRUE,
  use_mr_presso     = TRUE,
  use_mr_horse      = TRUE,
  use_mr_grip       = TRUE,
  NbDistribution    = 1000,
  SignifThreshold   = 0.05,
  mr_horse_n_iter   = 5000,
  mr_horse_n_burnin = 1000,
  mr_grip_parameters = NULL
)

mr_plots <- MR_forest(
  summary_df        = outcome1,
  effect            = "Beta",
  custom_xlim       = NULL,
  dot_size          = 3,
  axis_text_size    = 10,
  axis_title_size   = 12,
  pval_text_size    = 3,
  plot_width        = 8,
  plot_height       = 6,
  clamp_nonpositive = FALSE
)

# Retrieve the exact outcome/exposure labels stored in the object
out_name <- mr_plots@outcomes[1]   # "fi_49item"
exp_name <- mr_plots@exposures[1]  # "Zn"

# Export all method-level plots as PDF (commented - writes to disk)
export_forest_plots(mr_plots, save_dir = tempdir(), file_type = "pdf")

# Export plots for one outcome only
export_forest_plots(mr_plots, save_dir = tempdir(), outcome = out_name)
```

```
# Export plots for one exposure only
export_forest_plots(mr_plots, save_dir = tempdir(), exposure = exp_name)

# Export one specific outcome-exposure pair
export_forest_plots(mr_plots, save_dir = tempdir(), outcome = out_name, exposure = exp_name)
```

| plot_mr_scatter | *Plot MR Scatter Plots for Multiple Outcomes and Exposures* |

Description

Generates one scatter plot per outcome-exposure pair using base R graphics, with a regression line overlaid for each requested Mendelian Randomization (MR) method. Plot parameters are stored in an `MRScatterPlots` S4 object and rendered on demand at export time, so no files are written to disk during this call.

Usage

```
plot_mr_scatter(
  MR_input_data,
  plot.xlab = "Exposure",
  plot.ylab = "Outcome",
  methods.plot = c("IVW", "RAPS", "Egger", "PRESSO", "Horse", "GRIP"),
  NbDistribution_presso = 1000,
  SignifThreshold_presso = 0.05,
  mr_horse_n_iter = 5000,
  mr_horse_n_burnin = 1000,
  show.legend = TRUE,
  summary_df = NULL,
  effect_scale = "Beta",
  use_df_results = TRUE
)
```

Arguments

MR_input_data   Harmonised MR input data frame. Must contain `Outcome` and `Exposure` columns.

plot.xlab   Character string; prefix for the x-axis label. Default is `"Exposure"`.

plot.ylab   Character string; prefix for the y-axis label. Default is `"Outcome"`.

methods.plot   Character vector of MR methods to overlay as regression lines. Supported values: `"IVW"`, `"RAPS"`, `"Egger"`, `"PRESSO"`, `"Horse"`, `"GRIP"`.

NbDistribution_presso

Integer; number of simulated distributions for on-the-fly MR-PRESSO calculation. Default is `1000`.

SignifThreshold_presso

> Numeric; significance threshold for on-the-fly MR-PRESSO outlier test. Default is `0.05`.

mr_horse_n_iter

> Integer; number of Markov chain Monte Carlo (MCMC) iterations for on-the-fly MR-Horse. Default is `5000`.

mr_horse_n_burnin

> Integer; number of MCMC burn-in samples for on-the-fly MR-Horse. Default is `1000`.

show.legend
: Logical; whether to annotate each plot with method labels, beta estimates, and p-values. Default is `TRUE`.

summary_df
: Optional data frame of pre-calculated results from `run_mr_analysis()`. When supplied together with `use_df_results = TRUE`, avoids re-running the analysis.

effect_scale
: Character string matching the scale used in `summary_df`: `"Beta"`, `"OR"`, or `"HR"`. Default is `"Beta"`.

use_df_results
: Logical; if `TRUE` and `summary_df` is provided, method slopes are read from `summary_df` instead of being re-calculated. Default is `TRUE`.

Value

An MRScatterPlots object containing one `recordedplot` per outcome-exposure pair, together with outcome and exposure metadata. Use `export_scatter_plots()` to write plots to disk with optional filtering by outcome, exposure, or both.

Examples

```
data("fi_49item")
input1  <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(
  MR_input_data    = input1,
  outcome.form     = "Beta",
  use_ivw          = TRUE,
  use_raps         = FALSE,
  use_median       = FALSE,
  use_egger        = FALSE,
  use_mr_presso    = FALSE,
  use_mr_horse     = FALSE,
  use_mr_grip      = FALSE,
  NbDistribution   = 1000,
  SignifThreshold  = 0.05,
  mr_horse_n_iter  = 5000,
  mr_horse_n_burnin = 1000,
  mr_grip_parameters = NULL
)

# Pass pre-calculated results to avoid rerunning the analysis
plots <- plot_mr_scatter(
  MR_input_data = input1,
  summary_df    = outcome1,
  use_df_results = TRUE
```

```
)

# Inspect the object; for fi_49item this prints:
#   [1] fi_49item :: Zn

# Retrieve the exact outcome/exposure labels stored in the object
out_name <- plots@outcomes[1]   # "fi_49item"
exp_name <- plots@exposures[1]  # "Zn"

# Export all plots as PDF (commented — writes to disk)
export_scatter_plots(plots, save_dir = tempdir(), file_type = "pdf")

# Export one outcome only
export_scatter_plots(plots, save_dir = tempdir(), outcome = out_name)

# Export one exposure only
export_scatter_plots(plots, save_dir = tempdir(), exposure = exp_name)

# Export one specific pair
# export_scatter_plots(plots, save_dir = tempdir(), outcome = out_name, exposure = exp_name)

# Export as PNG instead
export_scatter_plots(plots, save_dir = tempdir(), file_type = "png")
```

---

run_mr_analysis   *Run MR Analysis for Multiple Outcomes*

Description

Performs causal inference analysis using multiple Mendelian Randomization (MR) methods across
one or more outcomes and exposures. Returns a combined results data frame. To save the output,
use standard R functions such as write.csv() or saveRDS() on the returned object.

Usage

```
run_mr_analysis(
  MR_input_data,
  outcome.form = NULL,
  use_ivw = TRUE,
  use_raps = TRUE,
  use_median = TRUE,
  use_egger = TRUE,
  use_mr_presso = TRUE,
  use_mr_horse = TRUE,
  use_mr_grip = TRUE,
  NbDistribution = 1000,
  SignifThreshold = 0.05,
```

```
  mr_horse_n_iter = 5000,
  mr_horse_n_burnin = 1000,
  mr_grip_parameters = NULL
)
```

Arguments

`MR_input_data`	Harmonised MR input data frame. Must contain Outcome and Exposure columns.
`outcome.form`	Character vector indicating the effect scale for each outcome: `"Beta"`, `"OR"` (odds ratio), or `"HR"` (hazard ratio). A single value is recycled across all outcomes. Defaults to `"Beta"`.
`use_ivw`	Logical; whether to run the Inverse Variance Weighted (IVW) method. Default is `TRUE`.
`use_raps`	Logical; whether to run the Robust Adjusted Profile Score (MR-RAPS) method. Default is `TRUE`.
`use_median`	Logical; whether to run the Weighted Median method. Default is `TRUE`.
`use_egger`	Logical; whether to run MR-Egger regression. Default is `TRUE`.
`use_mr_presso`	Logical; whether to run the Mendelian Randomization Pleiotropy RESidual Sum and Outlier (MR-PRESSO) method. Default is `TRUE`.
`use_mr_horse`	Logical; whether to run the MR-Horse method. Default is `TRUE`.
`use_mr_grip`	Logical; whether to run the Generalized Regression with Instrument Pairs (MR-GRIP) method. Default is `TRUE`.
`NbDistribution`	Integer; number of simulated distributions for MR-PRESSO. Default is `1000`.
`SignifThreshold`	Numeric; significance threshold for the MR-PRESSO outlier test. Default is `0.05`.
`mr_horse_n_iter`	Integer; number of Markov chain Monte Carlo (MCMC) iterations for MR-Horse. Default is `5000`.
`mr_horse_n_burnin`	Integer; number of MCMC burn-in samples for MR-Horse. Default is `1000`.
`mr_grip_parameters`	List of additional parameters passed to MR-GRIP. If `NULL`, default parameters are used.

Value

A data frame combining results across all outcomes and exposures. Each row represents one outcome-exposure pair. Columns include estimates, confidence intervals (CI), and p-values for each method, together with diagnostic flags (e.g., F-statistic below 10, significant heterogeneity). Use `write.csv()` or `saveRDS()` to save the returned object.

Examples

```
data("fi_49item")
input1 <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(
  MR_input_data     = input1,
  outcome.form      = "Beta",
  use_ivw           = TRUE,
  use_raps          = FALSE,
  use_median        = FALSE,
  use_egger         = FALSE,
  use_mr_presso     = FALSE,
  use_mr_horse      = FALSE,
  use_mr_grip       = FALSE,
  NbDistribution    = 1000,
  SignifThreshold   = 0.05,
  mr_horse_n_iter   = 5000,
  mr_horse_n_burnin = 1000,
  mr_grip_parameters = NULL
)


data("fried_frailty")
input2 <- harmonize_mr_data(df = fried_frailty)
outcome2 <- run_mr_analysis(
  MR_input_data     = input2,
  outcome.form      = "OR",
  use_ivw           = TRUE,
  use_raps          = TRUE,
  use_median        = TRUE,
  use_egger         = TRUE,
  use_mr_presso     = TRUE,
  use_mr_horse      = TRUE,
  use_mr_grip       = TRUE,
  NbDistribution    = 1000,
  SignifThreshold   = 0.05,
  mr_horse_n_iter   = 5000,
  mr_horse_n_burnin = 1000,
  mr_grip_parameters = NULL
)
```

---

showplot   *Display MR Plots on Screen*

Description

Renders plots stored in an MRScatterPlots, GWASForestPlots, or MRForestPlots object to the active graphics device. You can display all plots or filter to a specific outcome, exposure, or outcome-exposure pair.

Usage

```
showplot(object, outcome = NULL, exposure = NULL)

## S4 method for signature 'MRScatterPlots'
showplot(object, outcome = NULL, exposure = NULL)

## S4 method for signature 'GWASForestPlots'
showplot(object, outcome = NULL, exposure = NULL)

## S4 method for signature 'MRForestPlots'
showplot(object, outcome = NULL, exposure = NULL)
```

Arguments

object	An MRScatterPlots, GWASForestPlots, or MRForestPlots object.
outcome	Optional character string; if supplied, only plots whose outcome matches this value are displayed.
exposure	Optional character string; if supplied, only plots whose exposure matches this value are displayed.

Details

Filtering logic:

- **Both** outcome **and** exposure **supplied** — displays the single plot for that exact pair.
- outcome **only** — displays all exposures for that outcome.
- exposure **only** — displays all outcomes for that exposure.
- **Neither supplied** — displays every plot in the object.

Value

Invisibly returns object.

Methods (by class)

- showplot(MRScatterPlots): Display scatter plots from an MRScatterPlots object.
- showplot(GWASForestPlots): Display forest plots from a GWASForestPlots object.
- showplot(MRForestPlots): Display forest plots from an MRForestPlots object.

Examples

```
data("fi_49item")
input1   <- harmonize_mr_data(df = fi_49item)
outcome1 <- run_mr_analysis(MR_input_data = input1)

# Scatter plots
scatter <- plot_mr_scatter(MR_input_data = input1, summary_df = outcome1)
showplot(scatter)
```

```
showplot(scatter, outcome = "fi_49item")
showplot(scatter, outcome = "fi_49item", exposure = "Zn")

# GWAS forest plots
gwas_plots <- GWAS_forest(MR_input_data = input1, report_form = "Beta")
showplot(gwas_plots)
showplot(gwas_plots, exposure = "Zn")

# MR forest plots
mr_plots <- MR_forest(summary_df = outcome1, effect = "Beta")
showplot(mr_plots)
showplot(mr_plots, outcome = "fi_49item", exposure = "Zn")
```

Index