Package 'MRMCbinary'

May 7, 2026

Title Multi-Reader Multi-Case Analysis of Binary Diagnostic Tests

Version 1.0.6

Description Implements methods for comparing sensitivities and specificities in balanced (or fully crossed) multi-reader multi-case (MRMC) studies with binary diagnostic test results. It implements conditional logistic regression and provides score tests equivalent to Cochran's Q test (which corresponds to McNemar's test when comparing two modalities only). The methodology is based on Lee et al. (2026) <doi:10.1002/sim.70471>.

License MIT + file LICENSE

URL `https://github.com/seungjae2525/MRMCbinary`

BugReports `https://github.com/seungjae2525/MRMCbinary/issues`

Depends R (>= 4.2.0)

Imports DescTools, stats, survival (>= 3.8-1)

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

NeedsCompilation no

Author Seungjae Lee [aut, cre] (ORCID: <`https://orcid.org/0000-0001-5508-8634`>), Woojoo Lee [aut] (ORCID: <`https://orcid.org/0000-0001-7447-7045`>)

Maintainer Seungjae Lee <`seungjae2525@gmail.com`>

Repository CRAN

Date/Publication 2026-03-23 15:00:02 UTC

Contents

MRMCbinary-package 2
MRMCbinary 2
print.MRMCbinary 6
print.SensSpec 8
SensSpec 9
summary.MRMCbinary 11
VanDyke 13

Index 15

`MRMCbinary-package` *MRMCbinary: Multi-reader multi-case analysis of binary diagnostic tests*

Description

R package **MRMCbinary** is a package aimed at comparing the diagnostic performance of different modalities (i.e., sensitivities and specificities) in multi-reader multi-case (MRMC) studies with binary diagnostic test results.

Author(s)

Seungjae Lee `<seungjae2525@gmail.com>` and Woojoo Lee `<lwj221@gmail.com>`

References

Lee, S., Jang, S., and Lee, W. (2026). Evaluating Diagnostic Accuracy of Binary Medical Tests in Multi-Reader Multi-Case Study. *Statistics in Medicine*, 45(6–7), e70471. doi:10.1002/sim.70471

See Also

Useful links:

- `https://github.com/seungjae2525/MRMCbinary`
- Report bugs at `https://github.com/seungjae2525/MRMCbinary/issues`

`MRMCbinary` *Multi-reader multi-case analysis of binary diagnostic tests*

Description

`MRMCbinary()` is the main function of the `MRMCbinary` package and can be used to compare modality effects (as well as reader effects and modality effects together), as measured by sensitivity and specificity, in multi-reader multi-case (MRMC) studies with binary diagnostic test results.

Usage

```
MRMCbinary(
  data,
  Modality,
  Reader,
  Case,
  D,
  Y,
  measure,
```

```
  effect,
  interaction = NULL,
  reference.Modality = NULL,
  reference.Reader = NULL
)
```

Arguments

`data`	A data frame in which contains the modality identifiers (`Modality`), the reader identifiers (`Reader`), the case identifiers (`Case`), the true disease status (`D`), and the binary diagnostic test result (`Y`) from a multi-reader multi-case (MRMC) study.
`Modality`	Variable specifying the modality identifiers.
`Reader`	Variable specifying the reader identifiers.
`Case`	Variable specifying the case identifiers.
`D`	Variable specifying the true disease status, coded as 1 for diseased cases and 0 for non-diseased cases.
`Y`	Variable specifying the binary diagnostic test result, coded as 1 for test-positive results and 0 for test-negative results.
`measure`	Diagnostic accuracy measure to be analyzed (one of `"All"`, `"Sensitivity"`, and `"Specificity"`). `"All"` analyzes both sensitivity and specificity.
`effect`	Effect of interest to be evaluated (one of `"Modality"`, `"Reader"`, and `"Both"`). `"Both"` evaluates modality effects and reader effects together. See **Details**.
`interaction`	Logical indicating whether to include the reader-by-modality interaction effect. Specify this argument only when `effect` is `"Both"`; set `interaction = TRUE` to fit the model with interaction and `interaction = FALSE` to fit the model without interaction. Default: `NULL`. See **Details**.
`reference.Modality`	Reference level for the modality identifier variable.
`reference.Reader`	Reference level for the reader identifier variable.

Details

There are three effects that can be evaluated:

- `effect = "Modality"`: Used when the goal is to evaluate modality effects only. In this case, Cochran's Q test is reported, or McNemar's test when comparing two modalities only. When `effect = "Modality"`, `interaction` must be `NULL`.
- `effect = "Reader"`: Used when the goal is to evaluate reader effects only. In this case, Cochran's Q test is reported, or McNemar's test when comparing two readers only. When `effect = "Reader"`, `interaction` must be `NULL`.
- `effect = "Both"`: Used when the goal is to evaluate modality effects and reader effects together. In this case, `interaction` must be specified as `TRUE` or `FALSE`. Set `interaction = FALSE` to fit the additive model without reader-by-modality interaction, and set `interaction = TRUE` to fit the model including the reader-by-modality interaction effect. When `interaction`

= `TRUE`, Cochran's Q test is reported. When `interaction = FALSE`, Cochran's Q test or McNemar's test is not reported, because the corresponding Cochran's Q test is linked to the model with interaction rather than the additive model.

See Lee et al. (2026) for details.

Value

An object of class `MRMCbinary` containing the following components:

`CLR_sen`	Conditional logistic regression results for sensitivity.
`CLR_LRT_sen`	Likelihood ratio test results from the conditional logistic regression for sensitivity.
`CLR_Score_sen`	Score test results from the conditional logistic regression for sensitivity.
`CLR_Wald_sen`	Wald test results from the conditional logistic regression for sensitivity.
`Q_MN_sen`	Additional nonparametric test results for sensitivity, when available. For `effect = "Modality"`, this is Cochran's Q test, or McNemar's test when comparing two modalities only. For `effect = "Reader"`, this is Cochran's Q test, or McNemar's test when comparing two readers only. For `effect = "Both"` with `interaction = TRUE`, this is Cochran's Q test for reader effects, modality effects, and their interaction together. This component is not reported when `effect = "Both"` and `interaction = FALSE`.
`CLR_spe`	Conditional logistic regression results for specificity.
`CLR_LRT_spe`	Likelihood ratio test results from the conditional logistic regression for specificity.
`CLR_Score_spe`	Score test results from the conditional logistic regression for specificity.
`CLR_Wald_spe`	Wald test results from the conditional logistic regression for specificity.
`Q_MN_spe`	Additional nonparametric test results for specificity, when available. For `effect = "Modality"`, this is Cochran's Q test, or McNemar's test when comparing two modalities only. For `effect = "Reader"`, this is Cochran's Q test, or McNemar's test when comparing two readers only. For `effect = "Both"` with `interaction = TRUE`, this is Cochran's Q test for reader effects, modality effects, and their interaction together. This component is not reported when `effect = "Both"` and `interaction = FALSE`.
`formula`	Formula used in the conditional logistic regression.
`args`	List of arguments used in the `MRMCbinary` function.
`n.modality`	Total number of modalities.
`n.reader`	Total number of readers.
`n.case`	Total number of cases.
`effect`	Effect evaluated in the analysis.
`measure`	Diagnostic accuracy measure analyzed.
`interaction`	Interaction setting used in the analysis. This is `TRUE` or `FALSE` when `effect = "Both"`, and `NULL` otherwise.

`reference.Modality`	Reference level for the modality identifier variable.
`reference.Reader`	Reference level for the reader identifier variable.
`n.diseased`	Number of diseased cases. If `measure = "Specificity"`, this is `NULL`.
`n.nondiseased`	Number of non-diseased cases. If `measure = "Sensitivity"`, this is `NULL`.
`n.pos.diseased`	Number of test-positive cases among diseased cases. If `measure = "Specificity"`, this is `NULL`.
`n.pos.nondiseased`	Number of test-positive cases among non-diseased cases. If `measure = "Sensitivity"`, this is `NULL`.

The results for `MRMCbinary` objects can be printed with `print.MRMCbinary` and summarized with `summary.MRMCbinary`.

Note: When comparing specificities, the reported estimates are based on the false positive fraction (FPF), because the model for non-diseased cases targets logit(FPF). Hence, a positive coefficient implies lower specificity than the reference. For the `Estimate` and confidence interval reported by `print.MRMCbinary`, interpretation on the specificity scale requires reversing the sign of the reported log odds ratio and confidence interval limits. Equivalently, for the odds ratio and confidence interval reported by `summary.MRMCbinary`, if the reported values are OR and (L, U), then the corresponding specificity odds ratio and confidence interval are 1/OR and (1/U, 1/L), respectively.

References

Lee, S., Jang, S., and Lee, W. (2026). Evaluating Diagnostic Accuracy of Binary Medical Tests in Multi-Reader Multi-Case Study. *Statistics in Medicine*, 45(6–7), e70471. doi:10.1002/sim.70471

See Also

`print.MRMCbinary`, `summary.MRMCbinary`

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## See unique readers
unique(VanDyke$reader)

## See unique modalities
unique(VanDyke$treatment)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)

## Example usage of MRMCbinary function:
# When comparing the sensitivities and specificities between modalities
```

```
modality_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                              Case = case, D = truth, Y = Y, measure = "All",
                              effect = "Modality", interaction = NULL,
                              reference.Modality = "1", reference.Reader = NULL)

# When comparing the sensitivities and specificities between readers
reader_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                            Case = case, D = truth, Y = Y, measure = "All",
                            effect = "Reader", interaction = NULL,
                            reference.Modality = NULL, reference.Reader = "1")

# When comparing the sensitivities and specificities
#  between modalities and between readers together
#  not considering interaction between modalities and readers
both_result_wo_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                                 Case = case, D = truth, Y = Y, measure = "All",
                                 effect = "Both", interaction = FALSE,
                                 reference.Modality = "1", reference.Reader = "1")

# When comparing the sensitivities and specificities
#  between modalities and between readers together
#  considering interaction between modalities and readers
both_result_with_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                                   Case = case, D = truth, Y = Y, measure = "All",
                                   effect = "Both", interaction = TRUE,
                                   reference.Modality = "1", reference.Reader = "1")
```

`print.MRMCbinary` *Print method for* `MRMCbinary` *objects*

Description

Prints the results for an object of class `MRMCbinary`.

Usage

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

Arguments

`x`	An object of class `MRMCbinary`.
`...`	Further arguments (currently not used).

Details

Prints the results from the conditional logistic regression (CLR) analysis. In the printed CLR results, `Estimate` denotes the estimated log odds ratio and `SE` denotes its standard error.

When comparing specificities, the reported `Estimate` is based on the false positive fraction (FPF) among non-diseased cases. Therefore, a positive `Estimate` implies lower specificity than the reference. For interpretation on the specificity scale, use the negative of the reported `Estimate`.

Value

No return value, called for side effects.

See Also

`MRMCbinary`, `summary.MRMCbinary`, `print`

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## Extract unique modalities
unique(VanDyke$treatment)

## Extract Unique readers
unique(VanDyke$reader)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)

## Example usage of MRMCbinary function:
# When comparing the sensitivities and specificities between modalities
modality_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                              Case = case, D = truth, Y = Y, measure = "All",
                              effect = "Modality", interaction = NULL,
                              reference.Modality = "1", reference.Reader = NULL)
print(modality_result)

# When comparing the sensitivities and specificities between readers
reader_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                            Case = case, D = truth, Y = Y, measure = "All",
                            effect = "Reader", interaction = NULL,
                            reference.Modality = NULL, reference.Reader = "1")
print(reader_result)

# When comparing the sensitivities and specificities
#  between modalities and between readers together
#  not considering interaction between modalities and readers
both_result_wo_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
```

```
                                 Case = case, D = truth, Y = Y, measure = "All",
                                 effect = "Both", interaction = FALSE,
                                 reference.Modality = "1", reference.Reader = "1")
print(both_result_wo_int)

# When comparing the sensitivities and specificities
#  between modalities and between readers together
#  considering interaction between modalities and readers
both_result_with_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                                   Case = case, D = truth, Y = Y, measure = "All",
                                   effect = "Both", interaction = TRUE,
                                   reference.Modality = "1", reference.Reader = "1")
print(both_result_with_int)
```

`print.SensSpec` *Print method for* `SensSpec` *objects*

Description

Prints the results for an object of class `SensSpec`.

Usage

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

Arguments

`x`	An object of class `SensSpec`.
`...`	Further arguments (currently not used).

Details

Prints the sensitivity and specificity results stored in a `SensSpec` object.

Value

No return value, called for side effects.

See Also

`SensSpec`, `print`

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## Extract unique modalities
unique(VanDyke$treatment)

## Extract Unique readers
unique(VanDyke$reader)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)

## Example usage of SensSpec function:
senspe_result1 <- SensSpec(data = VanDyke, Modality = treatment,
                           Reader = reader, Case = case,
                           D = truth, Y = Y, percentage = FALSE, digits = 3)
print(senspe_result1)

# Report results as percentage points
senspe_result2 <- SensSpec(data = VanDyke, Modality = treatment,
                           Reader = reader, Case = case,
                           D = truth, Y = Y, percentage = TRUE, digits = 1)
print(senspe_result2)
```

`SensSpec` *Calculate sensitivity and specificity*

Description

`SensSpec()` is a function for calculating overall sensitivity and specificity, modality-specific sensitivity and specificity, and reader-specific sensitivity and specificity within each modality.

Usage

```
SensSpec(
  data,
  Modality,
  Reader,
  Case = NULL,
  D,
  Y,
  percentage = FALSE,
  digits = max(1L, getOption("digits") - 3L)
)
```

Arguments

`data`	A data frame containing the modality identifiers (`Modality`), the reader identifiers (`Reader`), the case identifiers (`Case`), the true disease status (`D`), and the binary diagnostic test result (`Y`).
`Modality`	Variable specifying the modality identifiers.
`Reader`	Variable specifying the reader identifiers.
`Case`	Variable specifying the case identifiers. This variable is not directly used in the calculation of sensitivity and specificity in `SensSpec()`. Default: `NULL`.
`D`	Variable specifying the true disease status, coded as 1 for diseased cases and 0 for non-diseased cases.
`Y`	Variable specifying the binary diagnostic test result, coded as 1 for test-positive results and 0 for test-negative results.
`percentage`	Logical indicating whether the results should be reported as percentages rather than decimal proportions. Default: `FALSE`.
`digits`	Number of significant digits used to format the reported results. Default: `max(1L, getOption("digits") - 3L)`.

Value

An object of class `SensSpec` containing the following components:

`Overall Result`	Overall sensitivity and specificity.
`Modality-specific Result`	Modality-specific sensitivity and specificity.
`Reader-specific Modality-specific Result`	Reader- and modality-specific sensitivity and specificity.
`digits`	Number of significant digits used to format the reported results.
`Readers`	Unique reader identifiers in the input data.
`Modalities`	Unique modality identifiers in the input data.

The results for `SensSpec` objects can be printed with `print.SensSpec`.

References

Yerushalmy, J. (1947). Statistical Problems in Assessing Methods of Medical Diagnosis, with Special Reference to X-Ray Techniques. *Public Health Reports (1896-1970)*, 62(40), 1432–1449.

See Also

`print.SensSpec`

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## Extract unique modalities
unique(VanDyke$treatment)

## Extract Unique readers
unique(VanDyke$reader)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)

## Example usage of SensSpec function:
# Report results as decimals
senspe_result1 <- SensSpec(data = VanDyke, Modality = treatment,
                           Reader = reader, Case = case,
                           D = truth, Y = Y, percentage = FALSE, digits = 3)

# Report results as percentage points
senspe_result2 <- SensSpec(data = VanDyke, Modality = treatment,
                           Reader = reader, Case = case,
                           D = truth, Y = Y, percentage = TRUE, digits = 1)
```

`summary.MRMCbinary` *Summary method for* `MRMCbinary` *objects*

Description

Summarizes the results for an object of class `MRMCbinary`.

Usage

```
## S3 method for class 'MRMCbinary'
summary(object, digits = max(1L, getOption("digits") - 3L), ...)
```

Arguments

`object`	An object of class `MRMCbinary`.
`digits`	Number of significant digits used to format the reported results. Default: `max(1L, getOption("digits") - 3L)`.
`...`	Further arguments (currently not used).

Details

Summarizes the results from the conditional logistic regression analysis. In the summarized results, the odds ratio, confidence interval for the odds ratio, and P value are reported.

When comparing specificities, the reported odds ratio and confidence interval are based on the false positive fraction (FPF) among non-diseased cases. Therefore, an odds ratio greater than 1 implies lower specificity than the reference. For interpretation on the specificity scale, if the reported odds ratio and confidence interval are OR and (L, U), then the corresponding specificity odds ratio and confidence interval are 1/OR and (1/U, 1/L), respectively.

Value

No return value, called for side effects.

See Also

`MRMCbinary`, `print.MRMCbinary`, `summary`

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## Extract unique modalities
unique(VanDyke$treatment)

## Extract Unique readers
unique(VanDyke$reader)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)

## Example usage of MRMCbinary function:
# When comparing the sensitivities and specificities between modalities
modality_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                              Case = case, D = truth, Y = Y, measure = "All",
                              effect = "Modality", interaction = NULL,
                              reference.Modality = "1", reference.Reader = NULL)
summary(modality_result, digits = 3)

# When comparing the sensitivities and specificities between readers
reader_result <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                            Case = case, D = truth, Y = Y, measure = "All",
                            effect = "Reader", interaction = NULL,
                            reference.Modality = NULL, reference.Reader = "1")
summary(reader_result, digits = 3)

# When comparing the sensitivities and specificities
#  between modalities and between readers together
```

```
#  not considering interaction between modalities and readers
both_result_wo_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                                 Case = case, D = truth, Y = Y, measure = "All",
                                 effect = "Both", interaction = FALSE,
                                 reference.Modality = "1", reference.Reader = "1")
summary(both_result_wo_int, digits = 3)

# When comparing the sensitivities and specificities
#  between modalities and between readers together
#  considering interaction between modalities and readers
both_result_with_int <- MRMCbinary(data = VanDyke, Modality = treatment, Reader = reader,
                                   Case = case, D = truth, Y = Y, measure = "All",
                                   effect = "Both", interaction = TRUE,
                                   reference.Modality = "1", reference.Reader = "1")
summary(both_result_with_int, digits = 3)
```

`VanDyke` *Multi-reader multi-case dataset*

Description

Example data from a study comparing the relative performance of cinematic presentation of magnetic resonance imaging (CINE MRI) and single spin-echo magnetic resonance imaging (SE MRI) for the detection of thoracic aortic dissection (Van Dyke et al., 1993).

Usage

`VanDyke`

Format

A data frame with 1140 rows and 7 variables:

reader Reader identifiers for the five radiologists.

treatment Treatment identifiers for the two imaging modalities.

case Case identifiers for 114 cases.

case2 Example case identifiers representing cases nested within readers.

case3 Example case identifiers representing cases nested within treatments.

truth Indicator for thoracic aortic dissection (i.e., true disease status): 1 = thoracic aortic dissection present, 0 = thoracic aortic dissection absent

rating Five-point ratings given to case images by the radiologists (i.e., diagnostic test result): 1 = definitely no aortic dissection, 2 = probably no aortic dissection, 3 = unsure about aortic dissection, 4 = probably aortic dissection, or 5 = definitely aortic dissection

Details

This example compares the relative performance of CINE MRI and SE MRI in detecting thoracic aortic dissection. There are 45 patients with aortic dissection and 69 patients without aortic dissection, and all were imaged with both SE MRI and CINE MRI. This dataset is also available in the `MRMCaov` package. See **Source**.

Source

This data are available at `https://perception.lab.uiowa.edu` and `https://github.com/brian-j-smith/MRMCaov/tree/master/data`.

References

Van Dyke, C. W., White, R. D., Obuchowski, N. A., Geisinger, M. A., Lorig, R. J., & Meziane, M. A. (1993). Cine MRI in the diagnosis of thoracic aortic dissection. 79th RSNA Meetings. *Chicago, IL*, 28.

Examples

```
## Load example data
data(VanDyke)

## Return the first parts of an object
head(VanDyke)

## Extract unique modalities
unique(VanDyke$treatment)

## Extract Unique readers
unique(VanDyke$reader)

## Create binary test results (Y_ijk)
VanDyke$Y <- as.numeric(VanDyke$rating >= 3)
```

Index

∗ **datasets**
    `VanDyke`, 13
∗ **methods**
    `MRMCbinary`, 2
    `SensSpec`, 9
∗ **package**
    `MRMCbinary-package`, 2
∗ **print**
    `print.MRMCbinary`, 6
    `print.SensSpec`, 8
∗ **summary**
    `summary.MRMCbinary`, 11

`MRMCbinary`, 2, *7*, *12*
`MRMCbinary-package`, 2

`print`, *7, 8*
`print.MRMCbinary`, *5*, 6, *12*
`print.SensSpec`, 8, *10*

`SensSpec`, *8*, 9
`summary`, *12*
`summary.MRMCbinary`, *5*, *7*, 11

`VanDyke`, 13