Package ‘rangecondprob’

March 2, 2026

Title Tools for Conditional Probability

Version 0.4.1

Description Streamline the calculation of conditional probabilities for various numeric ranges in an R DataFrame. It automates the need to convert numerical data into categorical data for conditional probability calculation, making it ideal for quick and preliminary data analysis.

License GPL-3

Depends R (>= 3.0.0), dplyr

Encoding UTF-8

RoxygenNote 7.3.3

URL `https://github.com/kimmanlui/calc_cond_prob`

BugReports `https://github.com/kimmanlui/calc_cond_prob/issues`

NeedsCompilation no

Author Kim Man Lui [aut, cre]

Maintainer Kim Man Lui <`cskmlui@gmail.com`>

Repository CRAN

Date/Publication 2026-03-02 22:00:02 UTC

Contents

`calc_cond_prob` *Calculate Conditional Probability*

Description

Calculate Conditional Probability

Usage

```
calc_cond_prob(
  clean_data,
  formula_string = NULL,
  range_list,
  cond_evaluation = NULL,
  col_name_list = NULL,
  verbose = FALSE
)
```

Arguments

`clean_data` A data.frame containing input data for analysis. It must include all variables referenced in ‘formula_string‘ and ‘col_name_list‘.

`formula_string` An optional formula string in the format 'y ~ x1 + x2 + ...'. This specifies the relationship between dependent and independent variables. If provided, it determines the conditional evaluation and the list of column names.

`range_list` A list of ranges or boundaries corresponding to each column in ‘col_name_list‘. This parameter is mandatory and must contain appropriate range definitions.

`cond_evaluation`
An optional string for the conditional evaluation expression. If not provided, it defaults to the left-hand side of ‘formula_string‘.

`col_name_list` An optional list of column names for the analysis. If not specified, it is derived from ‘formula_string‘.

`verbose` A boolean indicating more message. When set to TRUE, additional output will be printed to help trace the computation steps.

Value

A list containing the results of the conditional probability calculation, the good chance evaluation, and the adjusted range list.

Examples

```
## Prepare some sample data
df<-data.frame(exam_lang_score=c(80,88,85,82,34,34),age=c(6,7,8,6,7,8),height =c(5,6,6,7,5,7))
## Find P(exam_lang_score >= 80 ~ age ) where age is divided into 3 groups.
calc_cond_prob(df, "exam_lang_score >= 80  ~ age ",  range_list=list(3))
```

```
## Find P(exam_lang_score >= 80 ~ age  )
## where age is divided into two groups as (5, 6.5) and (6.5 , 10)
calc_cond_prob(df, "exam_lang_score >= 80  ~ age ", range_list=list( list(c(5,6.5), c(6.5,10) )))
## the above is the same as below
calc_cond_prob(df, "exam_lang_score >= 80  ~ age ",  range_list=list( c(5,6.5,10) ))
```

goodchance *Filter Data Based on Odds*

Description

Filter Data Based on Odds

Usage

```
goodchance(df, col_name = "odd", upper = 0.75, lower = 0.25)
```

Arguments

`df`	A data.frame containing the data to be evaluated. The DataFrame must include the column specified by ‘col_name‘ containing odds values.
`col_name`	A string representing the name of the column in the DataFrame that contains the odds values. The default is ’odd’, which means the function will look for an ’odd’ column in the DataFrame.
`upper`	A numeric threshold for the upper bound of the odds. Rows with an ’odd’ value greater than or equal to this threshold will be included in the output. The default value is 0.75.
`lower`	A numeric threshold for the lower bound of the odds. Rows with an ’odd’ value less than or equal to this threshold and greater than 0 will also be included in the output. The default value is 0.25.

Value

A filtered data.frame containing only the rows where the odds meet the specified conditions (either above the upper threshold or below the lower threshold and greater than 0).

Examples

```
## Prepare some sample data
df<-data.frame(exam_lang_score=c(80,88,85,82,34,34),age=c(6,7,8,6,7,8),height =c(5,6,6,7,5,7))
## Find P(exam_lang_score >= 80 ~ age + height)
## where age is divided into 3 groups and height into 4 groups.
res=calc_cond_prob(df, "exam_lang_score >= 80 ~ age + height", range_list=list( 3,4))
## Use the results to calculate the conditional probability for
## the power set of 'age' and 'height'.
summary_result_list=shortSummary(res[[1]], "age + height ", combination=1)
## Extract the results of the high and low odds obtained from
## executing the cal_cond_prob() function.
lapply(summary_result_list, goodchance, upper=0.7, lower=0.25)
```

shortSummary *Generate Short Summary of Grouped Data after the Output by calc_cond_prob*

Description

Generate Short Summary of Grouped Data after the Output by calc_cond_prob

Usage

```
shortSummary(df, coln = "Weekday , wkhwk.c.bp", combination = 1)
```

Arguments

df	A data.frame containing the raw data to be summarized. The DataFrame must include the columns specified in the ‘coln‘ parameter for grouping.
coln	A comma-separated string of column names to be used for grouping in the summary. Whitespace will be trimmed from the specified column names. For example: "Weekday, wkhwk.c.bp".
combination	An integer indicating how to handle column combinations for the summary. When set to 1, the function generates all possible combinations of the columns specified in ‘coln‘. For any other value, only the specified columns will be used.

Value

A list of data.frames containing the summarized results for each group based on the specified columns. Each data.frame will include the specified grouping columns, the sum of hits (‘hit_sum‘), the sum of totals (‘total_sum‘), and the calculated odds (‘odd‘).

A list.

Examples

```
## Prepare some sample data
df<-data.frame(exam_lang_score=c(80,88,85,82,34,34),age=c(6,7,8,6,7,8),height =c(5,6,6,7,5,7))
## Find P(exam_lang_score >= 80 ~ age + height)
## where age is divided into 3 groups and height into 4 groups.
res=calc_cond_prob(df, "exam_lang_score >= 80 ~ age + height", range_list=list( 3,4))
## Use the results to calculate the conditional probability for
## the power set of 'age' and 'height'.
shortSummary(res[[1]], "age + height ", combination=1)
```

Index