Package 'PVBcorrect'

May 7, 2026

Type Package

Title Partial Verification Bias Correction for Diagnostic Accuracy

Version 0.3.1

Maintainer Wan Nor Arifin <wnarifin@gmail.com>

URL https://github.com/wnarifin/PVBcorrect/

Description Performs partial verification bias (PVB) correction for binary diagnostic tests, where PVB arises from selective patient verification in diagnostic accuracy studies. Supports correction of important accuracy measures -- sensitivity, specificity, positive predictive values and negative predictive value -- under missing-at-random and missing-not-at-random missing data mechanisms. Available methods and references are ``Begg and Greenes' methods'' in Alonzo & Pepe (2005) <doi:10.1111/j.1467-9876.2005.00477.x> and deGroot et al. (2011) <doi:10.1016/j.annepidem.2010.10.004>; ``Multiple imputation'' in Harel & Zhou (2006) <doi:10.1002/sim.2494>, ``EM-based logistic regression'' in Kosinski & Barnhart (2003) <doi:10.1111/1541-0420.00019>; ``Inverse probability weighting'' in Alonzo & Pepe (2005) <doi:10.1111/j.1467-9876.2005.00477.x>; ``Inverse probability bootstrap sampling'' in Nahorniak et al. (2015) <doi:10.1371/journal.pone.0131765> and Arifin & Yusof (2022) <doi:10.3390/diagnostics12112839>; ``Scaled inverse probability resampling methods'' in Arifin & Yusof (2025) <doi:10.1371/journal.pone.0321440>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports boot, mice

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

NeedsCompilation no

Author Wan Nor Arifin [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0001-7786-4251>)

Repository CRAN

Date/Publication 2025-10-08 08:40:18 UTC

Contents

`PVBcorrect-package` *PVBcorrect: A package to perform partial verification bias correction for estimates of accuracy measures in diagnostic accuracy studies*

Description

The package contains a number of functions to perform partial verification bias (PVB) correction for estimates of accuracy measures in diagnostic accuracy studies. The available methods are: Begg and Greenes' method (as extended by Alonzo & Pepe, 2005), Begg and Greenes' method 1 and 2 (with PPV and NPV as extended by deGroot et al, 2011), Inverse Probability Bootstrap (IPB) sampling method (Arifin & Yusof, 2022; Nahorniak et al., 2015), Scaled Inverse Probability Resampling methods (Arifin & Yusof, 2023; Arifin & Yusof, 2025), multiple imputation method by logistic regression (Harel & Zhou, 2006), and EM-based logistic regression method (Kosinski & Barnhart, 2003).

General function

`view_table`

PVB correction main functions

acc_cca, acc_ebg, acc_ipb, acc_sipw, acc_mi, acc_em

PVB correction additional functions

acc_bg, acc_dg1, acc_dg2

Data set

cad_pvb

Author(s)

Maintainer: Wan Nor Arifin <wnarifin@gmail.com> (ORCID) [copyright holder]

References

1. Alonzo, T. A., & Pepe, M. S. (2005). Assessing accuracy of a continuous screening test in the presence of verification bias. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54(1), 173–190.
2. Arifin, W. N., & Yusof, U. K. (2025). Partial Verification Bias Correction Using Scaled Inverse Probability Resampling for Binary Diagnostic Tests. medRxiv. https://doi.org/10.1101/2025.03.09.25323631
3. Arifin, W. N., & Yusof, U. K. (2022). Partial Verification Bias Correction Using Inverse Probability Bootstrap Sampling for Binary Diagnostic Tests. Diagnostics, 12(11), 2839.
4. Arifin, W. N. (2023). Partial verification bias correction in diagnostic accuracy studies using propensity score-based methods (PhD thesis, Universiti Sains Malaysia). https://erepo.usm.my/handle/123456789/1918
5. Arifin, W. N., & Yusof, U. K. (2022). Partial Verification Bias Correction Using Inverse Probability Bootstrap Sampling for Binary Diagnostic Tests. Diagnostics, 12, 2839.
6. Begg, C. B., & Greenes, R. A. (1983). Assessment of diagnostic tests when disease verification is subject to selection bias. Biometrics, 207–215.
7. de Groot, J. A. H., Janssen, K. J. M., Zwinderman, A. H., Bossuyt, P. M. M., Reitsma, J. B., & Moons, K. G. M. (2011). Correcting for partial verification bias: a comparison of methods. Annals of Epidemiology, 21(2), 139–148.
8. Harel, O., & Zhou, X.-H. (2006). Multiple imputation for correcting verification bias. Statistics in Medicine, 25(22), 3769–3786.
9. He, H., & McDermott, M. P. (2012). A robust method using propensity score stratification for correcting verification bias for binary tests. Biostatistics, 13(1), 32–47.
10. Kosinski, A. S., & Barnhart, H. X. (2003). Accounting for nonignorable verification bias in assessment of diagnostic tests. Biometrics, 59(1), 163–171.

See Also

Useful links:

- https://github.com/wnarifin/PVBcorrect/

acc_bg *PVB correction by Begg and Greenes' method with asymptotic normal CI*

Description

PVB correction by Begg and Greenes' method with asymptotic normal CI. This is limited to no covariate.

Usage

```
acc_bg(data, test, disease, ci = FALSE, ci_level = 0.95, description = TRUE)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

acc_results The accuracy results.

References

1. Begg, C. B., & Greenes, R. A. (1983). Assessment of diagnostic tests when disease verification is subject to selection bias. Biometrics, 207–215.
2. Harel, O., & Zhou, X.-H. (2006). Multiple imputation for correcting verification bias. Statistics in Medicine, 25(22), 3769–3786.
3. Zhou, X.-H. (1993). Maximum likelihood estimators of sensitivity and specificity corrected for verification bias. Communications in Statistics-Theory and Methods, 22(11), 3177–3198.
4. Zhou, X.-H. (1994). Effect of verification bias on positive and negative predictive values. Statistics in Medicine, 13(17), 1737–1745.
5. Zhou, X.-H., Obuchowski, N. A., & McClish, D. K. (2011). Statistical Methods in Diagnostic Medicine (2nd ed.). John Wiley & Sons.

Examples

```
acc_bg(data = cad_pvb, test = "T", disease = "D")  # equivalent to result by acc_ebg()
acc_bg(data = cad_pvb, test = "T", disease = "D", ci = TRUE)
  # the CIs are slightly differerent from result by acc_ebg()
```

---

acc_cca *Complete Case Analysis, CCA*

Description

Perform Complete Case Analysis, CCA, used for complete data and multiple imputation, MI.

Usage

```
acc_cca(data, test, disease, ci = FALSE, ci_level = 0.95, description = TRUE)
```

Arguments

`data`	A data frame, with at least "Test" and "Disease" variables.
`test`	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
`disease`	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
`ci`	View confidence interval (CI). The default is `FALSE`.
`ci_level`	Set the CI width. The default is 0.95 i.e. 95% CI.
`description`	Print the name of this analysis. The default is `TRUE`. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

acc_results The accuracy results.

Examples

```
acc_cca(data = cad_pvb, test = "T", disease = "D")
acc_cca(data = cad_pvb, test = "T", disease = "D", ci = TRUE)
```

acc_dg1 *PVB correction by Begg and Greenes' method 1 (deGroot et al, no covariate)*

Description

Perform PVB correction by Begg and Greenes' method 1 as described in deGroot et al (2011), in which it also includes PPV and NPV calculation.

Usage

```
acc_dg1(data, test, disease, description = TRUE)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A data frame object containing the accuracy results.

References

1. de Groot, J. A. H., Janssen, K. J. M., Zwinderman, A. H., Bossuyt, P. M. M., Reitsma, J. B., & Moons, K. G. M. (2011). Correcting for partial verification bias: a comparison of methods. Annals of Epidemiology, 21(2), 139–148.

Examples

```
acc_dg1(data = cad_pvb, test = "T", disease = "D")  # equivalent to result by acc_ebg()
```

acc_dg2 *PVB correction by Begg and Greenes' method 2 (deGroot et al, one covariate)*

Description

Perform PVB correction by Begg and Greenes' method 2 as described in deGroot et al (2011), in which it also includes PPV and NPV calculation. This is limited to only one covariate.

Usage

```
acc_dg2(data, test, disease, covariate, description = TRUE)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A data frame object containing the accuracy results.

References

1. de Groot, J. A. H., Janssen, K. J. M., Zwinderman, A. H., Bossuyt, P. M. M., Reitsma, J. B., & Moons, K. G. M. (2011). Correcting for partial verification bias: a comparison of methods. Annals of Epidemiology, 21(2), 139–148.

Examples

```
acc_dg2(data = cad_pvb, test = "T", disease = "D", covariate = "X3")
  # equivalent to acc_ebg(), saturated_model
```

acc_ebg *PVB correction by extended Begg and Greenes' method*

Description

Perform PVB correction by Begg and Greenes' method (as extended by Alonzo & Pepe, 2005).

Usage

```
acc_ebg(
  data,
  test,
  disease,
  covariate = NULL,
  saturated_model = FALSE,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "basic",
  R = 999,
  seednum = NULL,
  show_fit = FALSE,
  show_boot = FALSE,
  r_print_freq = 100,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
saturated_model	Set as TRUE to obtain the original Begg and Greenes' (1983) when all possible interactions are included.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
ci_type	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI. See boot.ci for details.
R	The number of bootstrap samples. Default R = 999.

seednum	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
show_fit	Set to TRUE to view model fit summary for the logistic regression model.
show_boot	Set to TRUE to show bootstrap iterations.
r_print_freq	Print the current bootstrap sample number at each specified interval. Default r_print_freq = 100.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

boot_data An object of class "boot" from boot. Contains Sensitivity, Specificity, PPV, and NPV

boot_ci_data A list of objects of type "bootci" from boot.ci. Contains Sensitivity, Specificity, PPV, NPV.

acc_results The accuracy results.

References

1. Alonzo, T. A., & Pepe, M. S. (2005). Assessing accuracy of a continuous screening test in the presence of verification bias. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54(1), 173–190.
2. Begg, C. B., & Greenes, R. A. (1983). Assessment of diagnostic tests when disease verification is subject to selection bias. Biometrics, 207–215.
3. He, H., & McDermott, M. P. (2012). A robust method using propensity score stratification for correcting verification bias for binary tests. Biostatistics, 13(1), 32–47.

Examples

```
# point estimates
acc_ebg(data = cad_pvb, test = "T", disease = "D")
acc_ebg(data = cad_pvb, test = "T", disease = "D", covariate = "X3")

# with bootstrapped confidence interval
acc_ebg(data = cad_pvb, test = "T", disease = "D", ci = TRUE, seednum = 12345)
```

---

acc_em *PVB correction by EM-based logistic regression method*

Description

Perform PVB correction by EM-based logistic regression method.

Usage

```
acc_em(
  data,
  test,
  disease,
  covariate = NULL,
  mnar = TRUE,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "basic",
  R = 999,
  seednum = NULL,
  show_t = TRUE,
  t_max = 500,
  cutoff = 1e-04,
  t_print_freq = 100,
  return_t = FALSE,
  r_print_freq = 100,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
mnar	The default is assuming missing not at random (MNAR) missing data mechanism, MNAR = TRUE. Set this to FALSE to obtain results assuming missing at random (MAR) missing data mechanism. This will be equivalent to using acc_ebg.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
ci_type	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI. See boot.ci for details.
R	The number of bootstrap samples. Default R = 999.
seednum	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
show_t	Print the current EM iteration number t. The default is TRUE.
t_max	The maximum iteration number for EM. Default t_max = 500. It is recommended to increase the number when covariates are included.

`cutoff` The cutoff value for the minimum change between iteration. This defines the convergence of the EM procedure. Default `cutoff = 0.0001`. This can be set to a larger value to test the procedure.

`t_print_freq` Print the current EM iteration number t at each specified interval. Default `t_print_freq = 100`.

`return_t` Return the final EM iteration number t. This can be used for the purpose of checking the EM convergence. The default is `FALSE`, but is set to TRUE when `ci = TRUE`.

`r_print_freq` Print the current bootstrap sample number at each specified interval. Default `r_print_freq = 100`.

`description` Print the name of this analysis. The default is `TRUE`. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

boot_data An object of class "boot" from `boot`. Contains Sensitivity, Specificity, PPV, NPV and t (i.e. EM iteration taken for convergence). Use acc_em_object$boot_data$t[,5] to check the t.

boot_ci_data A list of objects of type "bootci" from from `boot.ci`. Contains Sensitivity, Specificity, PPV, and NPV.

acc_results The accuracy results.

References

1. Kosinski, A. S., & Barnhart, H. X. (2003). Accounting for nonignorable verification bias in assessment of diagnostic tests. Biometrics, 59(1), 163–171.

Examples

```
# For sample run, test with low R boot number, low t_max, low cutoff
# The results will not be good

# without covariate
em_out = acc_em(data = cad_pvb, test = "T", disease = "D", ci = TRUE, seednum = 12345,
                R = 2, t_max = 100, cutoff = 0.01)
em_out$acc_results
em_out$boot_data$t  # bootstrapped data, 1:5 columns are Sn, Sp, PPV, NPV,
                    # t (i.e. EM iteration taken for convergence)
em_out$boot_ci_data
```

acc_ipb *PVB correction by inverse probability bootstrap sampling (IPB)*

Description

Perform PVB correction by inverse probability bootstrap sampling.

Usage

```
acc_ipb(
  data,
  test,
  disease,
  covariate = NULL,
  saturated_model = FALSE,
  option = 2,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "norm",
  b = 1000,
  seednum = NULL,
  return_data = FALSE,
  return_detail = FALSE,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
saturated_model	Set as TRUE to obtain the original Begg and Greenes' (1983) when all possible interactions are included.
option	1 = IPW weight, 2 = W_h weight, described in Arifin (2023), modified weight of Krautenbacher (2017). The default is option = 2. For small weights, option = 2 is more stable (Arifin, 2023).
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.

`ci_type`	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI.
`b`	The number of bootstrap samples, b.
`seednum`	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
`return_data`	Return data for the bootstrapped samples.
`return_detail`	Return accuracy measures for each of the bootstrapped samples.
`description`	Print the name of this analysis. The default is `TRUE`. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

data_each_sample Raw data for each bootstrap sample, available with `return_data = TRUE`

acc_each_sample Accuracy results for each bootstrap sample, available with `return_detail = TRUE`

acc_results The accuracy results.

References

1. Arifin, W. N., & Yusof, U. K. (2022). Partial Verification Bias Correction Using Inverse Probability Bootstrap Sampling for Binary Diagnostic Tests. Diagnostics, 12(11), 2839.
2. Arifin, W. N. (2023). Partial verification bias correction in diagnostic accuracy studies using propensity score-based methods (PhD thesis, Universiti Sains Malaysia). https://erepo.usm.my/handle/123456789/1918
3. Krautenbacher, N., Theis, F. J., & Fuchs, C. (2017). Correcting Classifiers for Sample Selection Bias in Two-Phase Case-Control Studies. Computational and Mathematical Methods in Medicine, 2017, 1–18.
4. Nahorniak, M., Larsen, D. P., Volk, C., & Jordan, C. E. (2015). Using inverse probability bootstrap sampling to eliminate sample induced bias in model based analysis of unequal probability samples. PLoS One, 10(6), e0131765.

Examples

```
# point estimates
acc_ipb(data = cad_pvb, test = "T", disease = "D", b = 100, seednum = 12345)
acc_ipb(data = cad_pvb, test = "T", disease = "D", covariate = "X3",
        b = 100, seednum = 12345)

# with confidence interval
acc_ipb(data = cad_pvb, test = "T", disease = "D", ci = TRUE,
        b = 100, seednum = 12345)  # use small b for testing
```

acc_ipw *PVB correction by Inverse Probability Weighting Estimator method*

Description

Perform PVB correction by Inverse Probability Weighting Estimator method (Alonzo & Pepe, 2005).

Usage

```
acc_ipw(
  data,
  test,
  disease,
  covariate = NULL,
  saturated_model = FALSE,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "basic",
  R = 999,
  seednum = NULL,
  show_fit = FALSE,
  show_boot = FALSE,
  r_print_freq = 100,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
saturated_model	Set as TRUE to obtain the original Begg and Greenes' (1983) when all possible interactions are included.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
ci_type	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI. See boot.ci for details.
R	The number of bootstrap samples. Default R = 999.

seednum	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
show_fit	Set to TRUE to view model fit summary for the logistic regression model.
show_boot	Set to TRUE to show bootstrap iterations.
r_print_freq	Print the current bootstrap sample number at each specified interval. Default r_print_freq = 100.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

acc_results The accuracy results.

References

1. Alonzo, T. A., & Pepe, M. S. (2005). Assessing accuracy of a continuous screening test in the presence of verification bias. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54(1), 173–190.
2. He, H., & McDermott, M. P. (2012). A robust method using propensity score stratification for correcting verification bias for binary tests. Biostatistics, 13(1), 32–47.

Examples

```
# point estimates
acc_ipw(data = cad_pvb, test = "T", disease = "D")
acc_ipw(data = cad_pvb, test = "T", disease = "D", covariate = "X3")

# with bootstrapped confidence interval
acc_ipw(data = cad_pvb, test = "T", disease = "D", ci = TRUE, R = 99, seednum = 12345)
```

---

acc_mi *PVB correction by multiple imputation*

Description

Perform PVB correction by multiple imputation.

Usage

```
acc_mi(
  data,
  test,
  disease,
  covariate = NULL,
  ci = FALSE,
```

```
  ci_level = 0.95,
  m = 100,
  seednum = NA,
  method = "logreg",
  mi_print = FALSE,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
m	The number of imputation, m.
seednum	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
method	Imputation method. The default is "logreg". Other allowed methods are "logreg.boot", "pmm", "midastouch", "sample", "cart", "rf". See mice for details of these methods.
mi_print	Print multiple imputation history on console. This is mice print option. The default is FALSE.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

acc_results The accuracy results.

References

1. Harel, O., & Zhou, X.-H. (2006). Multiple imputation for correcting verification bias. Statistics in Medicine, 25(22), 3769–3786.

Examples

```
# with logreg
acc_mi(data = cad_pvb, test = "T", disease = "D", ci = TRUE, seednum = 12345, m = 5)

# with other imputation method. e.g. predictive mean matching "pmm"
acc_mi(data = cad_pvb, test = "T", disease = "D", ci = TRUE, seednum = 12345, m = 5,
       method = "pmm")

# with covariate and confidence interval
acc_mi(data = cad_pvb, test = "T", disease = "D", covariate = "X3",
       ci = TRUE, seednum = 12345, m = 5)
```

`acc_sipw` *PVB correction by scaled inverse probability weighted resampling (SIPW)*

Description

Perform PVB correction by scaled inverse probability weighted resampling.

Usage

```
acc_sipw(
  data,
  test,
  disease,
  covariate = NULL,
  saturated_model = FALSE,
  option = 2,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "basic",
  b = 1000,
  R = 999,
  seednum = NULL,
  return_data = FALSE,
  return_detail = FALSE,
  show_boot = FALSE,
  r_print_freq = 100,
  description = TRUE
)
```

Arguments

`data`	A data frame, with at least "Test" and "Disease" variables.
`test`	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.

`disease`	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
`covariate`	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
`saturated_model`	Set as `TRUE` to obtain the original Begg and Greenes' (1983) when all possible interactions are included.
`option`	1 = IPW weight, 2 = W_h weight, described in Arifin (2023), modified weight of Krautenbacher (2017). The default is `option = 2`, which is more stable for small weights (Arifin, 2023).
`ci`	View confidence interval (CI). The default is `FALSE`.
`ci_level`	Set the CI width. The default is 0.95 i.e. 95% CI.
`ci_type`	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI. See `boot.ci` for details.
`b`	The number of repeated samples, b.
`R`	The number of bootstrap samples. Default `R = 999`.
`seednum`	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
`return_data`	Return data for the bootstrapped samples.
`return_detail`	Return accuracy measures for each of the bootstrapped samples.
`show_boot`	Set to `TRUE` to show bootstrap iterations.
`r_print_freq`	Print the current bootstrap sample number at each specified interval. Default `r_print_freq = 100`.
`description`	Print the name of this analysis. The default is `TRUE`. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

boot_data An object of class "boot" from `boot`. Contains Sensitivity, Specificity, PPV, and NPV

boot_ci_data A list of objects of type "bootci" from `boot.ci`. Contains Sensitivity, Specificity, PPV, NPV.

acc_results The accuracy results.

References

1. Arifin, W. N., & Yusof, U. K. (2025). Partial verification bias correction using scaled inverse probability resampling for binary diagnostic tests. PloS One, 20(9), e0321440.
2. Arifin, W. N., & Yusof, U. K. (2022). Partial Verification Bias Correction Using Inverse Probability Bootstrap Sampling for Binary Diagnostic Tests. Diagnostics, 12(11), 2839.
3. Arifin, W. N. (2023). Partial verification bias correction in diagnostic accuracy studies using propensity score-based methods (PhD thesis, Universiti Sains Malaysia). https://erepo.usm.my/handle/123456789/1918

4. Krautenbacher, N., Theis, F. J., & Fuchs, C. (2017). Correcting Classifiers for Sample Selection Bias in Two-Phase Case-Control Studies. Computational and Mathematical Methods in Medicine, 2017, 1–18.
5. Nahorniak, M., Larsen, D. P., Volk, C., & Jordan, C. E. (2015). Using inverse probability bootstrap sampling to eliminate sample induced bias in model based analysis of unequal probability samples. PLoS One, 10(6), e0131765.

Examples

```
# point estimates
acc_sipw(data = cad_pvb, test = "T", disease = "D", b = 100, seednum = 12345)
acc_sipw(data = cad_pvb, test = "T", disease = "D", covariate = "X3",
         b = 100, seednum = 12345)

# with bootstrapped confidence interval
acc_sipw(data = cad_pvb, test = "T", disease = "D", ci = TRUE,
         b = 100, R = 9, seednum = 12345)  # use small b, R for testing
```

`acc_sipwb` *PVB correction by scaled inverse probability weighted balanced resampling (SIPW-B).*

Description

Perform PVB correction by scaled inverse probability weighted balanced resampling. SIPW-B only gives resultsfor Sensitivity and Specificity, for PPV and NPV please use SIPW instead.

Usage

```
acc_sipwb(
  data,
  test,
  disease,
  covariate = NULL,
  saturated_model = FALSE,
  option = 2,
  rel_size = 1,
  ci = FALSE,
  ci_level = 0.95,
  ci_type = "basic",
  b = 1000,
  R = 999,
  seednum = NULL,
  return_data = FALSE,
  return_detail = FALSE,
  show_boot = FALSE,
  r_print_freq = 100,
  description = TRUE
)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
covariate	The name(s) of covariate(s), i.e. other variables associated with either test or disease status. Specify as name vector, e.g. c("X1", "X2") for two or more variables. The variables must be in formats acceptable to GLM.
saturated_model	Set as TRUE to obtain the original Begg and Greenes' (1983) when all possible interactions are included.
option	1 = IPW weight, 2 = W_h weight, described in Arifin (2023), modified weight of Krautenbacher (2017). The default is option = 2, which is more stable for small weights (Arifin, 2023).
rel_size	ratio control:case, D=0:D=1. The default is 1.
ci	View confidence interval (CI). The default is FALSE.
ci_level	Set the CI width. The default is 0.95 i.e. 95% CI.
ci_type	Set confidence interval (CI) type. Acceptable types are "norm", "basic", "perc", and "bca", for bootstrapped CI. See boot.ci for details.
b	The number of repeated samples, b.
R	The number of bootstrap samples. Default R = 999.
seednum	Set the seed number for the bootstrapped CI. The default is not set, so it depends on the user to set it outside or inside the function.
return_data	Return data for the bootstrapped samples.
return_detail	Return accuracy measures for each of the bootstrapped samples.
show_boot	Set to TRUE to show bootstrap iterations.
r_print_freq	Print the current bootstrap sample number at each specified interval. Default r_print_freq = 100.
description	Print the name of this analysis. The default is TRUE. This can be turned off for repeated analysis, for example in bootstrapped results.

Value

A list object containing:

acc_results The accuracy results.

References

1. Arifin, W. N., & Yusof, U. K. (2025). Partial verification bias correction using scaled inverse probability resampling for binary diagnostic tests. PloS One, 20(9), e0321440.
2. Arifin, W. N., & Yusof, U. K. (2022). Partial Verification Bias Correction Using Inverse Probability Bootstrap Sampling for Binary Diagnostic Tests. Diagnostics, 12(11), 2839.

3. Arifin, W. N. (2023). Partial verification bias correction in diagnostic accuracy studies using propensity score-based methods (PhD thesis, Universiti Sains Malaysia). https://erepo.usm.my/handle/123456789/1918
4. Krautenbacher, N., Theis, F. J., & Fuchs, C. (2017). Correcting Classifiers for Sample Selection Bias in Two-Phase Case-Control Studies. Computational and Mathematical Methods in Medicine, 2017, 1–18.
5. Nahorniak, M., Larsen, D. P., Volk, C., & Jordan, C. E. (2015). Using inverse probability bootstrap sampling to eliminate sample induced bias in model based analysis of unequal probability samples. PLoS One, 10(6), e0131765.

Examples

```
# point estimates
acc_sipwb(data = cad_pvb, test = "T", disease = "D", b = 100, seednum = 12345)
acc_sipwb(data = cad_pvb, test = "T", disease = "D", covariate = "X3",
         b = 100, seednum = 12345)

# with bootstrapped confidence interval
acc_sipwb(data = cad_pvb, test = "T", disease = "D", ci = TRUE,
         b = 100, R = 9, seednum = 12345)  # use small b, R for testing
```

---

cad_pvb *SPECT Thallium test data set*

Description

Single-photon-emission computed-tomography (SPECT) thallium is a non-invasive diagnostic test used to diagnose coronary artery disease (CAD). SPECT thallium test was performed on 2688 patients. CAD is diagnosed when stenosis exceeds 50% of the artery, as evaluated by coronary angiography (gold standard). Only 471 patients underwent the coronary angiography for verification of the CAD status. The rest of the patients were unverified (82.5%).

Usage

```
cad_pvb
```

Format

A data frame with 2688 rows and five variables:

T: SPECT thallium test, T: Binary, 1 = Positive, 0 = Negative

D: CAD, D: Binary, 1 = Yes, 0 = No

X1: Gender (covariate), X_1: Binary, 1 = Male, 0 = Female

X2: Stress mode (covariate), X_2: Binary, 1 = Dipyridamole (Medication for stress test when the patient is unable to exercise), 0 = Exercise

X3: Age (covariate), X_3: Binary, 1 = 60 years and above, 0 = Below 60 years

Source

1. Cecil, M. P., Kosinski, A. S., Jones, M. T., Taylor, A., Alazraki, N. P., Pettigrew, R. I., & Weintraub, W. S. (1996). The importance of work-up (verification) bias correction in assessing the accuracy of SPECT thallium-201 testing for the diagnosis of coronary artery disease. Journal of Clinical Epidemiology, 49(7), 735–742.

2. Kosinski, A. S., & Barnhart, H. X. (2003). Accounting for nonignorable verification bias in assessment of diagnostic tests. Biometrics, 59(1), 163–171.

---

diapha_pvb *Diaphanography test data set*

Description

Diaphanography test is a noninvasive method (diagnostic test) of breast examination by transillumination using visible or infrared light to detect the presence of breast cancer. The test was performed on 900 patients. Only 88 patients were verified by breast tissue biopsy for histological examination (gold standard test). The percentage of unverified patients is 90.2%.

Usage

```
diapha_pvb
```

Format

A data frame with 900 rows and three variables:

disease: Breast cancer, $disease$: Binary, 1 = Yes, 0 = No

test: Diaphanography, $test$: Binary, 1 = Positive, 0 = Negative

verified: Verified, $verified$: Binary, 1 = Yes, 0 = No

Source

1. Marshall, V., Williams, D. C., & Smith, K. D. (1984). Diaphanography as a means of detecting breast cancer. Radiology, 150(2), 339–343.

hepatic_pvb *Hepatic scintigraphy test data set*

Description

The data set pertains to hepatic scintigraphy, a diagnostic imaging technique used for detecting liver cancer. The test was performed on 650 patients, where 344 patients were verified by liver pathological examination (gold standard test). The percentage of unverified patients is 47.1%.

Usage

```
hepatic_pvb
```

Format

A data frame with 650 rows and three variables:

disease: Liver cancer, $disease$: Binary, 1 = Yes, 0 = No

test: Hepatic scintigraphy, $test$: Binary, 1 = Positive, 0 = Negative

verified: Verified, $verified$: Binary, 1 = Yes, 0 = No

Source

1. Drum, D. E., & Christacopoulos, J. S. (1972). Hepatic scintigraphy in clinical decision making. Journal of Nuclear Medicine, 13(12), 908–915.

view_table *Test vs Disease/Gold Standard cross-classification table*

Description

View Test vs Disease/Gold Standard cross-classification table.

Usage

```
view_table(data, test, disease, show_unverified = FALSE, show_total = FALSE)
```

Arguments

data	A data frame, with at least "Test" and "Disease" variables.
test	The "Test" variable name, i.e. the test result. The variable must be in binary; positive = 1, negative = 0 format.
disease	The "Disease" variable name, i.e. the true disease status. The variable must be in binary; positive = 1, negative = 0 format.
show_unverified	Optional. Set to TRUE to view observations with unverified disease status. The default is FALSE.
show_total	Optional. Set to TRUE to view total by test result. The default is FALSE.

Value

A cross-classification table.

Examples

```
str(cad_pvb)  # built-in data

view_table(data = cad_pvb, test = "T", disease = "D")  # without unverified observations
view_table(data = cad_pvb, test = "T", disease = "D", show_total = TRUE)
  # also with total observations by test result

view_table(data = cad_pvb, test = "T", disease = "D", show_unverified = TRUE)
  # with unverified observations
view_table(data = cad_pvb, test = "T", disease = "D", show_unverified = TRUE,
           show_total = TRUE)  # also with total observations by test result
```

Index