Package: autoMR (via r-universe)

June 12, 2026

Type Package

Title Automated Mendelian Randomization Workflows and Visualizations

Version 1.2.0

Maintainer Kelin Zhong <`kelinzhonguconn@gmail.com`>

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

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

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

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

Date/Publication 2026-06-12 06:20:02 UTC

RemoteUrl https://github.com/cran/autoMR

RemoteRef HEAD

RemoteSha 42b9e4c4159e208938d8a23832717a75ded1e0e2

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 = "png",
  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 `"png"`.
`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.

Examples

```
data("fi_49item")
input1   <- harmonize_mr_data(df = fi_49item)$input_df
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 = "png",
  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 `"png"`.
`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.

Examples

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

Assembles raw vectors into a working data frame, aligns outcome alleles to the exposure strand (Step 1), then standardizes the sign of `beta_exposure` (Step 2). Returns both a full working data frame (`check_df`) and a slim, renamed input data frame (`input_df`), mirroring the output of `harmonize_mr_data()`.

Usage

```
format_mr_input(
  Instrument,
  beta_exposure,
  se_exposure,
  beta_outcome,
  se_outcome,
  Outcome,
  Exposure,
  ALLELE1 = NULL,
  ALLELE0 = NULL,
  A1FREQ = NULL,
  ALLELE1_outcome = NULL,
  ALLELE0_outcome = NULL,
  A1FREQ_outcome = NULL,
  beta_sign = c("positive", "negative")
)
```

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 (NEA) on the exposure strand (ALLELE1 = NEA_exposure).
`ALLELE0`	Optional character vector for effect alleles (EA) on the exposure strand (ALLELE0 = EA_exposure).
`A1FREQ`	Optional numeric vector for effect allele frequencies (exposure dataset).
`ALLELE1_outcome`	Optional character vector for non-effect alleles in the outcome dataset. If `NULL`, assumed identical to `ALLELE1`.
`ALLELE0_outcome`	Optional character vector for effect alleles in the outcome dataset. If `NULL`, assumed identical to `ALLELE0`.
`A1FREQ_outcome`	Optional numeric vector for effect allele frequencies in the outcome dataset.
`beta_sign`	Character string controlling the target sign for `beta_exposure`. One of `"positive"` (default, forces beta_exposure >= 0) or `"negative"` (forces beta_exposure <= 0). When a row is flipped, `beta_outcome` is negated as a side effect and all allele columns are swapped consistently. Ignored when no allele columns are supplied (strand identity is unknown without allele information).

Value

A named list with two elements:

`check_df` Full working data frame retaining all allele columns (NEA_exposure, EA_exposure, NEA_outcome, EA_outcome, A1FREQ_exposure, A1FREQ_outcome) after harmonization. Useful for quality-checking the harmonization results.

`input_df` Slim data frame ready for MR analysis, with columns: Instrument, beta_exposure, se_exposure, beta_outcome, se_outcome, Outcome, Exposure, ALLELE1, ALLELE0, A1FREQ.

Examples

```
data("fi_49item")

# Without allele columns
result1 <- 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(result1$input_df)

# With allele columns (enables alignment + sign standardization)
result2 <- 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_exposure,
  ALLELE0         = fi_49item$EA_exposure,
  A1FREQ          = fi_49item$A1FREQ_exposure,
  ALLELE1_outcome = fi_49item$NEA_outcome,
  ALLELE0_outcome = fi_49item$EA_outcome,
  A1FREQ_outcome  = fi_49item$A1FREQ_outcome
)
head(result2$check_df)
head(result2$input_df)

# Force all exposure betas to be negative
result3 <- 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_exposure,
  ALLELE0       = fi_49item$EA_exposure,
  beta_sign     = "negative"
)
head(result3$input_df)
```

`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.

`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 = "Beta",
  custom_xlim = NULL,
  dot_size = 2,
  axis_text_size = 10,
  axis_title_size = 12,
  digits = 2,
  label_text_size = 3,
  log_scale = FALSE
)
```

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".
`custom_xlim`	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.
`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.
`log_scale`	Logical; if `TRUE` and `report_form` is `"OR"` or `"HR"`, estimates are kept on the log scale instead of being exponentiated. The x-axis label becomes `log(OR)` or `log(HR)` and the reference line is drawn at 0. Default is `FALSE`.

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("merged_data")
input3 <- harmonize_mr_data(df = merged_data)$input_df

gwas_plots <- GWAS_forest(
  MR_input_data   = input3,
  report_form     = c("Beta","OR"),
  custom_xlim     = NULL,
  dot_size        = 2,
  axis_text_size  = 10,
  axis_title_size = 12,
  digits          = 2,
  label_text_size = 3,
  log_scale       = FALSE
)

# 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 PNG (commented - writes to disk)
# export_forest_plots(gwas_plots, save_dir = tempdir(), file_type = "jpeg")

# 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

Takes a data frame that already contains all required MR columns, aligns outcome alleles to the exposure strand (Step 1), then standardizes the sign of `beta_exposure` (Step 2). Returns both a full working data frame (`check_df`) and a slim, renamed input data frame (`input_df`), mirroring the output of `format_mr_input()`.

Usage

```
harmonize_mr_data(
  df,
  Outcome = NULL,
  Exposure = NULL,
  beta_sign = c("positive", "negative")
)
```

Arguments

`df`	Data frame containing required columns: `beta_exposure`, `se_exposure`, `beta_outcome`, `se_outcome`, `Outcome`, `Exposure`. Allele columns may be supplied as `NEA_exposure`/`EA_exposure`/`NEA_` or as `ALLELE1` (NEA) / `ALLELE0` (EA) which are treated as exposure alleles.
`Outcome`	Optional; character string to set as the Outcome name when the column is absent from `df`.
`Exposure`	Optional; character string to set as the Exposure name when the column is absent from `df`.
`beta_sign`	Character string controlling the target sign for `beta_exposure` after harmonization. One of `"positive"` (default, forces beta_exposure >= 0) or `"negative"` (forces beta_exposure <= 0). When a row is flipped, `beta_outcome` is negated as a side effect and all allele columns are swapped consistently.

Value

A named list with two elements:

- `check_df` Full working data frame retaining all allele columns after harmonization. Useful for quality-checking results.
- `input_df` Slim data frame ready for MR analysis, with columns: Instrument, beta_exposure, se_exposure, beta_outcome, se_outcome, Outcome, Exposure, ALLELE1, ALLELE0, A1FREQ.

Examples

```
result1 <- harmonize_mr_data(df = fi_49item)
head(result1$check_df)
head(result1$input_df)

# Force all exposure betas to be negative
result2 <- harmonize_mr_data(df = fi_49item, beta_sign = "negative")
head(result2$input_df)
```

`merged_data` *Merged Frailty Index Dataset (49-items and fried frailty)*

Description

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

Format

A data frame.

`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,
  custom_xlim = NULL,
  dot_size = 3,
  axis_text_size = 10,
  axis_title_size = 12,
  pval_text_size = 3,
  clamp_nonpositive = FALSE,
  log_scale = TRUE
)
```

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").
`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.

`clamp_nonpositive`	Logical; whether non-positive estimates should be clamped to a small positive value before log-transformation.
`log_scale`	Logical; if `TRUE` and `effect` is `"OR"` or `"HR"`, the x-axis label shows `log(OR)` or `log(HR)`. Note that `MR_forest` always plots on the log scale internally for OR/HR; this argument only controls the axis label. Default is `TRUE`.

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("merged_data")
input3   <- harmonize_mr_data(df = merged_data)$input_df
outcome3 <- run_mr_analysis(
  MR_input_data     = input3,
  outcome.form      = c("Beta","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
)

mr_plots <- MR_forest(
  summary_df        = outcome3,
  effect            = c("Beta","OR"),
  custom_xlim       = NULL,
  dot_size          = 3,
  axis_text_size    = 10,
  axis_title_size   = 12,
  pval_text_size    = 3,
  clamp_nonpositive = FALSE,
  log_scale         = TRUE
)

# 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 = "png")
```

```
# 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)
```

`mr_grip` *MR-GRIP: a modified MR-Egger model with the Genotype Recoding Invariance Property*

Description

Implements the modified MR-Egger model with the Genotype Recoding Invariance Property (MR-GRIP) due to Dudbridge and Bowden et al. (2025). The model multiplies each term of MR-Egger by the genotype-phenotype association, making results invariant to allele coding. The `parameters` argument is accepted for API compatibility but is not used; all weights and computations follow the original GRIP specification exactly.

Usage

```
mr_grip(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`parameters`	Unused; accepted for API compatibility only.

Value

List with elements: `b`, `se`, `pval` (causal estimate, SE, p-value); `b_i`, `se_i`, `pval_i` (intercept and its SE and p-value); `b.adj`, `se.adj`, `pval.adj` (weak-instrument-adjusted estimate); `nsnp`, `mod`, `smod`, `dat`.

`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"),
  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,
  custom_xlim = NULL,
  custom_ylim = NULL,
  symmetric_ylim = 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"`. When `effect_scale` is `"OR"` or `"HR"`, the label is automatically prefixed with `log(OR)` or `log(HR)`.
`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`.
`custom_xlim`	Optional numeric vector of length 2 for x-axis limits. If `NULL`, limits are determined from the data. Useful for excluding outlier instruments from the visible range without removing them from the analysis.
`custom_ylim`	Optional numeric vector of length 2 for y-axis limits. If `NULL`, limits are determined from the data.
`symmetric_ylim`	Logical; if `TRUE` (the default), the y-axis limits are forced to be symmetric around zero (the null on the beta / log scale) for every effect scale. If `FALSE`, Beta outcomes remain symmetric while OR/HR outcomes use the data range with 10% padding. Ignored when `custom_ylim` is supplied.

Value

An `MRScatterPlots` object containing one plot parameter list per outcome-exposure pair, together with outcome and exposure metadata. Each parameter list holds all data and settings needed to render the plot on demand — including `show.legend` — so settings like `show.legend = FALSE` are correctly applied at export time. Use `export_scatter_plots()` to write plots to disk with optional filtering by outcome, exposure, or both.

Examples

```
data("merged_data")
input3  <- harmonize_mr_data(df = merged_data)$input_df
outcome3 <- run_mr_analysis(
  MR_input_data     = input3,
  outcome.form      = c("Beta","OR"),
  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  = input3,
  summary_df     = outcome3,
  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`	Accepted for API compatibility; not used by MR-GRIP.

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)$input_df
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)$input_df
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
)



data("merged_data")
input3 <- harmonize_mr_data(df = merged_data)$input_df
outcome3 <- run_mr_analysis(
```

```
  MR_input_data     = input3,
  outcome.form     = c("Beta","OR"), ## First outcome use Beta and second outcome use 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)
```

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`.

Examples

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