Package: psrwe (via r-universe)

May 16, 2026

Title PS-Integrated Methods for Incorporating RWE in Clinical Studies

Version 3.2

Description High-quality real-world data can be transformed into scientific real-world evidence (RWE) for regulatory and healthcare decision-making using proven analytical methods and techniques. For example, propensity score (PS) methodology can be applied to pre-select a subset of real-world data containing patients that are similar to those in the current clinical study in terms of covariates, and to stratify the selected patients together with those in the current study into more homogeneous strata. Then, methods such as the power prior approach or composite likelihood approach can be applied in each stratum to draw inference for the parameters of interest. This package provides functions that implement the PS-integrated RWE analysis methods proposed in Wang et al. (2019) <doi:10.1080/10543406.2019.1657133>, Wang et al. (2020) <doi:10.1080/10543406.2019.1684309> and Chen et al. (2020) <doi:10.1080/10543406.2020.1730877>.

Depends methods, R (>= 4.0), rstan (>= 2.26.0), Rcpp (>= 1.0.5)

License GPL (>= 3)

URL `https://github.com/olssol/psrwe`

BugReports `https://github.com/olssol/psrwe/issues`

LinkingTo BH (>= 1.72.0-3), rstan (>= 2.26.0), Rcpp (>= 1.0.5), RcppEigen (>= 0.3.3.7.0), StanHeaders (>= 2.26.0), RcppParallel (>= 5.0.2)

Imports parallel (>= 3.2), cowplot (>= 1.0.0), dplyr (>= 0.8.5), ggplot2 (>= 3.3.2), randomForest (>= 4.6-14), survival, rstantools (>= 2.1.1)

Suggests knitr, rmarkdown

Encoding UTF-8

LazyData true

ByteCompile true

SystemRequirements GNU make

NeedsCompilation yes

RoxygenNote 7.2.3

VignetteBuilder knitr

Config/pak/sysreqs make

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

Date/Publication 2024-07-16 16:49:11 UTC

RemoteUrl https://github.com/olssol/psrwe

RemoteRef HEAD

RemoteSha e22c84c25f58594b2cf3bc53fbecfcbfb488fb2f

Contents

`psrwe-package` *PS-Integrated Methods for Incorporating RWE in Clinical Studies*

Description

This package provide R functions for conducting clinical studies with real-world evidence (RWE) incorporated in the study design and analysis.

PS-integrated power prior

We extend the Bayesian power prior approach for a single-arm study (the current study) to leverage external real-world data (RWD). We use propensity score methodology to pre-select a subset of real-world data containing patients that are similar to those in the current study in terms of covariates, and to stratify the selected patients together with those in the current study into more homogeneous strata. The power prior approach is then applied in each stratum to obtain stratum-specific posterior distributions, which are combined to complete the Bayesian inference for the parameters of interest.

PS-integrated composite likelihood

A propensity score-integrated composite likelihood (PSCL) approach is developed for cases in which the control arm of a two-arm randomized controlled trial (RCT) (treated vs. control) is augmented with patients from real-world data (RWD) containing both clinical outcomes and covariates at the patient-level. The PSCL approach first estimates the propensity score for every patient as the probability of the patient being in the RCT rather than the RWD, and then stratifies all patients into strata based on the estimated propensity scores. Within each propensity score stratum, a composite likelihood function is specified and utilized to down-weight the information contributed by the RWD source. Estimates of the stratum-specific parameters are obtained by maximizing the composite likelihood function. These stratum-specific estimates are then combined to obtain an overall population-level estimate of the parameter of interest.

References

Chen WC, Wang C, Li H, Lu N, Tiwari R, Xu Y, Yue LQ. Propensity score-integrated composite likelihood approach for augmenting the control arm of a randomized controlled trial by incorporating real-world data. *Journal of Biopharmaceutical Statistics*. 2020; 30(3):508-520.

Wang C, Lu N, Chen WC, Li H, Tiwari R, Xu Y, Yue LQ. Propensity score-integrated composite likelihood approach for incorporating real-world evidence in single-arm clinical studies. *Journal of Biopharmaceutical Statistics*. 2020; 30(3):495-507.

Wang C, Li H, Chen WC, Lu N, Tiwari R, Xu Y, Yue LQ. Propensity score-integrated power prior approach for incorporating real-world evidence in single-arm clinical studies. *Journal of Biopharmaceutical Statistics*. 2019; 29(5):731-748.

`ex_dta` *Example dataset*

Description

Example dataset of a single arm study.

Usage

```
data(ex_dta)
```

Format

A data frame with the following variables:

Group current, rwd

Y_Bin Binary outcome

Y_Con Continuous outcome

Y_Surv Survival outcome in days

Status Event status (0=alive, 1=dead)

V1-V7 Covariates

`ex_dta_rct` *Example dataset*

Description

Example dataset of a randomized study.

Usage

```
data(ex_dta_rct)
```

Format

A data frame with the following variables:

Group current, rwd

Arm control, treatment

Y_Con Continuous outcome

V1-V7 Covariates

`get_distance` *Distance between two distributions*

Description

Calculate difference measures using different metrics.

Usage

```
get_distance(
  cov0,
  cov1,
  metric = c("ovl", "ksd", "astd", "std", "abd", "ley", "mhb", "omkss")
)
```

Arguments

`cov0`	Vector (or matrix for `metric = "mhb"`) of samples from the first distribution.
`cov1`	Vector (or matrix for `metric = "mhb"`) of samples from the second distribution.
`metric`	Metric to use for calculating the distance with options:

`ovl`	Overlapping area(default)
`ksd`	Kullback-Leibler distance
`astd`	Standardized absolute mean difference
`std`	Standardized mean difference
`abd`	Absolute difference in means
`ley`	Levy distance
`mhb`	Mahalanobis distance
`omkss`	One minus Kolmogorov-Smirnov statistic

Value

A real value of the distance.

Examples

```
x <- rnorm(100,  mean = 0, sd = 1)
y <- rnorm(1000, mean = 1, sd = 2)
get_distance(x, y, "ovl")
get_distance(x, y, "abd")
```

`plot.PSRWE_DTA` *Plot PS distributions*

Description

S3 method for visualizing PS adjustment

Usage

```
## S3 method for class 'PSRWE_DTA'
plot(x, plot_type = c("ps", "balance", "diff"), ...)
```

Arguments

`x`	Class `RWE_DWITHPS` created by `psrwe_*` functions
`plot_type`	Types of plots. **ps** PS density plot **balance** Covariate balance plot **diff** Standardized mean differences, metric = std or astd
`...`	Additional parameter for the plot

`plot.PSRWE_DTA_MAT` *Plot PS distributions*

Description

S3 method for visualizing PS adjustment based on matching.

Usage

```
## S3 method for class 'PSRWE_DTA_MAT'
plot(x, ...)
```

Arguments

`x`	A list of class `PSRWE_DTA_MAT` that is generated using the `psrwe_match` function.
`...`	Parameters for `plot.PSRWE_DAT`

See Also

`plot.PSRWE_DTA`

`plot.PSRWE_RST` *Plot estimation results for power prior approach*

Description

S3 method plotting estimation results

Usage

```
## S3 method for class 'PSRWE_RST'
plot(x, ...)
```

Arguments

`x`	A list of class `PSRWE_RST` that is generated using the `psrwe_powerp`, `psrwe_compl`, or `psrwe_survkm` function.
`...`	Additional parameters.

`print.PSRWE_BOR` *Print borrow information*

Description

Print summary information of borrowing

Usage

```
## S3 method for class 'PSRWE_BOR'
print(x, ...)
```

Arguments

`x`	A list of class `PSRWE_BOR` that is generated using the `psrwe_borrow` function.
`...`	Additional parameters

See Also

`psrwe_borrow`

`print.PSRWE_DTA` *Print PS estimation results*

Description

Print summary information of PS estimation results

Usage

```
## S3 method for class 'PSRWE_DTA'
print(x, ...)
```

Arguments

`x`	A list of class `PSRWE_DTA` that is generated using the `psrwe_est` function.
`...`	Parameters for `summery.PSRWE_DTA`

See Also

`summary.PSRWE_DTA`

`print.PSRWE_DTA_MAT` *Print PS estimation results*

Description

Print summary information of PS estimation results

Usage

```
## S3 method for class 'PSRWE_DTA_MAT'
print(x, ...)
```

Arguments

`x`	A list of class `PSRWE_DTA_MAT` that is generated using the `psrwe_match` function.
`...`	Additional parameters

See Also

`summary.PSRWE_DTA_MAT`

`print.PSRWE_RST` *Print estimation results*

Description

Print summary information of outcome mean estimation results

Usage

```
## S3 method for class 'PSRWE_RST'
print(x, ...)
```

Arguments

`x`	A list of class `PSRWE_RST` that is generated using the `psrwe_powerp`, `psrwe_compl`, or `psrwe_survkm` function.
`...`	Additional parameters

See Also

`summary.PSRWE_RST`

`print.PSRWE_RST_OUTANA` *Print outcome analysis results*

Description

Print detail information of outcome analysis results

Usage

```
## S3 method for class 'PSRWE_RST_OUTANA'
print(x, show_details = FALSE, show_rct = FALSE, show_pred_tps = NULL, ...)
```

Arguments

`x`	A list of class `PSRWE_RST_OUTANA` that is generated using the `psrwe_outana` function.
`show_details`	Print out more observed summary
`show_rct`	Print out more analysis summary for RCT arms
`show_pred_tps`	Specified time points to be shown
`...`	Additional parameters

`psrwe_borrow` *Get number of subjects borrowed from each statum*

Description

Based on PS distances or number of current control subjects, split the total number of subjects to be borrowed from the external data source to each stratum

Usage

```
psrwe_borrow(
  dtaps,
  total_borrow,
  method = c("distance", "inverse_distance", "n_current", "n_external"),
  .drop_arg_fml = FALSE,
  ...
)
```

Arguments

`dtaps`	A class `PSRWE_DTA` or `PSRWE_DTA_MAT` object.
`total_borrow`	Total number of subjects to be borrowed
`method`	Method to split `total_borrow` for a class `PSRWE_DTA` object, which can be based on distance (`method = "distance"`) or inverse distance (`method = "inverse_distance"`). Other possible options include `"n_current"` and `"n_external"` that use the proportion of stratum sample size based on the current and external data, respectively. Ignored for class `PSRWE_DTA_MAT` object.
`.drop_arg_fml`	internal use to drop arguments and call, this is only used in cjk.
`...`	Additional parameters for `summary.PSRWE_DTA`.

Value

A class `PSRWE_BORR` list. It appends the following items to the `dtaps`:

Proportion Proportion splitting the number of total borrow among strata.

N_Borrow The number of to be borrowed subjects in each stratum.

Alpha Weight parameter value in each stratum.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
                    v_covs = paste("V", 1:7, sep = ""),
                    v_grp = "Group",
                    cur_grp_level = "current")
```

```
ps_borrow <- psrwe_borrow(total_borrow = 20, dta_ps)
ps_borrow

## Use different similarity metric
ps_borrow_omkss <- psrwe_borrow(total_borrow = 20, dta_ps,
                                metric = "omkss")
ps_borrow_omkss
```

`psrwe_ci` *Confidence/Credible Interval for PS-Integrated Estimation*

Description

Estimate the confidence/credible interval for the PS-integrated approach.

Usage

```
psrwe_ci(
  dta_psrst,
  method_ci = c("wald", "wilson"),
  conf_int = 0.95,
  conf_type = c("log_log", "plain"),
  ...
)
```

Arguments

`dta_psrst`	A returned object with class `PSRWE_EST`
`method_ci`	A method name for confidence interval (default wald)
`conf_int`	A two-sided level of confidence/credible limits (default 0.95)
`conf_type`	A type name of transformation for the confidence interval of PSKM approach
`...`	Other options

Details

`method_ci = "wilson"` is for binary outcomes only. `conf_type = "log_log"` is for `ps_km` only.

Value

A list with class name `PSRWE_EST`.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
ps_rst <- psrwe_compl(ps_borrow, v_outcome = "Y_Con")
rst <- psrwe_ci(ps_rst)
rst
```

`psrwe_compl` *PS-Integrated Composite Likelihood Estimation*

Description

Estimate the mean of the outcome based on PS-integrated composite likelihood approach. Variance is estimated by Jack-Knife method. Applies to the case when there is only one external data source.

Usage

```
psrwe_compl(
  dta_psbor,
  v_outcome = "Y",
  outcome_type = c("continuous", "binary"),
  stderr_method = c("jk", "sjk", "cjk", "sbs", "cbs", "none"),
  n_bootstrap = 200,
  ...
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`v_outcome`	Column name corresponding to the outcome.
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`stderr_method`	Method for computing StdErr, see Details
`n_bootstrap`	Number of bootstrap samples (for bootstrap stderr)
`...`	Parameters for `rwe_cl`

Details

`stderr_method` include `jk` as default using Jackknife method within each stratum, `sjk` for simple Jackknife method for combined estimates such as point estimates in single arm or treatment effects in RCT, or `cjk` for complex Jackknife method including refitting PS model, matching, trimming, calculating borrowing parameters, and combining overall estimates. Note that `sjk` may take a while longer to finish and `cjk` will take even much longer to finish. The `sbs` and `cbs` is for simple and complex Bootstrap methods.

Value

A data frame with class name `PSRWE_RST`. It contains the composite estimation of the mean for each stratum as well as the jackknife estimation for each subject. The results can be further summarized by its S3 method `summary`. The results can be also analyzed by `psrwe_outana` for outcome analysis and inference.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
rst       <- psrwe_compl(ps_borrow, v_outcome = "Y_Con")
rst
```

`psrwe_compl_watt` *PS-Integrated Composite Likelihood Estimation (WATT)*

Description

Estimate the mean of the outcome based on PS-integrated composite likelihood approach with weights of ATT (WATT). Variance is estimated by Jack-Knife method. Applies to the case when there is only one external data source.

Usage

```
psrwe_compl_watt(
  dta_psbor,
  v_outcome = "Y",
  outcome_type = c("continuous", "binary"),
  stderr_method = c("jk", "sjk", "cjk", "sbs", "cbs", "none"),
  n_bootstrap = 200,
  ...
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`v_outcome`	Column name corresponding to the outcome.
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`stderr_method`	Method for computing StdErr, see Details
`n_bootstrap`	Number of bootstrap samples (for bootstrap stderr)
`...`	Parameters for `rwe_cl_watt`

Details

`stderr_method` include `jk` as default using Jackknife method within each stratum, `sjk` for simple Jackknife method for combined estimates such as point estimates in single arm or treatment effects in RCT, or `cjk` for complex Jackknife method including refitting PS model, matching, trimming, calculating borrowing parameters, and combining overall estimates. Note that `sjk` may take a while longer to finish and `cjk` will take even much longer to finish. The `sbs` and `cbs` is for simple and complex Bootstrap methods.

Value

A data frame with class name `PSRWE_RST`. It contains the composite estimation of the mean for each stratum as well as the jackknife estimation for each subject. The results can be further summarized by its S3 method `summary`. The results can be also analyzed by `psrwe_outana` for outcome analysis and inference.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current",
       nstrata = 1)
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
rst       <- psrwe_compl_watt(ps_borrow, v_outcome = "Y_Bin")
rst
```

`psrwe_est` *Estimate propensity scores*

Description

Estimate propensity scores using logistic regression or random forest model.

Usage

```
psrwe_est(
  data,
  ps_fml = NULL,
  ps_method = c("logistic", "randomforest"),
  v_covs = "V1",
  v_grp = "Group",
  cur_grp_level = 1,
  v_arm = NULL,
  ctl_arm_level = NULL,
  stra_ctl_only = TRUE,
  nstrata = 5,
```

```
  trim_ab = c("both", "above", "below", "none"),
  .drop_arg_fml = FALSE,
  ...
)
```

Arguments

`data`	Data frame with group assignment and covariates.
`ps_fml`	Propensity score (PS) formula. If `NULL`, all covariates will be included in the PS model in a linear form.
`ps_method`	Method to calculate propensity scores. Can be set to `logistic` for logistic regression or `randomforest` for a random forest approach.
`v_covs`	Column names corresponding to covariates.
`v_grp`	Column name corresponding to group assignment.
`cur_grp_level`	Group level for the current study. Default is `cur_grp_level = 1`. Ignored for single arm studies.
`v_arm`	Column name corresponding to arm assignment.
`ctl_arm_level`	Arm level for the control arm. Ignored for single-arm studies.
`stra_ctl_only`	Create strata by control arm patients only. Default `TRUE`. Ignored by single arm studies. For randomized studies, when `stra_ctl_only` is `FALSE`, strata are created based on the PS scores of the entire current study patients.
`nstrata`	Number of PS strata to be created.
`trim_ab`	Trim external subjects who are above or below the range of current study. Default `both` trims both above and below. Other options include `above` for above only, `below` for below only, and `none` for no trimming.
`.drop_arg_fml`	internal use to drop arguments and call, this is only used in cjk.
`...`	Additional parameters for calculating the propensity score to be used in `randomForest` or `glm` .

Value

A list of class `PSRWE_DAT` with items:

data Original data with column `_ps_` for estimated PS scores and `_strata_` for PS stratum added.

ps_fml PS formula for estimated PS scores.

is_rct Whether the current study is a randomized study.

nstrata Number of strata.

Examples

```
data(ex_dta)
psrwe_est(ex_dta,
      v_covs = paste("V", 1:7, sep = ""),
      v_grp = "Group",
      cur_grp_level = "current")
```

`psrwe_infer` *Inference for the PS-Integrated Estimation*

Description

Inference for the PS-integrated approach.

Usage

```
psrwe_infer(
  dta_psrst,
  alternative = c("less", "greater", "two_sided"),
  mu = 0,
  method_pval = c("wald", "score", "score_weighted"),
  ...
)
```

Arguments

`dta_psrst`	A returned object with class `PSRWE_EST`
`alternative`	A character string for the alternative hypothesis that must be one of `"less"` (default), `"greater"`, or `"two_sided"` (for log-rank and RMST only)
`mu`	A number indicating the true value of the parameter of interest (or the difference in means for two arms), `mu = 0` when the test is log-rank or RMST
`method_pval`	A method name for p-value (default wald), no impact for Bayesian method, and `method = "score"` only is for binary outcome in single arm study (i.e., comparing with a PG set by `mu`)
`...`	Other options

Value

A list with class name `PSRWE_EST`.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
ps_rst <- psrwe_compl(ps_borrow, v_outcome = "Y_Con")
rst <- psrwe_infer(ps_rst)
rst
```

`psrwe_match` *PS matching*

Description

Match patients in external data source with patients in current study based on PS using nearest neighbor method.

Usage

```
psrwe_match(
  dta_ps,
  ratio = 3,
  strata_covs = NULL,
  caliper = 1,
  seed = NULL,
  method = c("nnwor", "optm"),
  .drop_arg_fml = FALSE,
  ...
)
```

Arguments

`dta_ps`	A list of class `PSRWE_DAT` that is generated using the `psrwe_est` function.
`ratio`	Matching ratio (RWD : Current) with default value 3 meaning 3:1 matching.
`strata_covs`	Stratification covariates for matching.
`caliper`	PS matching caliper width. Default 1. This specifies a width (euclidean distance) on the probability scale.
`seed`	Random seed.
`method`	matching algorithm for PS matching.
`.drop_arg_fml`	internal use to drop arguments and call, this is only used in cjk.
`...`	Additional parameters for matching

Value

A list of class `PSRWE_DTA_MAT` with items:

data Original data with column `_ps_` for estimated PS scores, `match_id` for matched current study subject ID, and `_strata_` for PS stratum added.

ps_fml PS formula for estimated PS scores.

nstrata Number of strata.

ratio Matching ratio.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
                     v_covs = paste("V", 1:7, sep = ""),
                     v_grp = "Group",
                     cur_grp_level = "current")

## With neighest neighbor
.remark_nnmatch <- function() {
  dta_ps_mat <- psrwe_match(dta_ps, ratio = 2, strata_covs = "V1",
                            seed = 123)
  dta_ps_mat
}
## Unmark below to run
# .remark_nnmatch()

## With optmatch
.remark_optmatch <- function() {
    warning("The optmatch package may restrict use (academic license).")
    dta_ps_opt <- psrwe_match(dta_ps, ratio = 2,
                               strata_covs = "V1",
                               method = "optm", seed = 123)
    dta_ps_opt
}
## Unmark below to run if optmatch is available.
# .remark_optmatch()
```

`psrwe_outana` *Outcome Analysis for PS-Integrated Estimation*

Description

Report outcome analysis for the PS-integrated approach.

Usage

```
psrwe_outana(
  dta_psrst,
  method_ci = c("wald", "wilson"),
  conf_type = c("log_log", "plain"),
  conf_int = 0.95,
  alternative = c("less", "greater", "two_sided"),
  mu = 0,
  method_pval = c("wald", "score", "score_weighted"),
  ...
)
```

Arguments

`dta_psrst`	A returned object with class `PSRWE_EST`
`method_ci`	A method name for confidence interval (default wald)
`conf_type`	A type name of transformation for the confidence interval of PSKM approach (default log_log)
`conf_int`	A two-sided level of confidence/credible limits (default 0.95)
`alternative`	A character string for the alternative hypothesis that must be one of `"less"` (default) or `"greater"`, or `"two_sided"` (for log-rank and RMST only)
`mu`	A number indicating the true value of the parameter of interest (or the difference in means for two arms), `mu = 0` when the test is log-rank or RMST
`method_pval`	A method name for p-value (default wald), no impact for Bayesian method, and `method = "score"` only is for binary outcome in single arm study (i.e., comparing with a PG set by `mu`)
`...`	Other options

Details

This function is mainly for summarizing and reporting the outcome analysis for the PS-integrated estimation. The input `dta_psrst` can be generated from the functions `psrwe_powerp`, `psrwe_compl`, and `psrwe_survkm`. See the functions `psrwe_ci` and `psrwe_infer` for the options of outcome analyses.

Value

A list with class name `PSRWE_EST_OUTANA`.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
ps_rst <- psrwe_compl(ps_borrow, v_outcome = "Y_Con")
rst <- psrwe_outana(ps_rst)
rst
```

`psrwe_powerp` *Get posterior samples based on PS-power prior approach*

Description

Draw posterior samples of the parameters of interest for the PS-power prior approach

Usage

```
psrwe_powerp(
  dta_psbor,
  v_outcome = "Y",
  outcome_type = c("continuous", "binary"),
  prior_type = c("fixed", "random"),
  ...,
  seed = NULL
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`v_outcome`	Column name corresponding to the outcome.
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`prior_type`	Whether treat power parameter as fixed (`fixed`) or fully Bayesian (`random`).
`...`	extra parameters for calling function `rwe_stan`.
`seed`	Random seed.

Value

A class `PSRWE_RST` list with the following objects

Observed Observed mean and SD of the outcome by group, arm and stratum

Control A list of estimated mean and SD of the outcome by stratum in the control arm

Treatment A list of estimated mean and SD of the outcome by stratum in the treatment arm for RCT

Effect A list of estimated mean and SD of the treatment effect by stratum for RCT

Borrow Borrowing information from `dta_psbor`

stan_rst Result from STAN sampling

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
rst <- psrwe_powerp(ps_borrow, v_outcome = "Y_Con", seed = 123)
```

`psrwe_powerp_watt` *Get posterior samples based on PS-power prior approach (WATT)*

Description

Draw posterior samples of the parameters of interest for the PS-power prior approach with weights of ATT (WATT)

Usage

```
psrwe_powerp_watt(
  dta_psbor,
  v_outcome = "Y",
  outcome_type = c("continuous", "binary"),
  prior_type = c("fixed", "random"),
  ...,
  seed = NULL
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`v_outcome`	Column name corresponding to the outcome.
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`prior_type`	Whether treat power parameter as fixed (`fixed`) or fully Bayesian (`random`).
`...`	extra parameters for calling function `rwe_stan`.
`seed`	Random seed.

Value

A class `PSRWE_RST` list with the following objects

Observed Observed mean and SD of the outcome by group, arm and stratum

Control A list of estimated mean and SD of the outcome by stratum in the control arm

Treatment A list of estimated mean and SD of the outcome by stratum in the treatment arm for RCT

Effect A list of estimated mean and SD of the treatment effect by stratum for RCT

Borrow Borrowing information from `dta_psbor`

stan_rst Result from STAN sampling

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current",
       nstrata = 1)
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
rst <- psrwe_powerp_watt(ps_borrow, v_outcome = "Y_Bin", seed = 123)
```

`psrwe_survkm` *PS-Integrated Kaplan-Meier Estimation*

Description

Estimate the mean of a survival outcome at a given time point based on PS-integrated Kaplan-Meier approach. Variance can be estimated by Jackknife methods. Apply to the case when there is only one external data source.

Usage

```
psrwe_survkm(
  dta_psbor,
  pred_tp,
  v_time = "time",
  v_event = "event",
  stderr_method = c("naive", "jk", "sjk", "cjk", "sbs", "cbs", "none"),
  n_bootstrap = 200,
  ...
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`pred_tp`	A numeric value corresponding to time of interest (e.g., 365 days or 1 year)
`v_time`	Column name corresponding to event time
`v_event`	Column name corresponding to event status
`stderr_method`	Method for computing StdErr, see Details
`n_bootstrap`	Number of bootstrap samples (for bootstrap stderr)
`...`	Additional Parameters

Details

`stderr_method` includes `naive` as default which mostly follows Greenwood formula, `jk` using Jackknife method within each stratum, `sjk` using simple Jackknife method for combined estimates such as point estimates in single arm or treatment effects in RCT, or `cjk` for complex Jackknife method including refitting PS model, matching, trimming, calculating borrowing parameters, and combining overall estimates. Note that `sjk` may take a while longer to finish and `cjk` will take even much longer to finish. The `sbs` and `cbs` is for simple and complex Bootstrap methods.

Value

A data frame with class name `PSRWE_RST`. It contains the composite estimation of the mean for each stratum as well as the Jackknife estimation. The results can be further summarized by its S3 method `summary`. The results can be also analyzed by `psrwe_outana` for outcome analysis and inference.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
       v_covs = paste("V", 1:7, sep = ""),
       v_grp = "Group",
       cur_grp_level = "current")
ps_borrow <- psrwe_borrow(total_borrow = 30, dta_ps)
rst       <- psrwe_survkm(ps_borrow,
                          pred_tp = 365,
                          v_time = "Y_Surv",
                          v_event = "Status")
rst
```

`psrwe_survlrk` *PS-Integrated Log-Rank Test For Comparing Time-to-event Outcomes*

Description

Log-rank test evaluates two-arm RCT for up to a given time point. Variance can be estimated by Jackknife methods. Apply to the case when there is only one external data source and two-arm RCT.

Usage

```
psrwe_survlrk(
  dta_psbor,
  pred_tp,
  v_time = "time",
  v_event = "event",
  stderr_method = c("naive", "jk", "sjk", "cjk", "sbs", "cbs", "none"),
  n_bootstrap = 200,
  ...
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`pred_tp`	A numeric value corresponding to time of interest (e.g., 365 days or 1 year)
`v_time`	Column name corresponding to event time
`v_event`	Column name corresponding to event status
`stderr_method`	Method for computing StdErr (see Details)
`n_bootstrap`	Number of bootstrap samples (for bootstrap stderr)
`...`	Additional Parameters

Details

`stderr_method` includes `naive` as default which mostly follows Greenwood formula, `jk` using Jackknife method within each stratum, `sjk` using simple Jackknife method for combined estimates such as point estimates in single arm or treatment effects in RCT, or `cjk` for complex Jackknife method including refitting PS model, matching, trimming, calculating borrowing parameters, and combining overall estimates. Note that `sjk` may take a while longer to finish and `cjk` will take even much longer to finish. The `sbs` and `cbs` is for simple and complex Bootstrap methods.

Value

A data frame with class name `PSRWE_RST_TESTANA`. It contains the test statistics of each stratum as well as the Jackknife estimation. The results can be further summarized by its S3 method `summary`. The results can be also analyzed by `psrwe_outana` for outcome analysis and inference.

Examples

```
data(ex_dta_rct)
dta_ps_rct <- psrwe_est(ex_dta_rct,
                        v_covs = paste("V", 1:7, sep = ""),
                        v_grp = "Group", cur_grp_level = "current",
                        v_arm = "Arm", ctl_arm_level = "control",
                        ps_method = "logistic", nstrata = 5,
                        stra_ctl_only = FALSE)
ps_bor_rct <- psrwe_borrow(dta_ps_rct, total_borrow = 30)
rst_lrk <- psrwe_survlrk(ps_bor_rct,
                         pred_tp = 365,
                         v_time = "Y_Surv",
                         v_event = "Status")
rst_lrk
```

`psrwe_survrmst` *PS-Integrated Restricted Mean Survival Time (RMST) Test For Comparing Time-to-event Outcomes*

Description

RMST test evaluates two-arm RCT for up to a given time point. Variance can be estimated by Jackknife methods. Apply to the case when there is only one external data source and two-arm RCT.

Usage

```
psrwe_survrmst(
  dta_psbor,
  pred_tp,
  v_time = "time",
  v_event = "event",
  stderr_method = c("naive", "jk", "sjk", "cjk", "sbs", "cbs", "none"),
  n_bootstrap = 200,
  ...
)
```

Arguments

`dta_psbor`	A class `PSRWE_BOR` object generated by `psrwe_borrow`.
`pred_tp`	A numeric value corresponding to time of interest (e.g., 365 days or 1 year)
`v_time`	Column name corresponding to event time
`v_event`	Column name corresponding to event status
`stderr_method`	Method for computing StdErr, see Details
`n_bootstrap`	Number of bootstrap samples (for bootstrap stderr)
`...`	Additional Parameters

Details

`stderr_method` includes `naive` as default which mostly follows Greenwood formula, `jk` using Jackknife method within each stratum, `sjk` using simple Jackknife method for combined estimates such as point estimates in single arm or treatment effects in RCT, or `cjk` for complex Jackknife method including refitting PS model, matching, trimming, calculating borrowing parameters, and combining overall estimates. Note that `sjk` may take a while longer to finish and `cjk` will take even much longer to finish. The `sbs` and `cbs` is for simple and complex Bootstrap methods.

Value

A data frame with class name `PSRWE_RST_TESTANA`. It contains the test statistics of each stratum as well as the Jackknife estimation. The results can be further summarized by its S3 method `summary`. The results can be also analyzed by `psrwe_outana` for outcome analysis and inference.

Examples

```
data(ex_dta_rct)
dta_ps_rct <- psrwe_est(ex_dta_rct,
                        v_covs = paste("V", 1:7, sep = ""),
                        v_grp = "Group", cur_grp_level = "current",
                        v_arm = "Arm", ctl_arm_level = "control",
                        ps_method = "logistic", nstrata = 5,
                        stra_ctl_only = FALSE)
ps_bor_rct <- psrwe_borrow(dta_ps_rct, total_borrow = 30)
rst_rmst <- psrwe_survrmst(ps_bor_rct,
                           pred_tp = 365,
                           v_time = "Y_Surv",
                           v_event = "Status")
rst_rmst
```

`rwe_cl` *Composite Likelihood Estimation*

Description

Estimate parameter of interest based composite likelihood for a single PS stratum

Usage

```
rwe_cl(
  dta_cur,
  dta_ext,
  n_borrow = 0,
  outcome_type = c("continuous", "binary"),
  equal_sd = TRUE
)
```

Arguments

`dta_cur`	Vector of outcome from a PS stratum in current study
`dta_ext`	Vector of outcome from a PS stratum in external data source
`n_borrow`	Number of subjects to be borrowed
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`equal_sd`	Boolean. whether sd is the same between the current study and external data source

Value

Maximum composite likelihood estimator of the mean

Examples

```
x <- rnorm(100,  mean = 0, sd = 1)
y <- rnorm(1000, mean = 1, sd = 2)
rwe_cl(x, y, n_borrow = 20, equal_sd = FALSE)
```

`rwe_cl_watt` *Composite Likelihood Estimation (WATT)*

Description

Estimate parameter of interest based composite likelihood for a single PS stratum with weights of ATT (WATT).

Usage

```
rwe_cl_watt(
  dta_cur,
  dta_ext,
  n_borrow = 0,
  dta_ext_watt_di = NULL,
  outcome_type = c("continuous", "binary"),
  equal_sd = TRUE
)
```

Arguments

`dta_cur`	Vector of outcome from a PS stratum in current study
`dta_ext`	Vector of outcome from a PS stratum in external data source
`n_borrow`	Number of subjects to be borrowed
`dta_ext_watt_di`	Weights of ATT for subjects in external data source
`outcome_type`	Type of outcomes: `continuous` or `binary`.
`equal_sd`	Boolean. whether sd is the same between the current study and external data source

Value

Maximum composite likelihood estimator of the mean

Examples

```
x <- rnorm(100,  mean = 0, sd = 1)
y <- rnorm(1000, mean = 1, sd = 2)
rwe_cl_watt(x, y, n_borrow = 20, equal_sd = FALSE)
```

`rwe_cut` *Create strata*

Description

Cut a sequence of numbers into bins.

The cut points are chosen such that there will with equal numbers in each bin for `x`. By default, values of `y` that are outside the range of `x` will be excluded from the bins, unless they are in the `keep_inx`.

Usage

```
rwe_cut(
  x,
  y = x,
  breaks = 5,
  keep_inx = NULL,
  trim_ab = c("both", "above", "below", "none")
)
```

Arguments

`x`	Vector of values based on which cut points will be determined
`y`	Vector of values to be cut, default to be the same as `x`
`breaks`	Number of cut points
`keep_inx`	Indices of `y` that will be categorized as 1 or the largest bin even if their values are out of range of `x`, i.e. the `y`'s that will not be trimmed
`trim_ab`	Trim external subjects who are above or below the range of current study. Default `both` trims both above and below. Other options include `above` for above only, `below` for below only, and `none` for no trimming.

Value

A vector of stratum assignment for `y`. The `y`'s that are outside the range of `x` and not in `keep_inx` are assigned `NA` in the result.

Examples

```
x <- rnorm(100,  mean = 0, sd = 1)
y <- rnorm(1000, mean = 1, sd = 2)
rwe_cut(x, y, breaks = 5)
```

`rwe_km` *Kaplan-Meier Estimation*

Description

Estimate survival probability based on Kaplan-Meier estimator for a single PS stratum

Usage

```
rwe_km(
  dta_cur,
  dta_ext = NULL,
  n_borrow = 0,
  pred_tps = NULL,
  stderr_method = "naive"
)
```

Arguments

`dta_cur`	Matrix of time and event from a PS stratum in current study
`dta_ext`	Matrix of time and event from a PS stratum in external data source
`n_borrow`	Number of subjects to be borrowed
`pred_tps`	Time points to be estimated (unique and sorted)
`stderr_method`	Method for computing StdErr (available for naive only)

Value

Estimation of survival probabilities at time `pred_tps`

`rwe_lrk` *Log-rank Estimation*

Description

Estimate log-rank estimates for a single PS stratum

Usage

```
rwe_lrk(
  dta_cur,
  dta_ext,
  dta_cur_trt,
  n_borrow = 0,
  pred_tps = NULL,
  stderr_method = "naive"
)
```

Arguments

`dta_cur`	Matrix of time and event from a PS stratum in current study (control arm only)
`dta_ext`	Matrix of time and event from a PS stratum in external data source (control arm only)
`dta_cur_trt`	Matrix of time and event from a PS stratum in current study (treatment arm only)
`n_borrow`	Number of subjects to be borrowed
`pred_tps`	All time points of events (unique and sorted)
`stderr_method`	Method for computing StdErr (available for naive only)

Value

Estimation of log-rank estimates at time `pred_tps`

`rwe_rmst` *RMST Estimation*

Description

Estimate RMST estimates for a single PS stratum

Usage

```
rwe_rmst(
  dta_cur,
  dta_ext,
  dta_cur_trt,
  n_borrow = 0,
  pred_tps = NULL,
  stderr_method = "naive"
)
```

Arguments

`dta_cur`	Matrix of time and event from a PS stratum in current study (control arm only)
`dta_ext`	Matrix of time and event from a PS stratum in external data source (control arm only)
`dta_cur_trt`	Matrix of time and event from a PS stratum in current study (treatment arm only)
`n_borrow`	Number of subjects to be borrowed
`pred_tps`	All time points of events (unique and sorted)
`stderr_method`	Method for computing StdErr (available for naive only)

Value

Estimation of RMST estimates at time `pred_tps`

`rwe_stan` *Call STAN models*

Description

Call STAN models. Called by `psrwe_powerp`.

Usage

```
rwe_stan(
  lst_data,
  stan_mdl = c("powerps", "powerpsbinary", "powerp"),
  chains = 4,
  iter = 2000,
  warmup = 1000,
  control = list(adapt_delta = 0.95),
  ...
)
```

Arguments

`lst_data`	List of study data to be passed to STAN
`stan_mdl`	STAN model including **powerps** PS-power prior model for continuous outcomes **powerpsbinary** PS-power prior model for binary outcomes **powerp** Power prior model
`chains`	STAN parameter. Number of Markov chainsm
`iter`	STAN parameter. Number of iterations
`warmup`	STAN parameter. Number of burnin.
`control`	STAN parameter. See `rstan::stan` for details.
`...`	other options to call STAN sampling such as `thin`, `algorithm`. See `rstan::sampling` for details.#'

Value

Result from STAN sampling

`summary.PSRWE_DTA` *Summarize PS estimation and stratification results*

Description

Get number of subjects and the distances of PS distributions for each PS stratum.

Usage

```
## S3 method for class 'PSRWE_DTA'
summary(
  object,
  metric = c("ovl", "ksd", "std", "abd", "ley", "mhb", "omkss"),
  min_n0 = 10,
  ...
)
```

Arguments

`object`	A list of class `PSRWE_DAT` that is generated using the `psrwe_est` function.
`metric`	Metric to use for calculating the distance with options: `ovl` Overlapping area(default) `ksd` Kullback-Leibler distance `astd` Standardized absolute mean difference `std` Standardized mean difference `abd` Absolute difference in means `ley` Levy distance `mhb` Mahalanobis distance `omkss` One minus Kolmogorov-Smirnov statistic
`min_n0`	threshold for number of external subjects, below which the external data in the current stratum will be ignored by setting the PS distance to 0. Default value 10.
`...`	Additional parameters.

Value

A list with columns:

Summary A data frame with Stratum, number of subjects in RWD, current study, number of subjects in control and treatment arms for RCT studies, and distance in PS distributions.

Overall A data frame with overall number of not-trimmed subjects in RWD, number of patients in current study, number of subjects in control and treatment arms for RCT studies, and distance in PS distributions.

N Vector of total number of total RWD patients, number of trimmed RWD patients, and total number of current study patients.

ps_fml PS model.

Distance_metric Metric used for calculating the distance.

Examples

```
data(ex_dta)
dta_ps <- psrwe_est(ex_dta,
                    v_covs = paste("V", 1:7, sep = ""),
                    v_grp = "Group",
                    cur_grp_level = "current")
dta_ps

## With different similarity metric
print(dta_ps, metric = "omkss")
dta_ps_sum <- summary(dta_ps, metric = "omkss")
```

`summary.PSRWE_DTA_MAT` *Summarize PS estimation and matching results*

Description

Get number of subjects for each PS stratum.

Usage

```
## S3 method for class 'PSRWE_DTA_MAT'
summary(object, ...)
```

Arguments

`object`	A list of class `PSRWE_DTA_MAT` that is generated using the `psrwe_match` function.
`...`	Additional parameters.

Value

A list with columns:

- **Summary** A data frame with Stratum (defined by covariates), number of subjects in RWD, current study, number of subjects in control and treatment arms for RCT studies.
- **Overall** A data frame with overall number of not-trimmed subjects in RWD, number of patients in current study, number of subjects in control and treatment arms for RCT studies.
- **N** Vector of total number of total RWD patients, number of trimmed RWD patients, total number of current study patients, number of current control patients with less than `ratio` matched RWD subjects.
- **ps_fml** PS model.
- **N_Match** Number of current control subjects matched with `ratio`, 0 and other number of RWD subjects.
- **ratio** Matching ratio.

`summary.PSRWE_RST` *Summarize overall estimation results*

Description

S3 method summarizing overall estimation results

Usage

```
## S3 method for class 'PSRWE_RST'
summary(object, ...)
```

Arguments

`object`	A list of class `PSRWE_RST` that is generated using the `psrwe_powerp`, `psrwe_compl`, or `psrwe_survkm` function.
`...`	Additional parameters.

Value

A list with data frames for the borrowing and estimation results.

`summary.PSRWE_RST_OUTANA` *Summary outcome analysis results*

Description

Summary information of outcome analysis results

Usage

```
## S3 method for class 'PSRWE_RST_OUTANA'
summary(object, pred_tps = NULL, ...)
```

Arguments

`object`	A list of class `PSRWE_RST_OUTANA` that is generated using the `psrwe_outana` function.
`pred_tps`	Specified time points
`...`	Additional parameters

Index