

Package ‘CategoricalDataAnalysis’

December 4, 2017

Type Package

Title Categorical Data Analysis

Version 1.0

Date 2017-11-30

Author Maham Niaz, Saqib Ali

Maintainer Maham Niaz <maham.niaz@sjsu.edu>

Description This package is used for analyzing two categorical variables.

License GPL (>= 2)

Exports count_mat, plotlocalor, chisq.indep, odds.ratios, catbarchart,
continuous2categorical

Imports ggplot2, gridExtra

R topics documented:

CategoricalDataAnalysis-package	2
catbarchart	2
chisq.indep	3
continuous2categorical	5
count_mat	6
crabs	7
crabs2	8
odds.ratios	9
plotlocalor	10
Index	12

CategoricalDataAnalysis-package
Categorical Data Analysis

Description

This package is used for analyzing two categorical variables.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Maham Niaz, Saqib Ali

Maintainetextlive-fonts-extraner: Maham Niaz <maham.niaz@sjsu.edu>

Examples

```
data("crabs2")  
catbarchart(continous2categorical(crabs2))
```

catbarchart *Plot Barchart for Categorical Data*

Description

This function plot a Barchart for Categorical Data. The Dataframe must have Categorical Data. The Last Column of the Data Frame must be the Response Variable

Usage

```
catbarchart(x)
```

Arguments

x A Dataframe with Categorical Data. Last Column is the Response Variable

Author(s)

Saqib Ali

Examples

catbarchart create Barchart of Categorical Data. The last colmn of the Dataset should be the Response Variable. All

```
data("crabs2")
head(crabs2)
catbarchart(continuous2categorical(crabs2))

## The function is currently defined as
function (x)
{
  xcolumnnames <- colnames(x)
  responsecol <- ncol(x)
  plot_hist <- function(column, data, response) ggplot(data,
    aes(x = get(column), ..count..) + geom_bar(aes(fill = get(response)),
    position = "dodge") + xlab(column) + scale_fill_discrete(name = response)
  myplots <- lapply(colnames(x), plot_hist, data = x, response = xcolumnnames[responsecol])
  myplots <- myplots[-length(myplots)]
  grid.arrange(grobs = myplots, ncol = 1)
}
```

chisq.indep

Testing for independence between two categorical variable

Description

This function takes in contingency matrix and tests for Chi Squared Independence. The function returns the two test statistics. X^2 and G^2 , which is Pearson test statistic and Likelihood Ratio test statistic respectively.

Usage

```
chisq.indep(m, level = 0.05, digits = 4, print = TRUE)
```

Arguments

m	m is an at least two by two matrix or the contingency matrix. Preferably with rows corresponding to explanatory variable and coulmsns corresponding to response variable.
level	level is the significance level of the test. The null hypothesis is rejected if the p-value is less than a predetermined level, alpha. alpha is called the significance level, and is the probability of rejecting the null hypothesis given that it is true (a type I error). The default value is set to 0.05.
digits	integer indicating the number of decimal places or significant digits to be used. The default is set to 4.

print Default is set to TRUE. If print is set to TRUE the output of the test gets printed. If you do not want to see the output, set it equal to FALSE.

Value

If print is set to TRUE returns the value of level, degree of freedom, critical value rounded to the nearest digit, value of pearson statistic and value of likelihood ratio test statistic

Author(s)

Maham Niaz

Examples

```
#attach dataset crabs
data("crabs2")
#create a contingency matrix for crabs color and satellite
m = table(crabs2$color, crabs2$satellite)
# returns chi squared test of independence for the two variable, color of the crab and satellite status, which is ei
chisq.indep(m)

## The function is currently defined as
function (m, level = 0.05, digits = 4, print = TRUE)
{
  r.sum <- rowSums(m)
  c.sum <- colSums(m)
  n <- sum(m)
  exp.ct <- outer(r.sum, c.sum, "*")/n
  res <- m - exp.ct
  p.res <- res/sqrt(exp.ct)
  X.sq <- sum(p.res^2)
  G.sq <- 2 * sum(m * (log(m) - log(exp.ct)))
  df <- (nrow(m) - 1) * (ncol(m) - 1)
  c.val <- qchisq(level, df = df, lower.tail = FALSE)
  est.se <- sqrt(exp.ct * outer((1 - r.sum/n), (1 - c.sum/n),
                                "*"))
  s.res <- res/est.se
  if (print) {
    cat("Chi-squared test of independence\n")
    cat(" Level = ", level, ", df = ", df, ", critical value = ",
        round(c.val, digits), "\n", sep = "")
    cat(" X-squared = ", round(X.sq, digits), "\n", sep = "")
    cat(" G-squared = ", round(G.sq, digits), sep = "")
    if(X.sq > c.val | G.sq > c.val){
      cat("\n", sep = "", "The test statistic value is greater than critical value. We reject the null hypothesis and c
    } else {
      cat("\n", sep = "", "The test statistic value is less than critical value. We fail to reject the null hypothesis a
    }
  }
}
```

```

    }
}

```

continuous2categorical *continuous2categorical function.*

Description

continuous2categorical function. This function takes a data frame of continuous variables and converts to a data frame of categorical variables. The last variable is the response variable.

Usage

```
continuous2categorical(x)
```

Arguments

x	A dataframe with Continuous variables for Factors. Last column is the Response Variable
---	---

Value

A Dataframe with the Categorical variables. Last column is the Response variable

Examples

```

data("crabs2")
continuous2categorical(crabs2)

## The function is currently defined as
function (x)
{
  numberoffactors <- ncol(x) - 1
  out <- data.frame(0, matrix(nrow = nrow(x), ncol = 1))
  for (i in 1:numberoffactors) {
    labs <- c("low", "low-medium", "medium", "medium-high",
              "high")
    vartemp <- cut(x[, i], breaks = 5, labels = labs)
    out[i] <- vartemp
  }
  i <- i + 1
  out[i] <- x[i]
  colnames(out) <- colnames(x)
  return(data.frame(out))
}

```

count_mat

*creating contingency matrix for categorical data analysis***Description**

Takes a data frame of at least two observations of two categorical variables and returns a contingency table of the data

Usage

```
count_mat(df)
```

Arguments

df df is a data frame with dimentions nx2 of two categorical variabllles.

Value

m a contingency matrix of numerical values with dimension kxn where k is the number of categories in the first variable and n is the number of categories in the second variable

Note

works similar to the table() function

Author(s)

Maham Niaz

Examples

```
#create vector 1 with three levels
a = c("A","A","B","A", "B","B","C","A","C","B")
#create vector 2 with 4 levels
b = c(1,2,1,4,1,2,2,3,4,3)
# create dataframe with a and b vectors as columns
df = cbind(a,b)
#return count matrix
m = count_mat(df)
m

## The function is currently defined as
function (df)
{
  df_dim <- dim(df)
  if (df_dim[2] == 2 && length(df_dim) == 2) {
```

```

factor_df1 <- as.factor(df[, 1])
factor_df2 <- as.factor(df[, 2])
lev_col1 = levels(factor_df1)
lev_col2 = levels(factor_df2)
len_col1 = length(lev_col1)
len_col2 = length(lev_col2)
val = 1
for (i in lev_col1) {
  for (j in lev_col2) {
    val = c(val, length(which(df[, 1] == i & df[,
      2] == j)))
  }
}
out = matrix(val[-1], byrow = TRUE, nrow = length(lev_col1),
  dimnames = list(lev_col1, lev_col2))
}
else (out = "check dimension")
return(out)
}

```

crabs	<i>Horseshoe crabs data on characteristics of female crabs. The data includes color spine width weight and the number of satellites attracted by the male and female pair</i>
-------	---

Description

contains the data analyzed by Brockmann (1996) and is discussed extensively in Agresti (2002). This is a space-delimited text file in which the variable names appear in the first row. Background

Usage

```
data("crabs")
```

Format

A data frame with 174 observations on the following 5 variables.

V1 a factor with levels 2 3 4 5 color

V2 a factor with levels 1 2 3 spine

V3 a factor with levels 21.0 22.0 22.5 22.9 23.0 23.1 23.2 23.4 23.5 23.7 23.8 23.9 24.0 24.1 24.2 24.3 24.5 24.7 24.8 24.9 25.0 25.1 25.2 25.3 25.4 25.5 25.6 25.7 25.8 25.9 26.0 26.1 26.2 26.3 26.5 26.7 26.8 27.0 27.1 27.2 27.3 27.4 27.5 27.6 27.7 27.8 27.9 28.0 28.2 28.3 28.4 28.5 28.7 28.9 29.0 29.3 29.5 29.7 29.8 30.0 30.2 30.3 30.5 31.7 31.9 33.5 width

V4 a factor with levels 0 1 10 11 12 14 15 2 3 4 5 6 7 8 9 num.satellites

V5 a factor with levels 1200 1300 1400 1475 1550 1600 1650 1700 1800 1850 1900 1950 1967 2000 2025 2050 2100 2150 2175 2200 2225 2250 2275 2300 2350 2400 2450 2500 2550 2600 2625 2650 2700 2750 2800 2850 2867 2900 2925 2950 3000 3025 3050 3100 3150 3200 3225 3250 3275 3300 3325 3500 3600 3725 3850 5200 weight

Source

<http://www.math.montana.edu/shancock/courses/stat539/data/horseshoe.txt>

Examples

```
data(crabs)
str(crabs) #gives the summary of the dataset ;
plot(crabs)
```

crabs2	<i>contains the data analyzed by Brockmann (1996) and is discussed extensively in Agresti (2002). This is a dataframe</i>
--------	---

Description

contains the data analyzed by Brockmann (1996) and is discussed extensively in Agresti (2002).
This is a dataframe

Usage

```
data("crabs2")
```

Format

A data frame with 173 observations on the following 5 variables.

color a numeric vector
spine a numeric vector
width a numeric vector
weight a numeric vector
satellite a logical vector

Examples

```
data(crabs2)
head(crabs2)
str(crabs2)
## maybe str(crabs2) ; plot(crabs2) ...
```

odds.ratios	<i>creating a table with local or global odds ratios</i>
-------------	--

Description

This function takes in a contingency table and returns local or global odds ratios for all the subtables formed from the table. The function gives out the odds ratios in the form of $n-1$ by $m-1$ matrix.

Usage

```
odds.ratios(m, type = "local")
```

Arguments

<code>m</code>	The two dimensional contingency table for which all the local and global odds ratios are required
<code>type</code>	the type of odds ratios required. This argument can take values local or global only. The default is set to local. Note that global odds ratios make sense for ordinal data for both variables.

Value

<code>result</code>	a matrix of odds ratios is returned. The dimensions of the matrix are $n-1$ by $k-1$ where n and k are the number of rows and columns of contingency table <code>m</code>
---------------------	---

Author(s)

Maham Niaz

Examples

```
#attaching dataset crabs2
data("crabs2")
# create contingency matrix for variable spine and satellite
m = table(crabs2$spine, crabs2$satellite)
or1 = odds.ratios(m, "global")
or1 #gives matrix for global odds ratios
or2 = odds.ratios(m)
or2 #gives matrix for local odds ratios

## The function is currently defined as
function (m, type = "local")
{
  nr <- nrow(m)
  if (nr < 2)
    stop("number of rows is less than two")
  nc <- ncol(m)
  if (nc < 2)
```

```

      stop("number of columns is less than two")
if (length(type) > 1)
  stop("only one type is allowed")
opts <- c("local", "global")
type <- pmatch(type, opts)
if (is.na(type))
  stop("only \"local\" or \"global\" allowed for type")
result <- matrix(NA, nrow = nr - 1, ncol = nc - 1)
if (type == 1)
  for (i in 1:(nr - 1)) for (j in 1:(nc - 1)) result[i,
    j] <- m[i, j] * m[i + 1, j + 1]/(m[i, j + 1] * m[i +
    1, j])
if (type == 2)
  for (i in 1:(nr - 1)) for (j in 1:(nc - 1)) {
    num <- as.numeric(sum(m[1:i, 1:j])) * as.numeric(sum(m[(i +
    1):nr, (j + 1):nc]))
    den <- as.numeric(sum(m[1:i, (j + 1):nc])) * as.numeric(sum(m[(i +
    1):nr, 1:j]))
    result[i, j] <- num/den
  }
result
}

```

plotlocalor

plotting fourfold plots for odds ratios

Description

This function takes in a contingency table and returns $(k-1)(l-1)$ fourfold plots for odds ratio of all 2×2 subtables in the matrix m with dimensions $k \times l$

Usage

```
plotlocalor(m, col = c("azure4", "aquamarine4"))
```

Arguments

<code>m</code>	a two dimensional contingency matrix
<code>col</code>	The color of the four-fold plots. The default is azure4 and aquamarine4. The second color corresponds to the variable with higher odds of success.

Author(s)

Maham Niaz

Examples

```

#create 2x2 matrix
m = matrix(c(1,5,13,6), nrow=2)
plotlocalor(m) # returns a single plot shpwng descriptive summary of odds ratio

#create 4x4 matrix
m = matrix(c(1,5,13,6,3,5,14,16,36,45,4,6,5,8,9,56), nrow = 4)
plotlocalor(m) # returns 3x3 plots for the odds ratios of 2x2 subtables in the m matrix

#local odds ratios for crabs data
#load data
data("crabs2")
#subsetting two columns
data <- crabs2[,c(2,5)] #using color and satelite information of horseshoe crab
#create matrix
m = table(data)
#plot local odds ratios
plotlocalor(m)

## The function is currently defined as
function (m, col = c("azure4", "aquamarine4"))
{
  nr <- nrow(m)
  if (nr < 2)
    stop("number of rows is less than two")
  nc <- ncol(m)
  if (nc < 2)
    stop("number of columns is less than two")
  par(mfrow = c(nr - 1, nc - 1))
  for (i in 1:(nr - 1)) for (j in 1:(nc - 1)) {
    fourfoldplot(m[i:(i + 1), j:(j + 1)], color = col)
  }
}

```

Index

*Topic \textasciitildekw1

- catbarchart, [2](#)
- chisq.indep, [3](#)
- continous2categorical, [5](#)
- count_mat, [6](#)
- odds.ratios, [9](#)
- plotlocalor, [10](#)

*Topic \textasciitildekw2

- catbarchart, [2](#)
- chisq.indep, [3](#)
- continous2categorical, [5](#)
- count_mat, [6](#)
- odds.ratios, [9](#)
- plotlocalor, [10](#)

*Topic datasets

- crabs, [7](#)
- crabs2, [8](#)

*Topic package

- CategoricalDataAnalysis-package, [2](#)

catbarchart, [2](#)

CategoricalDataAnalysis
(CategoricalDataAnalysis-package),
[2](#)

CategoricalDataAnalysis-package, [2](#)

chisq.indep, [3](#)

continous2categorical, [5](#)

count_mat, [6](#)

crabs, [7](#)

crabs2, [8](#)

odds.ratios, [9](#)

plotlocalor, [10](#)