# Package 'AugBalWeight'

August 21, 2026

**Title** Augmented Balancing Weights as Linear Regression

**Version** 0.1.0

**Description** Implements augmented balancing weights for causal inference and linear functional estimation based on David Bruns-Smith, Oliver Dukes, Avi Feller, and Elizabeth L. Ogburn (2026) <doi:10.1093/jrsssb/qkaf019>. Establishes numerical equivalence between augmented balancing weight estimators and single linear models with weighted regression coefficients. Provides flexible routines for double ridge (l2 balancing), double lasso (l-infinity balancing), and generalized augmented linear outcome models. Features cross-validation procedures for tuning outcome penalty parameters, covariate balance, and Riesz loss. Supports robust influence-function-based standard errors, bootstrap confidence intervals, balance diagnostic tools, and counterfactual prediction for treatment effects such as average treatment effect (ATE) and average treatment effect on the treated (ATT), expanding upon the doubly robust estimation framework established by Robins, Rotnitzky, and Zhao (1994) <doi:10.1080/01621459.1994.10476818> and Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) <doi:10.1111/ectj.12097>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 4.0.0)

**Imports** stats, graphics, grDevices, utils

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Shikhar Tyagi [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1606-0844>),
Arvind Pandey [aut],
Bhupendra Singh [aut],
Vrijesh Tripathi [aut]

**Maintainer** Shikhar Tyagi <shikhar1093tyagi@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-08-21 12:41:17 UTC

# Contents



---

`aug_bal_ate` *Average Treatment Effect (ATE) Estimation via Augmented Balancing Weights*

---

## Description

Estimates Average Treatment Effect E[Y(1) - Y(0)] using augmented balancing weights.

## Usage

```
aug_bal_ate(
  Y,
  Z,
  X,
  type = "l2",
  outcome_model = "ridge",
  lambda = NULL,
  delta = NULL,
  tuning_method = "cv_outcome",
  alpha = 0.05
)
```

## Arguments

| | |
|---|---|
| `Y` | Outcome vector (n x 1). |
| `Z` | Binary treatment indicator (1 for treated, 0 for control). |
| `X` | Covariate feature matrix (n x d). |

| | |
|---|---|
| type | Weight penalty type: "l2" or "linf". Default is "l2". |
| outcome_model | Outcome model type: "ridge", "lasso", or "ols". Default is "ridge". |
| lambda | Outcome model penalty parameter. If NULL, auto-tuned. |
| delta | Weighting model penalty parameter. If NULL, auto-tuned. |
| tuning_method | Cross-validation method: "cv_outcome", "cv_balance", or "cv_riesz". |
| alpha | Significance level for confidence intervals. Default is 0.05. |

## Value

An object of class `augbalweight` representing the ATE estimator.

## References

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

## Examples

```
set.seed(123)
n <- 120
X <- matrix(rnorm(n * 4), n, 4)
Z <- rbinom(n, 1, 0.5)
Y <- 2 * Z + X %*% c(1, -1, 0.5, 0) + rnorm(n)
fit_ate <- aug_bal_ate(Y, Z, X)
fit_ate$estimate
```

---

# aug_bal_att *Average Treatment Effect on the Treated (ATT) Estimation*

---

## Description

Estimates Average Treatment Effect on the Treated E[Y(1) - Y(0) | Z = 1] using augmented balancing weights.

## Usage

```
aug_bal_att(
  Y,
  Z,
  X,
  type = "l2",
  outcome_model = "ridge",
  lambda = NULL,
  delta = NULL,
  tuning_method = "cv_outcome",
  alpha = 0.05
)
```

### Arguments

| | |
|---|---|
| Y | Outcome vector (n x 1). |
| Z | Binary treatment indicator (1 for treated, 0 for control). |
| X | Covariate feature matrix (n x d). |
| type | Weight penalty type: "l2" or "linf". Default is "l2". |
| outcome_model | Outcome model type: "ridge", "lasso", or "ols". Default is "ridge". |
| lambda | Outcome model penalty parameter. If NULL, auto-tuned. |
| delta | Weighting model penalty parameter. If NULL, auto-tuned. |
| tuning_method | Cross-validation method: "cv_outcome", "cv_balance", or "cv_riesz". |
| alpha | Significance level for confidence intervals. Default is 0.05. |

### Value

An object of class `augbalweight` representing the ATT estimator.

### References

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

### Examples

```
set.seed(123)
n <- 120
X <- matrix(rnorm(n * 4), n, 4)
Z <- rbinom(n, 1, 0.5)
Y <- 2.5 * Z + X %*% c(1, -1, 0.5, 0) + rnorm(n)
fit_att <- aug_bal_att(Y, Z, X)
fit_att$estimate
```

---

## aug_bal_weight *Main Fitting Function for Augmented Balancing Weights*

---

### Description

Fits augmented balancing weights for linear functional estimation and causal inference based on Bruns-Smith et al. (2026).

## Usage

```
aug_bal_weight(
  Y,
  X_p,
  X_q,
  Z = NULL,
  estimand = c("custom", "ATE", "ATT", "ATC"),
  type = c("l2", "linf", "ols"),
  outcome_model = c("ridge", "lasso", "ols"),
  lambda = NULL,
  delta = NULL,
  tuning_method = c("cv_outcome", "cv_balance", "cv_riesz"),
  normalize = TRUE,
  nonnegative = FALSE,
  nfolds = 5,
  nboot = 0,
  alpha = 0.05
)
```

## Arguments

| | |
|---|---|
| Y | Outcome vector of source sample. |
| X_p | Feature matrix of source sample (n_p x d). |
| X_q | Feature matrix of target sample (n_q x d) or vector of target feature means (Phi_q, 1 x d). |
| Z | Optional binary treatment vector (1 for treated, 0 for control) if custom data is supplied. |
| estimand | Target estimand: "custom", "ATE", "ATT", or "ATC". Default is "custom". |
| type | Weight penalty type: "l2" (ridge-augmented), "linf" (lasso-augmented), or "ols". Default is "l2". |
| outcome_model | Outcome model type: "ridge", "lasso", or "ols". Default is "ridge". |
| lambda | Outcome model penalty. If NULL, auto-tuned via cross-validation. |
| delta | Weighting model penalty. If NULL, auto-tuned via cross-validation. |
| tuning_method | Cross-validation method: "cv_outcome", "cv_balance", or "cv_riesz". Default is "cv_outcome". |
| normalize | Logical, whether to normalize weights to sum to n_p. Default is TRUE. |
| nonnegative | Logical, whether to enforce non-negative weights. Default is FALSE. |
| nfolds | Number of cross-validation folds. Default is 5. |
| nboot | Number of bootstrap replicates for standard error and confidence interval estimation. Default is 0. |
| alpha | Significance level for confidence intervals. Default is 0.05. |

## Value

An object of class `augbalweight` containing:

| | |
|---|---|
| `estimate` | Augmented balancing weight point estimate. |
| `se` | Robust standard error. |
| `ci` | 100 * (1 - alpha)% confidence interval. |
| `beta_aug` | Implied single linear regression coefficients. |
| `beta_ols` | Unregularized OLS regression coefficients. |
| `beta_reg` | Base outcome model coefficients. |
| `weights` | Estimated balancing weights vector. |
| `feature_shift_obs` | Observed mean feature shift. |
| `feature_shift_imp` | Implied mean feature shift from balancing weights. |
| `lambda` | Selected outcome model penalty parameter. |
| `delta` | Selected weighting model penalty parameter. |
| `call` | Matched function call. |

## References

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(100 * 5), 100, 5)
Y <- X_p %*% c(1, -0.5, 2, 0, 0) + rnorm(100)
target_mean <- colMeans(matrix(rnorm(50 * 5, mean = 0.4), 50, 5))
fit <- aug_bal_weight(Y, X_p, target_mean, type = "l2", outcome_model = "ridge")
print(fit)
summary(fit)
```

---

# `coef.augbalweight` *Extract Implied Augmented Coefficients*

---

## Description

Extract Implied Augmented Coefficients

## Usage

```
## S3 method for class 'augbalweight'
coef(object, ...)
```

### Arguments

| | |
|---|---|
| object | Object of class augbalweight. |
| ... | Additional arguments. |

### Value

A numeric vector containing the implied single linear regression coefficients (beta_aug) estimated by the augmented balancing weights model.

---

## compute_balancing_weights

*Compute Standalone Balancing Weights*

---

### Description

Computes balancing weights w and implied feature shifts for any source and target population (Bruns-Smith et al., 2026).

### Usage

```
compute_balancing_weights(
  X_p,
  target_mean,
  type = c("l2", "linf"),
  delta = 0.1,
  normalize = TRUE
)
```

### Arguments

| | |
|---|---|
| X_p | Source feature matrix (n_p x d). |
| target_mean | Vector of target feature means (Phi_q, 1 x d) or target feature matrix (n_q x d). |
| type | Weight penalty type: "l2" or "linf". Default is "l2". |
| delta | Balancing weights penalty parameter. Default is 0.1. |
| normalize | Logical, whether to normalize weights to sum to n_p. Default is TRUE. |

### Value

A list containing:

| | |
|---|---|
| weights | Estimated balancing weights vector (n_p x 1). |
| feature_shift_obs | Observed mean feature shift (Phi_q - bar_Phi_p). |
| feature_shift_imp | Implied mean feature shift from balancing weights. |

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(60 * 3), 60, 3)
target_mean <- colMeans(matrix(rnorm(30 * 3, mean = 0.5), 30, 3))
res <- compute_balancing_weights(X_p, target_mean, type = "l2", delta = 0.1)
summary(res$weights)
```

---

`double_lasso` *Lasso-Augmented L-infinity Balancing Weights (Double Lasso)*

---

## Description

Fits a lasso outcome model augmented with l-infinity balancing weights as described in Bruns-Smith et al. (2026). Demonstrates the double selection phenomenon where active features equal the union of outcome and balance sets.

## Usage

```
double_lasso(
  Y,
  X_p,
  target_mean,
  lambda = 0.05,
  delta = 0.05,
  normalize = TRUE
)
```

## Arguments

| | |
|---|---|
| `Y` | Outcome vector of source sample (n_p x 1). |
| `X_p` | Source feature matrix (n_p x d). |
| `target_mean` | Vector of target feature means (Phi_q, 1 x d). |
| `lambda` | Outcome model lasso regularization parameter. Default is 0.05. |
| `delta` | Balancing weights l-infinity threshold parameter. Default is 0.05. |
| `normalize` | Logical, whether to normalize weights to sum to n_p. Default is TRUE. |

## Value

A list containing:

| | |
|---|---|
| `estimate` | Augmented balancing weight point estimate. |
| `beta_aug` | Implied augmented regression coefficients. |
| `beta_ols` | Unregularized OLS regression coefficients. |
| `beta_lasso` | Lasso outcome model regression coefficients. |
| `weights` | Estimated l-infinity balancing weights vector. |

`feature_shift_obs`
: Observed mean feature shift (Phi_q - bar_Phi_p).

`feature_shift_imp`
: Implied mean feature shift from balancing weights.

`active_outcome` Active index set for outcome model (I_lambda).

`active_balance` Active index set for balance model (I_delta).

`active_union` Active index set for augmented estimator (I_aug = I_lambda U I_delta).

## References

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(100 * 6), 100, 6)
Y <- X_p %*% c(2, 0, 1.5, 0, 0, -1) + rnorm(100)
target_mean <- colMeans(matrix(rnorm(50 * 6, mean = 0.3), 50, 6))
fit <- double_lasso(Y, X_p, target_mean, lambda = 0.05, delta = 0.05)
fit$estimate
fit$active_union
```

---

# `double_ridge` *Ridge-Augmented L2 Balancing Weights (Double Ridge)*

---

## Description

Fits a ridge outcome model augmented with l2 balancing weights as established in Bruns-Smith et al. (2026). Computes point estimates, implied single linear regression coefficients, balancing weights, and feature shifts.

## Usage

```
double_ridge(Y, X_p, target_mean, lambda = 0.1, delta = 0.1, normalize = TRUE)
```

## Arguments

`Y` Outcome vector of source sample (n_p x 1).

`X_p` Source feature matrix (n_p x d).

`target_mean` Vector of target feature means (Phi_q, 1 x d).

`lambda` Outcome model ridge regularization parameter. Default is 0.1.

`delta` Balancing weights l2 regularization parameter. Default is 0.1.

`normalize` Logical, whether to normalize weights to sum to n_p. Default is TRUE.

## Value

A list containing:

| | |
|---|---|
| `estimate` | Augmented balancing weight point estimate. |
| `beta_aug` | Implied augmented regression coefficients. |
| `beta_ols` | Unregularized OLS regression coefficients. |
| `beta_ridge` | Ridge outcome model regression coefficients. |
| `weights` | Estimated l2 balancing weights vector. |
| `feature_shift_obs` | Observed mean feature shift (Phi_q - bar_Phi_p). |
| `feature_shift_imp` | Implied mean feature shift from balancing weights. |
| `gamma_equiv` | Equivalent single ridge regularization parameter. |
| `shrinkage_factors` | Element-wise shrinkage factors a_j. |

## References

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(100 * 5), 100, 5)
Y <- X_p %*% c(1, 2, 0, -1, 0.5) + rnorm(100)
target_mean <- colMeans(matrix(rnorm(50 * 5, mean = 0.5), 50, 5))
fit <- double_ridge(Y, X_p, target_mean, lambda = 0.1, delta = 0.05)
fit$estimate
```

---

# `lalonde_data` *LaLonde (1986) Job Training Demonstration Dataset*

---

## Description

A canonical dataset evaluated by LaLonde (1986) and Dehejia & Wahba (1999) from the National Supported Work (NSW) Demonstration. Used for demonstrating augmented balancing weights in causal inference and treatment effect estimation (Bruns-Smith et al., 2026).

## Usage

```
lalonde_data
```

## Format

A data frame with 445 observations and 12 variables:

**treat** Binary treatment indicator (1 if in job training program, 0 otherwise).

**age** Age of individual in years.

**educ** Years of completed education.

**black** Indicator for Black ethnicity (1 if Black, 0 otherwise).

**hisp** Indicator for Hispanic ethnicity (1 if Hispanic, 0 otherwise).

**married** Indicator for marital status (1 if married, 0 otherwise).

**re74** Real earnings in 1974 (US dollars).

**re75** Real earnings in 1975 (US dollars).

**re78** Real earnings in 1978 (outcome variable, US dollars).

**age2** Age squared.

**educ2** Years of education squared.

**re742** 1974 earnings squared.

## Source

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review*, 76(4), 604-620.

Dehejia, R. H., & Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association*, 94(448), 1053-1062.

Bruns-Smith, D., Dukes, O., Feller, A., & Ogburn, E. L. (2026). Augmented balancing weights as linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(3), 699-723. doi:10.1093/jrsssb/qkaf019

---

# plot.augbalweight *Diagnostic and Visualization Plot for Augmented Balancing Weights*

---

## Description

Generates balance diagnostic and covariate shift plots for an `augbalweight` object.

## Usage

```
## S3 method for class 'augbalweight'
plot(x, which = 1, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class `augbalweight`. |
| `which` | Plot type: 1 for balance diagnostic (observed vs implied shift), 2 for weight distribution histogram. Default is 1. |
| `...` | Additional graphical arguments. |

## Value

Invisibly returns the input augbalweight object x. Called for its side effect of generating diagnostic plots (covariate balance diagnostic when which = 1, or weight distribution histogram when which = 2).

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(80 * 4), 80, 4)
Y <- X_p %*% c(1, 2, 0, -1) + rnorm(80)
target_mean <- colMeans(matrix(rnorm(40 * 4, mean = 0.5), 40, 4))
fit <- aug_bal_weight(Y, X_p, target_mean)
plot(fit, which = 1)
plot(fit, which = 2)
```

---

# predict.augbalweight *Counterfactual Outcome Prediction for New Data*

---

## Description

Counterfactual Outcome Prediction for New Data

## Usage

```
## S3 method for class 'augbalweight'
predict(object, newdata, ...)
```

## Arguments

| | |
|---|---|
| object | Object of class augbalweight. |
| newdata | Matrix or data frame of new target covariates. |
| ... | Additional arguments. |

## Value

A numeric vector of counterfactual outcome predictions for target observations in newdata, computed using the implied augmented regression coefficients.

## print.augbalweight *Print Method for augbalweight*

### Description

Print Method for augbalweight

### Usage

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

### Arguments

| | |
|---|---|
| x | An object of class augbalweight. |
| ... | Additional arguments passed to print. |

### Value

Invisibly returns the input augbalweight object x. Called for its side effect of printing model estimation details, point estimates, standard errors, confidence intervals, and hyperparameter settings to the console.

## print.summary.augbalweight *Print Summary for augbalweight*

### Description

Print Summary for augbalweight

### Usage

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

### Arguments

| | |
|---|---|
| x | Summary object of class summary.augbalweight. |
| ... | Additional arguments. |

### Value

Invisibly returns the input summary.augbalweight object x. Called for its side effect of printing formatted coefficient tables and summary statistics to the console.

## summary.augbalweight *Summary Method for augbalweight*

### Description

Summary Method for augbalweight

### Usage

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

### Arguments

| | |
|---|---|
| object | An object of class augbalweight. |
| ... | Additional arguments. |

### Value

An object of class summary.augbalweight containing structured summary statistics:

| | |
|---|---|
| estimate | Point estimate for the target population functional. |
| se | Robust standard error of the estimate. |
| ci | Confidence interval at the specified significance level alpha. |
| coef_table | Data frame comparing unregularized OLS, base outcome model, and implied augmented regression coefficients alongside observed vs. implied feature shifts. |
| weight_summary | Summary statistics of the estimated balancing weights (min, 1st quartile, median, mean, 3rd quartile, max). |
| type | Weight penalty type used ("l2", "linf", or "ols"). |
| outcome_model | Base outcome model type used ("ridge", "lasso", or "ols"). |
| lambda | Selected outcome regularization parameter. |
| delta | Selected weighting regularization parameter. |

## tune_hyperparameters *Cross-Validation Hyperparameter Tuning for Augmented Balancing Weights*

### Description

Automatically selects outcome penalty lambda and weighting parameter delta using CV schemes described in Bruns-Smith et al. (2026).

## Usage

```
tune_hyperparameters(
  Y,
  X_p,
  target_mean,
  type = "l2",
  outcome_model = "ridge",
  tuning_method = "cv_outcome",
  nfolds = 5
)
```

## Arguments

| | |
|---|---|
| Y | Outcome vector (n_p x 1). |
| X_p | Source feature matrix (n_p x d). |
| target_mean | Vector of target feature means (Phi_q, 1 x d). |
| type | Balancing weight type: "l2" or "linf". |
| outcome_model | Outcome model type: "ridge", "lasso", or "ols". |
| tuning_method | Tuning strategy: "cv_outcome", "cv_balance", or "cv_riesz". |
| nfolds | Number of cross-validation folds. Default is 5. |

## Value

A list containing optimal parameters:

| | |
|---|---|
| lambda | Optimal outcome model parameter. |
| delta | Optimal balancing weight parameter. |
| grid_results | Matrix of grid search CV errors. |

## Examples

```
set.seed(123)
X_p <- matrix(rnorm(80 * 4), 80, 4)
Y <- X_p %*% c(1, 1, -1, 0) + rnorm(80)
target_mean <- colMeans(matrix(rnorm(40 * 4, mean = 0.2), 40, 4))
tuned <- tune_hyperparameters(Y, X_p, target_mean, tuning_method = "cv_outcome")
tuned$lambda
tuned$delta
```

`weights.augbalweight` *Extract Balancing Weights*

## Description

Extract Balancing Weights

## Usage

```
## S3 method for class 'augbalweight'
weights(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | Object of class `augbalweight`. |
| `...` | Additional arguments. |

## Value

A numeric vector containing the estimated observation-level balancing weights for the source sample.

# Index