

Package: TCPMOR (via r-universe)

August 25, 2024

Title Two Cut-Points with Maximum Odds Ratio

Version 1.0

Description Enables the computation of the 'two cut-points with maximum odds ratio (OR) value method' for data analysis, particularly suited for binary classification tasks. Users can identify optimal cut-points in a continuous variable by maximizing the odds ratio while maintaining an equal risk level, useful for tasks such as medical diagnostics, risk assessment, or predictive modeling.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports SemiPar, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Xianying He [aut], Shuo Yang [aut, cre], Yi Fei [ctb], Jinxin Zhang [aut]

Maintainer Shuo Yang <yangsh223@mail2.sysu.edu.cn>

Date/Publication 2024-03-25 18:50:02 UTC

Repository <https://shuo-yang-sysu.r-universe.dev>

RemoteUrl <https://github.com/cran/TCPMOR>

RemoteRef HEAD

RemoteSha 9e65c0f6a5f9ab9db1da742d5b1c8f91c1cc0adc

Contents

calculateCutoffs	2
createData	3
discretizeAge	3

findCutoffs	4
fitLogisticRegression	5
fitSemiParamModel	6

Index	7
--------------	----------

calculateCutoffs	<i>Calculate data filtering results and two cutoffs for given sensitivity and specificity threshold</i>
------------------	---

Description

Calculate data filtering results and two cutoffs for given sensitivity and specificity threshold

Usage

```
calculateCutoffs(dataC, seThreshold = 0.1, spThreshold = 0.1)
```

Arguments

dataC	Data frame containing columns: se, sp, age, y0, OR, y
seThreshold	Sensitivity threshold
spThreshold	Specificity threshold

Value

A list with two elements: filteredData(the filtered dataset) and cutoffs(the calculated two cutoffs)

Examples

```
# Generate simulated data dataC
dataC <- createData(200)

# Fit the semi-parametric model
spm.fit <- fitSemiParamModel(dataC)

# Find two cut-off points
dataC <- findCutoffs(spm.fit, dataC)

# Output the two cut-off points after limiting sensitivity se and specificity sp
result <- calculateCutoffs(dataC)
cutoffs <- result$cutoffs
dataC2 <- result$filteredData
print(cutoffs)
```

createData	<i>Generate simulation data</i>
------------	---------------------------------

Description

Generate simulation data

Usage

```
createData(n, seed = 123589)
```

Arguments

n	The total number of observations to generate
seed	Randomized seeds for ensuring reproducible results

Value

A data frame containing age and a binary outcome variable

Examples

```
dataC <- createData(200)
```

discretizeAge	<i>Discretize the age variable according to the two cut-off points</i>
---------------	--

Description

Discretize the age variable according to the two cut-off points

Usage

```
discretizeAge(data, cutoffs)
```

Arguments

data	Data frame with column: age
cutoffs	The cut-off points of the age range

Value

A modified data frame with a new column: age_p

Examples

```
# Generate simulated data dataC
dataC <- createData(200)

# Fit the semi-parametric model
spm.fit <- fitSemiParamModel(dataC)

# Find two cut-off points
dataC <- findCutoffs(spm.fit, dataC)

# Calculate the two cut-off points after limiting sensitivity se and specificity sp
result <- calculateCutoffs(dataC)
cutoffs <- result$cutoffs
dataC2 <- result$filteredData

# Discretize age variable based on the two cutoffs
dataC2 <- discretizeAge(dataC2, cutoffs)
```

findCutoffs	<i>Finding the two cut-off points</i>
-------------	---------------------------------------

Description

Finding the two cut-off points

Usage

```
findCutoffs(spm.fit, dataC)
```

Arguments

spm.fit	Fitted semi-parametric model object
dataC	Data frame containing age and binary outcome variables

Value

Data frame containing age, fitted lnOR, OR, se, sp, sse, and ssp

Examples

```
# Generate simulated data dataC
dataC <- createData(200)

# Fit the semi-parametric model
spm.fit <- fitSemiParamModel(dataC)

# Find two cut-off points
dataC <- findCutoffs(spm.fit, dataC)
```

fitLogisticRegression *Fit a logistic regression model and return the OR and 95% confidence interval*

Description

Fit a logistic regression model and return the OR and 95% confidence interval

Usage

```
fitLogisticRegression(data)
```

Arguments

data A data frame with columns: y, age_p

Value

A matrix of OR and 95% confidence intervals

Examples

```
# Generate simulated data dataC
dataC <- createData(200)

# Fit the semi-parametric model
spm.fit <- fitSemiParamModel(dataC)

# Find two cut-off points
dataC <- findCutoffs(spm.fit, dataC)

# Calculate the two cut-off points after limiting sensitivity se and specificity sp
result <- calculateCutoffs(dataC)
cutoffs <- result$cutoffs
dataC2 <- result$filteredData

# Discretize age variable based on the two cutoffs
dataC2 <- discretizeAge(dataC2, cutoffs)

# Fitting logistic regression models and obtaining OR values and 95% confidence intervals
OR_Results <- fitLogisticRegression(dataC2)
print(round(OR_Results, 3))
```

fitSemiParamModel	<i>Fit the data using a semi-parametric model to explore the nonlinear dose-response relationship between the independent variable and lnOR</i>
-------------------	---

Description

Fit the data using a semi-parametric model to explore the nonlinear dose-response relationship between the independent variable and lnOR

Usage

```
fitSemiParamModel(dataC)
```

Arguments

dataC Data frame containing age and binary outcome variables

Value

Fitted semi-parametric model object

Examples

```
# Generate simulated data dataC
dataC <- createData(200)

# Plot the nonlinear dose-response relationship between the independent variable and lnOR
spm.fit <- fitSemiParamModel(dataC)
plot(spm.fit, ylab = "lnOR", xlab = "age", shade = FALSE)
summary(spm.fit)
```

Index

calculateCutoffs, 2
createData, 3

discretizeAge, 3

findCutoffs, 4
fitLogisticRegression, 5
fitSemiParamModel, 6