

Package ‘CTP’

March 15, 2021

Type Package

Title Closed Testing Procedure (CTP)

Description This is a package for constructing hypothesis trees for treatment comparisons based on the closure principle and analysing the corresponding Closed Testing Procedures (CTP) using adjusted p-values. For reference, see Marcus, R., Peritz, E, and Gabriel, K.R. (1976) <doi:10.2307/2335748> and Bauer, P (1991) <doi:10.1002/sim.4780100609>.

Version 3.0

Depends clinfun, diagram, stats, emmeans

Imports dplyr, magrittr, MASS, survival, shape, Gmisc, grid

Suggests knitr

License GPL (>= 2)

LazyLoad yes

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Paul Jordan [aut, cre],
Matthias Herpers [aut]

Maintainer Paul Jordan <paul.jordan@roche.com>

Repository CRAN

Date/Publication 2021-03-15 09:10:05 UTC

R topics documented:

Adjust_p	2
Adjust_raw	2
AnalyseCTP	4
colorectal	6
CTP	6

CTPcompare	7
getCTPparms	7
getResponse	8
glucose	9
IntersectHypotheses	10
pasi	11
summary.ctp	11
TestingSet	13

Index	14
--------------	-----------

Adjust_p	<i>Adjusting p-values, auxiliary function</i>
----------	---

Description

Adjusting p-values, auxiliary function

Usage

Adjust_p(ctp.struc, ctp.pval)

Arguments

ctp.struc	Object of class ctp.str.
ctp.pval	Data frame with hypothesis names and unadjusted p-values. Output from CTPcompare.

Value

Adjusted p-values

Adjust_raw	<i>Adjusting raw p-values of a CTP</i>
------------	--

Description

Function that adjusts the raw p-values of the elementary hypotheses of a closed testing procedure. The raw p-values are adjusted according to the closure principle. The adjusted p-value is calculated as the maximum of the raw p-value from the current hypothesis in question and the raw p-values from all subsequent hypotheses that contain the current hypothesis.

Usage

```
Adjust_raw(
  ctp.struc,
  p.value,
  dataset.name = NULL,
  factor.name = NULL,
  factor.levels = NULL,
  model = NULL,
  test.name = NULL
)
```

Arguments

<code>ctp.struc</code>	Object generated by IntersectHypotheses
<code>p.value</code>	Vector of raw p-values in the order of the hypotheses created by summary.ctp.str
<code>dataset.name</code>	Character string naming the analysis dataset (optional - only for documentation purposes).
<code>factor.name</code>	Character string naming the factor whose levels are compared (optional - only for documentation purposes).
<code>factor.levels</code>	Vector of type "character" containing the levels of the treatment factor (optional - only for documentation purposes).
<code>model</code>	Model used in the analysis (optional - only for documentation purposes).
<code>test.name</code>	Character string naming the statistical test applied.

Value

An object of `oldClass = "ctp"` to be used for summarizing and plotting the results.

See Also

[IntersectHypotheses](#), [AnalyseCTP](#), [Display](#), [summary.ctp](#)

Examples

```
Pairwise <- IntersectHypotheses(list(c(1,2), c(1,3), c(1,4), c(2,3), c(2,4), c(3,4)))
Display(Pairwise)
summary(Pairwise)

# the vector of p-values calculated by another software

p.val <- c(
  0.4374,
  0.6485,
  0.4103,
  0.2203,
  0.1302,
  0.6725,
```

```

0.4704,
0.3173,
0.6762,
0.7112,
0.2866,
0.3362,
0.2871,
0.4633)

result <- Adjust_raw(ctp.struc=Pairwise, p.value=p.val)

## details may be documented

result <- Adjust_raw(Pairwise, p.value=p.val
,dataset.name="my Data", factor.name="Factor"
,factor.levels=c("A","B","C","D"), model=y~Factor
,test.name="my Test")

summary(result)
Display(result)

```

AnalyseCTP

Analysing a Closed Testing Procedure

Description

Calculation of p-values of a closed testing procedure (CTP). The function returns an object of oldClass "ctp"; `summary()` and `Display()` can be applied.

Usage

```
AnalyseCTP(ctp.struc, model, data, factor.name = NULL, test.name = "F", ...)
```

Arguments

<code>ctp.struc</code>	Object generated by the function IntersectHypotheses (structure of CTP)
<code>model</code>	model of the form <code>response~treatment</code> . If <code>testname="F"</code> , the model can be extended by covariates and other factors. In the case of a Logrank test the response must be imputed as <code>Surv(time, status)</code> .
<code>data</code>	Dataframe, missing values in the response or treatment variable are not allowed!
<code>factor.name</code>	Character string naming the factor whose levels are compared (treatment factor). By default the first variable of the right-hand side of the model formula is used.
<code>test.name</code>	One of the following strings <ul style="list-style-type: none"> "F" - F-Test (ANOVA from linear model, default) "glm" - generalised linear model

- "kruskal" -Kruskal-Wallis-Test
- "chisq" - Chi square test
- "prob" - Fisher's exact test for total number of observations <200 else Chi square test
- "lgrank" - Logrank-test
- "jonckheere" - Jonckheere-Terpstra test of ordered alternatives
- "glm" - generalised linear model, using function glm from stats.

... Additional arguments for the chosen test.

Details

The hypothesis tree of the closed testing procedure must be created using [IntersectHypotheses](#). For more details on the theory and the implementation as well for many examples, see the vignettes.

Value

An object of old class(ctp), consisting of a list with:

- CTPparms: List with objects describing the CTP setup.
- pvalues: Dataframe with all tested hypotheses, raw and adjusted p-values.

Note

This procedure is constructed for testing differences and two-sided hypotheses, but not for equivalence tests. It is further based on independent samples from the population involved (i.e. on parallel group designs, but not on cross-over designs).

See Also

[IntersectHypotheses](#), [Display](#), [summary.ctp.str](#), [summary.ctp](#), [Adjust_raw](#)

Examples

```
data(pasi)
three.to.first <- IntersectHypotheses(list(1:2,c(1,3),c(1,4)))
Display(three.to.first)
pasi.ctp.F1 <- AnalyseCTP(three.to.first,pasi.ch~dose,pasi)
summary(pasi.ctp.F1)
Display(pasi.ctp.F1)
```

`colorectal`*Data from a Study in Colorectal Cancer*

Description

A dataframe containing the responders to the treatment with two doses of an experimental drug or a standard treatment.

Usage

```
data(colorectal)
```

Format

A data frame with 104 observations and 2 variables:

dose a factor with levels control low high

responder a factor with levels no yes

`CTP`*CTP, a package for closed testing procedures*

Description

Library of functions to design and analyse closed testing procedures for the comparison of population parameters based on independent samples.

Details

Library of functions to design and analyse closed testing procedures for treatment comparisons. The comparison of the population parameters are based on independent samples. The function [IntersectHypotheses](#) creates the hypothesis tree (closure set) of a closed testing procedure. All possible intersecting hypotheses derived from the list of elementary hypotheses (i.e. the hypotheses to be tested) are produced. The resultant hypothesis tree will consist of the elementary hypotheses as well as all subsequent intersected hypotheses. The procedure ends when one hypothesis (the global hypothesis) remains. In this way, for each elementary hypothesis all hypotheses implying it can be found.

The analysis is performed using [AnalyseCTP](#). First the raw p-values are computed for all hypotheses of the hypothesis tree, then the p-values of the elementary hypotheses are adjusted according to the closure principle i.e. the adjusted p-value is calculated as the maximum of the raw p-value of the hypothesis in question and of the raw p-values of all hypotheses implying it.

Instead of applying [AnalyseCTP](#), the raw p-values can be computed using any other available software. For this purpose, the functions [Adjust_raw](#) is provided.

and [summary.ctp.str](#) generates a data frame comprising all the hypothesis of the hypothesis tree. The function [Adjust_raw](#) calculates the p-values for the given hypothesis tree from the raw p-values provided in the same order as the hypotheses occur in the data frame created by [summary.ctp.str](#).

The results are presented using the generic functions `summary` and `Display`.

Author(s)

J. Bock & P.Jordan

CTPcompare	<i>Comparisons of treatment levels</i>
------------	--

Description

Performs all required comparisons of treatment levels

Usage

```
CTPcompare(CTPparms, ...)
```

Arguments

CTPparms	Object of class CTPparms (i.e. generated by the function <code>getCTPparms</code>)
...	Additional arguments for the chosen test

Value

A data.frame with the hypotheses and the unadjusted p-values

getCTPparms	<i>Auxiliary function</i>
-------------	---------------------------

Description

Extracting CTP - parameters

Usage

```

getResponse(
  ctp.struc,
  model,
  dataset,
  factor.name = NULL,
  test.name = "F",
  ...
)

```

Arguments

ctp.struc	Object of class <code>ctp.str</code> .
model	Formula of the form <code>response~treatment</code> .
dataset	Dataframe.
factor.name	Character string naming the factor whose levels are compared (treatment factor). By default the first variable of the right-hand side of the model formula is used.
test.name	One of the following strings <ul style="list-style-type: none"> • "F" - F-Test (ANOVA, default) • "kruskal" -Kruskal-Wallis-Test • "chisq" - Chi square test • "prob" - Fisher's exact test for total number of observations <200 else Chi square test • "lgrank" - Logrank-test • "jonckheere" - Jonckheere-Terpstra test of ordered alternatives • "glm" - generalized linear model
...	Additional arguments for the chosen test

Value

A list with CTP parameters

getResponse

Auxiliary function

Description

Getting the response vector out of an object of class 'formula'

Usage

```
getResponse(formula, data)
```


Arguments

formula	Model formula
data	Dataframe

Value

Vector of response

Examples

```
data(pasi)

data(pasi)
getResponse(pasi.ch ~ dose, data=pasi)

library(survival)
data(ovarian)
ovarian$subgroups <- as.factor(10*ovarian$ecog.ps+ovarian$rx)

surv <- getResponse(Surv(futime,fustat)~rx, data=ovarian)
print(surv)
class(surv)
```

glucose

Data from a study in Diabetes II patients

Description

The dataframe glucose comprises the baseline values of fasting plasma glucose (mmol/L) and their changes from baseline to the end of the study.

Usage

```
data(glucose)
```

Format

A data frame with 89 observations and 3 variables:

GLUCOSE.BLA Fasting plasma glucose at baseline

GLUCOSE.CHANGE Changes in fasting plasma glucose from baseline

DOSE An ordered factor with levels PLACEBO < LOW < MEDIUM < HIGH

IntersectHypotheses *Intersection of hypotheses*

Description

Generation of the hypotheses tree of a closed testing procedure (CTP). The function returns an object of oldClass "ctp.str"; summary and Display can be applied.

Usage

```
IntersectHypotheses(hyplst)
```

Arguments

hyplst A list of integer vectors representing the elementary hypotheses.

Details

Sets of elementary hypotheses are described by lists of integer vectors eg. `list(1:2, c(1, 3), c(1, 4))` if the populations 2, 3 and 4 have to be compared to population 1. For the generation of the hypothesis tree of a closed testing procedure first all intersections of the elementary hypotheses; then all intersections of these intersections etc. have to be created. The set of hypotheses at each intersection level must be reduced by deleting double hypotheses and creating unions. The size of the hypothesis tree increases rapidly with growing number of elementary hypotheses, which can lead to memory and/or time problems! The intersection procedure ends if a single hypothesis (the global hypothesis) is left.

Value

- hypothesis The hypothesis tree described as lists hypotheses at each intersection level.
- relations The relation structure of the hypotheses tree used for plotting.
- hypnames Character representations of the hypotheses used for summary tables and plotting.
- connections The connections used for calculating the adjusted p-values.

Note

This procedure is constructed for null-hypotheses describing the equality of the same parameter for different populations.

Warning: The size of the hypothesis tree increases rapidly with growing number of elementary hypotheses. This can lead to memory and/or time problems!

See Also

[AnalyseCTP Adjust_raw](#)

Examples

```
hlist <- list(1:2, c(1,3), c(1,4))
htree <- IntersectHypotheses(hlist)
summary(htree)
Display(htree)

# compare in a set of six means all others to the fourth mean
#
five.to.fourth <- IntersectHypotheses(list(c(1,4),c(2,4),c(3,4),c(4,5),c(4,6)))
Display(five.to.fourth)
```

pasi

Data from a Psoriasis study

Description

The dataframe `pasi` comprises the changes in PASI-score (Psoriasis Area and Severity Index) from Baseline within two months in 72 patients treated with different doses of Etretin or Placebo in a double blind study.

Usage

```
data(pasi)
```

Format

A data frame with 72 observations and 3 variables:

treatment a factor with levels Etretin10mg Etretin25mg Etretin50mg Placebo

pasi.ch Changes in PASI score within two months

dose Dose of Etretin as ordered factor with levels Placebo < ET.10mg < ET.25mg < ET.50mg

summary.ctp

Summarising and plotting the results of hypothesis trees and closed testing procedures

Description

The functions `summary.ctp.str` and `Display.ctp.str` summarise and plot the hypothesis tree of a closed testing procedure, whereas the functions `summary.ctp` and `Display.ctp` summarise and plot the results of a closed testing procedure.

Usage

```
## S3 method for class 'ctp'
summary(object, digits = 4, ...)

## S3 method for class 'ctp.str'
summary(object, ...)

Display(object, ...)

## S3 method for class 'ctp'
Display(object, Type = "s", arrow = FALSE, ...)

## S3 method for class 'ctp.str'
Display(object, Type = "s", arrow = FALSE, ...)
```

Arguments

object	Object generated by AnalyseCTP or IntersectHypotheses .
digits	The minimum number of significant digits to be printed for the p-values in the summary.
...	Additional arguments passed to <code>Display</code> .
Type	Type of the plotted tree: "s" straight lines, "" tree type.
arrow	If TRUE, an arrow is plotted in the connections.

Details

```
@name Generic_Functions
@aliases summary Display
```

Value

- `summary.ctp.str` provides lists with dataframes comprising the hypothesis names as strings and the relationship between the hypotheses.
- `summary.ctp` provides a table of hypotheses, raw p-values and adjusted p-values.

Note

`Disply.ctp.str` plots the hypothesis tree.
`Display.ctp` plots the hypothesis tree together with the unadjusted and adjusted p-values.

See Also

[IntersectHypotheses](#), [AnalyseCTP](#)

TestingSet	<i>Testing set for an elementary hypothesis</i>
------------	---

Description

Testing set for an elementary hypothesis

Usage

```
TestingSet(ctp.struc, Hyp)
```

Arguments

ctp.struc	Object of class ctp.str.
Hyp	Elementary hypothesis (character variable).

Value

The testing set for the elementary hypothesis (character vector).

Examples

```
Pairwise <- IntersectHypotheses(list(c(1,2), c(1,3),  
                                     c(1,4), c(2,3), c(2,4), c(3,4)))  
Set24    <- TestingSet(Pairwise,"[24]")  
Set24
```

Index

* datasets

- colorectal, [6](#)
- glucose, [9](#)
- pasi, [11](#)

Adjust_p, [2](#)

Adjust_raw, [2](#), [5](#), [7](#), [10](#)

AnalyseCTP, [3](#), [4](#), [6](#), [7](#), [10](#), [12](#)

colorectal, [6](#)

CTP, [6](#)

CTPcompare, [7](#)

Display, [3](#), [5](#)

Display (summary.ctp), [11](#)

getCTParms, [7](#)

getResponse, [8](#)

glucose, [9](#)

IntersectHypotheses, [3–6](#), [10](#), [12](#)

pasi, [11](#)

summary.ctp, [3](#), [5](#), [11](#)

summary.ctp.str, [3](#), [5](#), [7](#)

TestingSet, [13](#)