# Package 'riskdiff'

May 9, 2026

**Title** Risk Difference Estimation with Multiple Link Functions and Inverse Probability of Treatment Weighting

**Version** 0.3.0

**Date** 2026-02-26

**Description** Calculates risk differences (or prevalence differences for cross-sectional data) and Number Needed to Treat (NNT) using generalized linear models with automatic link function selection. Provides robust model fitting with fallback methods, support for stratification and adjustment variables, inverse probability of treatment weighting (IPTW) for causal inference with NNT calculations, and publication-ready output formatting. Handles model convergence issues gracefully and provides confidence intervals using multiple approaches.
Methods are based on approaches described in Mark W. Donoghoe and Ian C. Marschner (2018) ``logbin: An R Package for Relative Risk Regression Using the Log-Binomial Model'' <doi:10.18637/jss.v086.i09> for robust GLM fitting, Peter C. Austin (2011) ``An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies'' <doi:10.1080/00273171.2011.568786> for IPTW methods, and standard epidemiological methods for risk difference estimation as described in Kenneth J. Rothman, Sander Greenland and Timothy L. Lash (2008, ISBN:9780781755641) ``Modern Epidemiology''.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Depends** R (>= 3.5.0)

**Imports** dplyr (>= 1.0.0), purrr, tibble, rlang, scales, stringr, stats, ggplot2

**Suggests** kableExtra, testthat (>= 3.0.0), knitr, rmarkdown, covr, mockery

**Config/testthat/edition** 3

**URL** https://github.com/jackmurphy2351/riskdiff

**BugReports** https://github.com/jackmurphy2351/riskdiff/issues

**LazyData** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** John D. Murphy [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7714-9976>, MPH, PhD)

**Maintainer** John D. Murphy <jackdmurphy@protonmail.com>

**Repository** CRAN

**Date/Publication** 2026-02-26 21:50:02 UTC

# Contents



---

`cachar_sample` *Synthetic Cancer Risk Factor Study Data*

---

## Description

A synthetic dataset inspired by cancer screening and risk factor patterns observed during an opportunistic screening program conducted at the Cachar Cancer Hospital and Research Centre in Northeast India, specifically designed to reflect authentic epidemiological relationships without using real patient data.

## Usage

```
cachar_sample
```

## Format

A data frame with 2,500 rows and 12 variables:

**id** Participant identifier (1 to 2500)

**age** Age in years (continuous, range 18-84)

**sex** Biological sex: "male" or "female"

**residence** Residence type: "rural", "urban", or "urban slum"

**smoking** Current smoking status: "No" or "Yes"

**tobacco_chewing** Current tobacco chewing: "No" or "Yes"

**areca_nut** Current areca nut use: "No" or "Yes"

**alcohol** Current alcohol use: "No" or "Yes"

**abnormal_screen** Binary outcome: 1 = abnormal screening (precancerous lesions or cancer), 0 = normal

**head_neck_abnormal** Binary outcome: 1 = head/neck abnormality detected, 0 = normal

**age_group** Age categories: "Under 40", "40-60", "Over 60"

**tobacco_areca_both** Combined exposure: "Yes" if both tobacco_chewing and areca_nut are "Yes", "No" otherwise

## Details

This synthetic dataset was designed to reflect authentic epidemiological patterns observed in Northeast India, particularly the distinctive tobacco and areca nut use patterns of the region. All data points are mathematically generated rather than collected from real individuals.

**Key epidemiological features modeled:**

- **Areca nut use**: Very high prevalence (~69%) reflecting regional cultural practices
- **Tobacco chewing**: Moderate to high prevalence (~53%), often used with areca nut
- **Smoking**: Lower prevalence (~13%) with strong male predominance
- **Cancer outcomes**: Realistic prevalence (~3.5%) for population-based screening, including both precancerous lesions and invasive cancers
- **Geographic patterns**: Predominantly rural population (~87%)

**Synthetic Data Advantages:** The synthetic approach preserves authentic statistical relationships while:

- Avoiding any privacy or ethical concerns
- Ensuring reproducible examples and tests
- Providing controlled demonstration scenarios
- Maintaining cultural authenticity for educational purposes

**Risk Factor Relationships:** The data models realistic dose-response relationships between multiple tobacco exposures and cancer outcomes, with particularly strong associations for areca nut use and head/neck abnormalities, reflecting authentic epidemiological patterns from this region.

## Note

This synthetic dataset is designed for educational and software demonstration purposes. While the statistical relationships reflect authentic epidemiological patterns, the data should not be used for research conclusions about real populations. The cultural patterns represented (high areca nut use, specific tobacco consumption practices) are authentic to Northeast India.

## Source

Synthetic dataset created for the riskdiff package. Inspired by cancer screening patterns observed in Northeast India but contains no real patient data. Statistical relationships designed to reflect authentic epidemiological patterns from this region for educational and methodological purposes.

## References

Epidemiological patterns modeled after studies of tobacco use and cancer risk in Northeast India. For research involving actual populations from this region, consult published literature on areca nut and tobacco-related cancer risks in South Asian populations.

Warnakulasuriya S, Trivedy C, Peters TJ (2002). "Areca nut use: an independent risk factor for oral cancer." BMJ, 324(7341), 799-800.

Gupta PC, Ray CS (2004). "Epidemiology of betel quid use." Annals of the Academy of Medicine, Singapore, 33(4 Suppl), 31-36.

## Examples

```
data(cachar_sample)
head(cachar_sample)

# Basic descriptive statistics
table(cachar_sample$areca_nut, cachar_sample$abnormal_screen)

# Regional tobacco use patterns
with(cachar_sample, table(areca_nut, tobacco_chewing))

# Simple risk difference for areca nut and abnormal screening
rd_areca <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut"
)
print(rd_areca)

# Age-adjusted analysis
rd_adjusted <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut",
  adjust_vars = "age"
)
print(rd_adjusted)
```

```
# Stratified by sex
rd_stratified <- calc_risk_diff(
  data = cachar_sample,
  outcome = "head_neck_abnormal",
  exposure = "smoking",
  strata = "sex"
)
print(rd_stratified)

# Multiple tobacco exposures comparison
rd_smoking <- calc_risk_diff(cachar_sample, "abnormal_screen", "smoking")
rd_chewing <- calc_risk_diff(cachar_sample, "abnormal_screen", "tobacco_chewing")
rd_areca <- calc_risk_diff(cachar_sample, "abnormal_screen", "areca_nut")

# Compare risk differences
cat("Risk differences for abnormal screening:\n")
cat("Smoking:", sprintf("%.1f%%", rd_smoking$rd * 100), "\n")
cat("Tobacco chewing:", sprintf("%.1f%%", rd_chewing$rd * 100), "\n")
cat("Areca nut:", sprintf("%.1f%%", rd_areca$rd * 100), "\n")

# Create summary table
cat(create_simple_table(rd_areca, "Abnormal Screening Risk by Areca Nut Use"))
```

---

`calc_iptw_weights` *Calculate Propensity Scores and IPTW Weights*

---

## Description

Calculates propensity scores and inverse probability of treatment weights for use in standardized risk difference estimation. Implements multiple approaches for weight calculation and includes diagnostic tools.

## Usage

```
calc_iptw_weights(
  data,
  treatment,
  covariates,
  method = "logistic",
  weight_type = "ATE",
  stabilize = TRUE,
  trim_weights = TRUE,
  trim_quantiles = c(0.01, 0.99),
  verbose = FALSE
)
```

## Arguments

| | |
|---|---|
| `data` | A data frame containing treatment and covariate data |
| `treatment` | Character string naming the binary treatment variable |
| `covariates` | Character vector of covariate names for propensity score model |
| `method` | Method for propensity score estimation: "logistic" (default), "probit", or "cloglog" |
| `weight_type` | Type of weights to calculate: "ATE" (average treatment effect, default), "ATT" (average treatment effect on treated), "ATC" (average treatment effect on controls) |
| `stabilize` | Logical indicating whether to use stabilized weights (default: TRUE) |
| `trim_weights` | Logical indicating whether to trim extreme weights (default: TRUE) |
| `trim_quantiles` | Vector of length 2 specifying quantiles for weight trimming (default: c(0.01, 0.99)) |
| `verbose` | Logical indicating whether to print diagnostic information (default: FALSE) |

## Details

**Propensity Score Estimation:**

The function fits a model predicting treatment assignment from covariates:

- **Logistic regression**: Standard approach, assumes logit link
- **Probit regression**: Uses probit link, may be more robust with extreme probabilities
- **Complementary log-log**: Useful when treatment is rare

**Weight Types:**

- **ATE weights**: 1/pi(X) for treated, 1/(1-pi(X)) for controls
- **ATT weights**: 1 for treated, pi(X)/(1-pi(X)) for controls
- **ATC weights**: (1-pi(X))/pi(X) for treated, 1 for controls

Where pi(X) is the propensity score (probability of treatment given X).

**Stabilized Weights:**

When stabilize=TRUE, weights are multiplied by marginal treatment probabilities to reduce variance while maintaining unbiasedness (Robins et al., 2000).

**Weight Trimming:**

Extreme weights can cause instability. Trimming replaces weights outside specified quantiles with the quantile values (Crump et al., 2009).

## Value

A list containing:

**data** Original data with added propensity scores and weights

**ps_model** Fitted propensity score model

**weights** Vector of calculated weights

**ps** Vector of propensity scores

**diagnostics** List of diagnostic information including balance statistics

**method** Method used for propensity score estimation

**weight_type** Type of weights calculated

## References

Austin PC (2011). "An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies." Multivariate Behavioral Research, 46(3), 399-424. doi:10.1080/00273171.2011.568786

Crump RK, Hotz VJ, Imbens GW, Mitnik OA (2009). "Dealing with Limited Overlap in Estimation of Average Treatment Effects." Biometrika, 96(1), 187-199.

Hernan MA, Robins JM (2020). Causal Inference: What If. Boca Raton: Chapman & Hall/CRC.

Robins JM, Hernan MA, Brumback B (2000). "Marginal Structural Models and Causal Inference in Epidemiology." Epidemiology, 11(5), 550-560.

## Examples

```
data(cachar_sample)

# Calculate ATE weights for areca nut use
iptw_result <- calc_iptw_weights(
  data = cachar_sample,
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking"),
  weight_type = "ATE"
)

# Check balance
print(iptw_result$diagnostics$balance_table)

# Calculate ATT weights (effect on the treated)
iptw_att <- calc_iptw_weights(
  data = cachar_sample,
  treatment = "tobacco_chewing",
  covariates = c("age", "sex", "residence", "areca_nut"),
  weight_type = "ATT"
)
```

---

`calc_risk_diff` *Calculate Risk Differences with Robust Model Fitting and Boundary Detection*

---

## Description

Calculates risk differences (or prevalence differences for cross-sectional data) using generalized linear models with identity, log, or logit links. Version 0.2.1 includes enhanced boundary detection, robust confidence intervals, and improved data quality validation to prevent extreme confidence intervals in stratified analyses.

The function addresses common convergence issues with identity link binomial GLMs by implementing a fallback strategy across multiple link functions, similar to approaches described in Donoghoe & Marschner (2018) for relative risk regression.

## Usage

```
calc_risk_diff(
  data,
  outcome,
  exposure,
  nnt = FALSE,
  adjust_vars = NULL,
  strata = NULL,
  link = "auto",
  alpha = 0.05,
  boundary_method = "auto",
  verbose = FALSE
)
```

## Arguments

| | |
|---|---|
| data | A data frame containing all necessary variables |
| outcome | Character string naming the binary outcome variable (must be 0/1 or logical) |
| exposure | Character string naming the exposure variable of interest |
| nnt | Logical indicating whether to return Number Needed to Treat instead of risk difference (default: FALSE) |
| adjust_vars | Character vector of variables to adjust for (default: NULL) |
| strata | Character vector of stratification variables (default: NULL) |
| link | Character string specifying link function: "auto", "identity", "log", or "logit" (default: "auto") |
| alpha | Significance level for confidence intervals (default: 0.05) |
| boundary_method | Method for handling boundary cases: "auto", "profile", "bootstrap", "wald" (default: "auto") |
| verbose | Logical indicating whether to print diagnostic messages (default: FALSE) |

## Details

**New in Version 0.2.2: NNT Calculation capability:**

When `nnt = TRUE`, the function returns Number Needed to Treat (NNT) instead of risk differences. NNT represents the number of individuals that need to be treated to prevent one additional adverse outcome. NNT is calculated as 1/|RD| and confidence intervals are transformed using the delta method. NNT is undefined when RD = 0 and is reported as Inf when |RD| < 0.001. For harmful exposures (RD > 0), this represents Number Needed to Harm (NNH).

## Value

A tibble of class "riskdiff_result" containing the following columns:

**exposure_var** Character. Name of exposure variable analyzed

**rd** Numeric. Risk difference estimate OR Number Needed to Treat if nnt=TRUE (see Details)

**ci_lower** Numeric. Lower bound of confidence interval (RD scale or NNT scale)

**ci_upper** Numeric. Upper bound of confidence interval (RD scale or NNT scale)

**p_value** Numeric. P-value for test of null hypothesis (risk difference = 0)

**model_type** Character. Link function successfully used ("identity", "log", "logit", or error type)

**n_obs** Integer. Number of observations used in analysis

**on_boundary** Logical. TRUE if MLE is on parameter space boundary

**boundary_type** Character. Type of boundary: "none", "upper_bound", "lower_bound", "separation", "both_bounds"

**boundary_warning** Character. Warning message for boundary cases (if any)

**ci_method** Character. Method used for confidence intervals ("wald", "profile", "bootstrap")

**...** Additional columns for stratification variables if specified

The returned object has attributes including the original function call and alpha level used. Risk differences are on the probability scale where 0.05 represents a 5 percentage point difference.

## References

Donoghoe MW, Marschner IC (2018). "logbin: An R Package for Relative Risk Regression Using the Log-Binomial Model." Journal of Statistical Software, 86(9), 1-22. doi:10.18637/jss.v086.i09

Marschner IC, Gillett AC (2012). "Relative Risk Regression: Reliable and Flexible Methods for Log-Binomial Models." Biostatistics, 13(1), 179-192.

Venzon DJ, Moolgavkar SH (1988). "A Method for Computing Profile-Likelihood-Based Confidence Intervals." Journal of the Royal Statistical Society, 37(1), 87-94.

Rothman KJ, Greenland S, Lash TL (2008). Modern Epidemiology, 3rd edition. Lippincott Williams & Wilkins.

## Examples

```
# Simple risk difference
data(cachar_sample)
rd_simple <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut"
)
print(rd_simple)

# Age-adjusted risk difference
rd_adjusted <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut",
  adjust_vars = "age"
)
print(rd_adjusted)

# Stratified analysis with enhanced error checking and boundary detection
```

```
rd_stratified <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut",
  strata = "residence",
  verbose = TRUE  # See diagnostic messages and boundary detection
)
print(rd_stratified)

# Check for boundary cases
if (any(rd_stratified$on_boundary)) {
  cat("Boundary cases detected!\n")
  boundary_rows <- which(rd_stratified$on_boundary)
  for (i in boundary_rows) {
    cat("Row", i, ":", rd_stratified$boundary_type[i], "\n")
  }
}

# Force profile likelihood CIs for enhanced robustness
rd_profile <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "areca_nut",
  boundary_method = "profile"
)

# Calculate Number Needed to Treat instead of risk difference
data(cachar_sample)
nnt_result <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "smoking",
  nnt = TRUE
)
print(nnt_result)

# NNT with adjustment variables
nnt_adjusted <- calc_risk_diff(
  data = cachar_sample,
  outcome = "abnormal_screen",
  exposure = "smoking",
  adjust_vars = "age",
  nnt = TRUE
)
```

---

`calc_risk_diff_iptw` *Calculate Standardized Risk Differences Using IPTW*

---

## Description

Calculates standardized risk differences using inverse probability of treatment weighting. This approach estimates causal effects under the assumption of no unmeasured confounding by creating a pseudo-population where treatment assignment is independent of measured confounders.

## Usage

```
calc_risk_diff_iptw(
  data,
  outcome,
  treatment,
  covariates,
  nnt = FALSE,
  iptw_weights = NULL,
  weight_type = "ATE",
  ps_method = "logistic",
  stabilize = TRUE,
  trim_weights = TRUE,
  alpha = 0.05,
  bootstrap_ci = FALSE,
  boot_n = 1000,
  verbose = FALSE
)
```

## Arguments

| | |
|---|---|
| data | A data frame containing outcome, treatment, and covariate data |
| outcome | Character string naming the binary outcome variable |
| treatment | Character string naming the binary treatment variable |
| covariates | Character vector of covariate names for propensity score model |
| nnt | Logical indicating whether to return Number Needed to Treat instead of risk difference (default: FALSE) |
| iptw_weights | Optional vector of pre-calculated IPTW weights |
| weight_type | Type of weights if calculating: "ATE", "ATT", or "ATC" (default: "ATE") |
| ps_method | Method for propensity score estimation (default: "logistic") |
| stabilize | Whether to use stabilized weights (default: TRUE) |
| trim_weights | Whether to trim extreme weights (default: TRUE) |
| alpha | Significance level for confidence intervals (default: 0.05) |
| bootstrap_ci | Whether to use bootstrap confidence intervals (default: FALSE) |
| boot_n | Number of bootstrap replicates if bootstrap_ci=TRUE (default: 1000) |
| verbose | Whether to print diagnostic information (default: FALSE) |

## Details

### Causal Interpretation:

IPTW estimates causal effects by weighting observations to create balance on measured confounders. The estimand depends on the weight type:

- **ATE**: Average treatment effect in the population
- **ATT**: Average treatment effect among those who received treatment
- **ATC**: Average treatment effect among those who did not receive treatment

### Standard Errors:

By default, uses robust (sandwich) standard errors that account for propensity score estimation uncertainty. Bootstrap confidence intervals are available as an alternative that may perform better with small samples.

### Assumptions:

1. **No unmeasured confounding**: All confounders are measured and included
2. **Positivity**: All subjects have non-zero probability of receiving either treatment
3. **Correct model specification**: Propensity score model is correctly specified

### Number Needed to Treat (NNT):

When `nnt = TRUE`, results are transformed to causal Number Needed to Treat. This represents the number of individuals who need to receive treatment to prevent one additional adverse outcome in the target population (defined by weight_type). NNT calculations preserve the causal interpretation of IPTW estimates under the assumptions of exchangeability, positivity, and consistency.

## Value

A tibble of class "riskdiff_iptw_result" containing:

**treatment_var** Character. Name of treatment variable

**rd_iptw** Numeric. IPTW-standardized risk difference OR Number Needed to Treat if nnt=TRUE

**ci_lower** Numeric. Lower confidence interval bound (RD scale or NNT scale)

**ci_upper** Numeric. Upper confidence interval bound (RD scale or NNT scale)

**p_value** Numeric. P-value for test of null hypothesis

**weight_type** Character. Type of weights used

**effective_n** Numeric. Effective sample size

**risk_treated** Numeric. Risk in treated group

**risk_control** Numeric. Risk in control group

## Examples

```
data(cachar_sample)

# Standard ATE estimation
rd_iptw <- calc_risk_diff_iptw(
  data = cachar_sample,
  outcome = "abnormal_screen",
```

```
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking")
)
print(rd_iptw)

# ATT estimation with bootstrap CI
rd_att <- calc_risk_diff_iptw(
  data = cachar_sample,
  outcome = "head_neck_abnormal",
  treatment = "tobacco_chewing",
  covariates = c("age", "sex", "residence", "areca_nut"),
  weight_type = "ATT",
  bootstrap_ci = TRUE,
  boot_n = 500
)
print(rd_att)

# Calculate causal NNT using IPTW
nnt_iptw <- calc_risk_diff_iptw(
  data = cachar_sample,
  outcome = "abnormal_screen",
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking"),
  nnt = TRUE
)
print(nnt_iptw)

# ATT-specific NNT with bootstrap CI
nnt_att <- calc_risk_diff_iptw(
  data = cachar_sample,
  outcome = "abnormal_screen",
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence"),
  weight_type = "ATT",
  bootstrap_ci = TRUE,
  boot_n = 500,
  nnt = TRUE
)
summary(nnt_att)
```

---

`check_iptw_assumptions` *Check IPTW Assumptions*

---

## Description

Provides diagnostic checks for key IPTW assumptions including positivity, balance, and model specification. Returns a comprehensive summary with recommendations for potential issues.

## Usage

```
check_iptw_assumptions(
  iptw_result,
  balance_threshold = 0.1,
  extreme_weight_threshold = 10,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| `iptw_result` | An iptw_result object from calc_iptw_weights() |
| `balance_threshold` | Threshold for acceptable standardized difference (default: 0.1) |
| `extreme_weight_threshold` | Threshold for flagging extreme weights (default: 10) |
| `verbose` | Whether to print detailed diagnostics (default: TRUE) |

## Value

A list containing:

**overall_assessment** Character indicating "PASS", "CAUTION", or "FAIL"

**positivity** List with positivity checks and recommendations

**balance** List with balance assessment and problematic variables

**weights** List with weight distribution diagnostics

**recommendations** Character vector of specific recommendations

## Examples

```
data(cachar_sample)

iptw_result <- calc_iptw_weights(
  data = cachar_sample,
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking")
)

# Check assumptions
assumptions <- check_iptw_assumptions(iptw_result)
print(assumptions$overall_assessment)
print(assumptions$recommendations)
```

`create_balance_plots` *Create Balance Plots for IPTW Analysis*

## Description

Creates visualizations to assess covariate balance before and after IPTW weighting. Includes love plots (standardized differences) and propensity score distribution plots.

## Usage

```
create_balance_plots(
  iptw_result,
  plot_type = "both",
  threshold = 0.1,
  save_plots = FALSE,
  plot_dir = "plots"
)
```

## Arguments

| | |
|---|---|
| `iptw_result` | An iptw_result object from calc_iptw_weights() |
| `plot_type` | Type of plot: "love" for standardized differences, "ps" for propensity score distributions, or "both" |
| `threshold` | Threshold for acceptable standardized difference (default: 0.1) |
| `save_plots` | Whether to save plots to files (default: FALSE) |
| `plot_dir` | Directory to save plots if save_plots=TRUE (default: "plots") |

## Details

### Love Plot:

Shows standardized differences for each covariate before and after weighting. Points represent standardized differences, with lines connecting before/after values. Horizontal lines show common thresholds (0.1, 0.25) for acceptable balance.

### Propensity Score Plot:

Shows distributions of propensity scores by treatment group before and after weighting. Good overlap indicates positivity assumption is met.

## Value

A ggplot object (if plot_type is "love" or "ps") or a list of ggplot objects (if plot_type is "both"). If ggplot2 is not available, returns a message and creates base R plots.

## Examples

```
data(cachar_sample)

# Calculate IPTW weights
iptw_result <- calc_iptw_weights(
  data = cachar_sample,
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking")
)

# Create balance plots
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plots <- create_balance_plots(iptw_result, plot_type = "both")
  print(plots$love_plot)
  print(plots$ps_plot)
}
```

---

`create_forest_plot` *Create Forest Plot for Risk Difference Results*

---

## Description

Creates a forest plot visualization of risk difference results, automatically detecting stratification variables and creating appropriate labels.

## Usage

```
create_forest_plot(results, title = "Risk Differences", max_ci_width = 50, ...)
```

## Arguments

| | |
|---|---|
| `results` | Results tibble from calc_risk_diff() |
| `title` | Plot title (default: "Risk Differences") |
| `max_ci_width` | Maximum CI width for display (default: 50) |
| `...` | Additional arguments passed to ggplot |

## Value

A ggplot object

## Examples

```
data(cachar_sample)
results <- calc_risk_diff(cachar_sample, "abnormal_screen", "areca_nut", strata = "residence")
create_forest_plot(results)
```

create_rd_table *Create Formatted Table of Risk Difference Results*

## Description

Creates a publication-ready table of risk difference results with appropriate grouping and formatting. Requires the kableExtra package for full functionality.

## Usage

```
create_rd_table(
  results,
  caption = "Risk Differences",
  include_model_type = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| results | Results tibble from calc_risk_diff() |
| caption | Table caption (default: "Risk Differences") |
| include_model_type | Whether to include model type column (default: FALSE) |
| ... | Additional arguments passed to kableExtra::kable() |

## Value

If kableExtra is available, returns a kable table object suitable for rendering in R Markdown or HTML. The table includes formatted risk differences, confidence intervals, and p-values with appropriate styling and footnotes. If kableExtra is not available, returns a formatted tibble with the same information in a basic data frame structure.

## Examples

```
data(cachar_sample)
results <- calc_risk_diff(cachar_sample, "abnormal_screen", "smoking")

# Basic table (works without kableExtra)
basic_table <- create_rd_table(results, caption = "Risk of Abnormal Cancer Screening")
print(basic_table)

# Enhanced table (requires kableExtra)
if (requireNamespace("kableExtra", quietly = TRUE)) {
  enhanced_table <- create_rd_table(
    results,
    caption = "Risk of Abnormal Cancer Screening by Smoking Status",
    include_model_type = TRUE
  )
```

```
  print(enhanced_table)
}
```

---

`create_simple_table` *Create a Simple Summary Table*

---

## Description

Creates a simple text-based summary table that doesn't require kableExtra.

## Usage

```
create_simple_table(results, title = "Risk Difference Results")
```

## Arguments

| | |
|---|---|
| `results` | Results tibble from calc_risk_diff() |
| `title` | Optional title for the table |

## Value

A formatted character vector representing the table

## Examples

```
data(cachar_sample)
results <- calc_risk_diff(cachar_sample, "abnormal_screen", "smoking")
cat(create_simple_table(results))
```

---

`create_summary_table` *Create Summary Table for Risk Difference Results*

---

## Description

Creates a formatted summary table that works with any stratification variables.

## Usage

```
create_summary_table(results, caption = "Risk Difference Results")
```

## Arguments

| | |
|---|---|
| `results` | Results tibble from calc_risk_diff() |
| `caption` | Table caption |

### Value

A data frame suitable for knitr::kable()

---

`format_risk_diff` *Format Risk Difference Results for Display*

---

### Description

Formats numerical values in risk difference results for presentation, with appropriate percentage formatting and rounding. Enhanced for v0.2.1 to handle boundary information and quality indicators with robust error handling.

### Usage

```
format_risk_diff(
  results,
  digits = 2,
  p_accuracy = 0.001,
  show_ci_method = FALSE,
  show_quality = FALSE,
  nnt_digits = 1
)
```

### Arguments

| | |
|---|---|
| `results` | Results tibble from calc_risk_diff() |
| `digits` | Number of decimal places for percentages (default: 2) |
| `p_accuracy` | Accuracy for p-values (default: 0.001) |
| `show_ci_method` | Logical indicating whether to show CI method in output (default: FALSE) |
| `show_quality` | Logical indicating whether to add quality indicators (default: TRUE) |
| `nnt_digits` | Number of decimal places for NNT formatting (default: 1) |

### Value

Tibble with additional formatted columns including:

**rd_formatted** Risk difference as formatted percentage string

**ci_formatted** Confidence interval as formatted string

**p_value_formatted** P-value with appropriate precision

**quality_indicator** Quality assessment (if show_quality = TRUE)

**ci_method_display** CI method information (if show_ci_method = TRUE)

## Examples

```
data(cachar_sample)
results <- calc_risk_diff(cachar_sample, "abnormal_screen", "areca_nut")
formatted <- format_risk_diff(results)
print(formatted)

# Show CI methods and quality indicators
formatted_detailed <- format_risk_diff(results, show_ci_method = TRUE, show_quality = TRUE)
print(formatted_detailed)

# Customize formatting
formatted_custom <- format_risk_diff(results, digits = 3, p_accuracy = 0.01, show_quality = FALSE)
print(formatted_custom)
```

---

# get_quality_legend *Get Quality Legend for Risk Difference Results*

---

## Description

Returns a legend explaining the quality indicators used in formatted results.

## Usage

```
get_quality_legend()
```

## Value

Character vector with quality indicator explanations

## Examples

```
quality_legend <- get_quality_legend()
cat(paste(quality_legend, collapse = "\n"))
```

---

# get_valid_boundary_types *Get Valid Boundary Types*

---

## Description

Returns the complete list of valid boundary types that can be returned by the boundary detection function.

### Usage

```
get_valid_boundary_types()
```

### Value

Character vector of valid boundary type names

---

## print.iptw_result *Print Method for IPTW Results*

---

### Description

Print Method for IPTW Results

### Usage

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

### Arguments

| | |
|---|---|
| x | An iptw_result object |
| ... | Additional arguments passed to print |

---

## print.nnt_iptw_result *Print Method for IPTW NNT Results*

---

### Description

Print Method for IPTW NNT Results

### Usage

```
## S3 method for class 'nnt_iptw_result'
print(x, digits = 1, ...)
```

### Arguments

| | |
|---|---|
| x | An nnt_iptw_result object from calc_risk_diff_iptw(..., nnt = TRUE) |
| digits | Number of decimal places for NNT estimates (default: 1) |
| ... | Additional arguments (ignored) |

## print.nnt_result *Print Method for NNT Results*

### Description

Print Method for NNT Results

### Usage

```
## S3 method for class 'nnt_result'
print(x, digits = 1, ...)
```

### Arguments

| | |
|---|---|
| x | An nnt_result object from calc_risk_diff(..., nnt = TRUE) |
| digits | Number of decimal places for NNT estimates (default: 1) |
| ... | Additional arguments (ignored) |

## print.riskdiff_iptw_result *Print Method for IPTW Risk Difference Results*

### Description

Print Method for IPTW Risk Difference Results

### Usage

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

### Arguments

| | |
|---|---|
| x | A riskdiff_iptw_result object |
| ... | Additional arguments passed to print |

## print.riskdiff_result *Print method for riskdiff_result objects*

### Description

Prints risk difference results in a formatted, readable way showing key statistics including risk differences, confidence intervals, model types used, and enhanced boundary case diagnostics for v0.2.1+.

### Usage

```
## S3 method for class 'riskdiff_result'
print(x, show_boundary = TRUE, show_quality = TRUE, ...)
```

### Arguments

| | |
|---|---|
| x | A riskdiff_result object from calc_risk_diff() |
| show_boundary | Logical indicating whether to show boundary case details (default: TRUE) |
| show_quality | Logical indicating whether to show quality indicators (default: TRUE) |
| ... | Additional arguments passed to print methods |

### Value

Invisibly returns the original riskdiff_result object (x). Called primarily for its side effect of printing formatted results to the console.

### Examples

```
data(cachar_sample)
result <- calc_risk_diff(cachar_sample, "abnormal_screen", "areca_nut")
print(result)

# Suppress boundary details for cleaner output
print(result, show_boundary = FALSE)
```

## summary.riskdiff_iptw_result *Summary Method for IPTW Risk Difference Results*

### Description

Provides a comprehensive summary of IPTW risk difference analysis including effect estimates, diagnostics, and interpretation guidance.

## Usage

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

## Arguments

| | |
|---|---|
| object | A riskdiff_iptw_result object |
| ... | Additional arguments (currently ignored) |

## Value

Invisibly returns the input object. Called primarily for side effects (printing summary).

## Examples

```
data(cachar_sample)

rd_iptw <- calc_risk_diff_iptw(
  data = cachar_sample,
  outcome = "abnormal_screen",
  treatment = "areca_nut",
  covariates = c("age", "sex", "residence", "smoking")
)

summary(rd_iptw)
```

# Index