

Package ‘maicChecks’

May 8, 2026

Type Package

Title Exact Matching and Matching-Adjusted Indirect Comparison (MAIC)

Version 0.2.0

Date 2025-03-03

Maintainer Lillian Yau <maicChecks@gmail.com>

Description The second version (0.2.0) contains implementation for exact matching which is an alternative to propensity score matching (see Glimm & Yau (2025)). The initial version (0.1.2) contains a collection of easy-to-implement tools for checking whether a MAIC can be conducted, as well as an alternative way of calculating weights (see Glimm & Yau (2021) <[doi:10.1002/pst.2210](https://doi.org/10.1002/pst.2210)>.)

Depends R (>= 3.5.0)

Imports data.table, tidyr, ggplot2, lpSolve, quadprog

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Lillian Yau [aut, cre],
Ekkehard Glimm [aut],
Xinlei Deng [aut] (ORCID: <<https://orcid.org/0000-0001-8129-6007>>)

Repository CRAN

Date/Publication 2025-03-03 15:40:02 UTC

Contents

eAD	2
eIPD	3
exmLP.2ipd	3

exmWt.2ipd	4
maicLP	6
maicMD	6
maicPCA	8
maicT2Test	9
maicWt	10
maxessWt	11
sim110	12

Index	13
--------------	-----------

eAD	<i>three AD scenarios</i>
-----	---------------------------

Description

Three artificial scenarios serves as the ad cases. This is used in Glimm & Yau (2021)

Usage

```
data(eAD)
```

Format

scen corresponds to scenarios A, B, and C in the reference manuscript (Glimm & Yau (2021)). Scenario A is very close to IPD center (see `data(ipd)`) and is within the IPD convex hull; scenario B is further away from IPD center but otherwise still inside the IPD convex hull; scenario C is outside IPD convex hull.

y1 a numeric vector

y2 a numeric vector

References

Glimm & Yau (2021)

Examples

```
data(eAD)
```

`eIPD`*an IPD set*

Description

an artificial data set serves as the IPD set. this is used in Glimm & Yau (2021)

Usage

```
data(eIPD)
```

Format

y1 a numeric vector

y2 a numeric vector

References

Glimm & Yau (2021)

Examples

```
data(eIPD)
```

`exmLP.2ipd`*Checks whether two IPD datasets can be matched with lpSolve::lp*

Description

Checks whether two IPD datasets can be matched with lpSolve::lp

Usage

```
exmLP.2ipd(  
  ipd1,  
  ipd2,  
  vars_to_match = NULL,  
  cat_vars_to_01 = NULL,  
  mean.constrained = FALSE  
)
```

Arguments

ipd1	a dataframe with n1 row and p column, where n1 is number of subjects of the first IPD, and p is the number of variables used in standardization.
ipd2	a dataframe with n2 row and p column, where n2 is number of subjects of the second IPD, and p is the number of variables used in standardization.
vars_to_match	variables used for matching. if NULL, use all variables.
cat_vars_to_01	variable names for the categorical variables that need to be converted to indicator variables.
mean.constrained	whether to restrict the weighted means to be within the ranges of observed means. Default is FALSE. When it is TRUE, there is a higher chance of not having a solution.

Details

If dummy variables are already created for the categorical variables in the data set, and are present in ipd1 and ipd2, then cat_vars_to_01 should be left as NULL.

Value

lp.check 0 = OS can be conducted; 2 = OS cannot be conducted

Author(s)

Lillian Yau

Examples

```
## Not run:
ipd1 <- sim110[sim110$study == 'IPD A',]
ipd2 <- sim110[sim110$study == 'IPD B',]
x <- exmLP.2ipd(ipd1, ipd2, vars_to_match = paste0('X', 1:5),
cat_vars_to_01 = paste0('X', 1:3), mean.constrained = FALSE)

## End(Not run)
```

exmWt.2ipd

Exact matching for two IPD's

Description

Exact matching for two IPD's

Usage

```
exmWt.2ipd(
  ipd1,
  ipd2,
  vars_to_match = NULL,
  cat_vars_to_01 = NULL,
  mean.constrained = FALSE
)
```

Arguments

`ipd1` a dataframe with `n` row and `p` column, where `n` is number of subjects and `p` is the number of variables used in matching.

`ipd2` the other IPD with the same number of columns

`vars_to_match` variables used for matching. if `NULL`, use all variables.

`cat_vars_to_01` a list of variable names for the categorical variables that need to be converted to indicator variables.

`mean.constrained` whether to restrict the weighted means to be within the ranges of observed means. Default is `FALSE`. When it is `TRUE`, there is a higher chance of not having a solution.

Details

If dummy variables are already created for the categorical variables in the data set, and are present in `ipd1` and `ipd2`, then `cat_vars_to_01` should be left as `NULL`.

Value

`ipd1` re-scaled weights of the exact matching by maximizing ESS for IPD 1, and the input IPD 1 data with categorical variables converted to 0-1 indicators

`ipd2` re-scaled weights of the exact matching by maximizing ESS for IPD 2, and the input IPD 2 data with categorical variables converted to 0-1 indicators

`wtd.summ` ESS for IPD 1, ESS for IPD 2, and weighted means of the matching variables

Author(s)

Lillian Yau

Examples

```
## Not run:
ipd1 <- sim110[sim110$study == 'IPD A',]
ipd2 <- sim110[sim110$study == 'IPD B',]
x <- exmWt.2ipd(ipd1, ipd2, vars_to_match = paste0('X', 1:5),
  cat_vars_to_01 = paste0('X', 1:3), mean.constrained = FALSE)

## End(Not run)
```

maicLP	<i>Checks if AD is within the convex hull of IPD using lp-solve</i>
--------	---

Description

Checks if AD is within the convex hull of IPD using lp-solve

Usage

```
maicLP(ipd, ad)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects and p is the number of variables used in matching.
ad	a dataframe with 1 row and p column. The matching variables should be in the same order as that in ipd. The function does not check this.

Value

lp.check	0 = AD is inside IPD, and MAIC can be conducted; 2 = otherwise
----------	--

References

Glimm & Yau (2021). "Geometric approaches to assessing the numerical feasibility for conducting matching-adjusted indirect comparisons", *Pharmaceutical Statistics*, 21(5):974-987. doi:10.1002/pst.2210.

Examples

```
## eAD[1,] is the scenario A in the reference paper,
## i.e. when AD is within IPD convex hull
maicLP(eIPD, eAD[1,2:3])

## eAD[3,] is the scenario C in the reference paper,
## i.e. when AD is outside IPD convex hull
maicLP(eIPD, eAD[3,2:3])
```

maicMD	<i>Checks if AD is within the convex hull of IPD using Mahalanobis distance</i>
--------	---

Description

Should only be used when all matching variables are normally distributed

Usage

```
maicMD(ipd, ad, n.ad = Inf)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects and p is the number of variables used in matching.
ad	a dataframe with 1 row and p column. The matching variables should be in the same order as that in ipd. The function does not check this.
n.ad	default is Inf assuming ad is a fixed (known) quantity with infinit accuracy. In most MAIC applications ad is only the sample statistics and n.ad is known.

Details

When AD does not have the largest Mahalanobis distance, in the original scale AD can still be outside of the IPD convex hull. On the other hand, when AD does have the largest Mahalanobis distance, in the original scale, AD is for sure outside the IPD convex hull.

Value

Prints a message whether AD is furthest away from 0, i.e. IPD center in terms of Mahalanobis distance. Also returns ggplot object for plotting.

md.dplot	dot-plot of AD and IPD in Mahalanobis distance
md.check	0 = AD has the largest Mahalanobis distance to the IPD center; 2 = otherwise

References

Glimm & Yau (2021). "Geometric approaches to assessing the numerical feasibility for conducting matching-adjusted indirect comparisons", *Pharmaceutical Statistics*, 21(5):974-987. doi:10.1002/pst.2210.

Examples

```
## Not run:
## eAD[1,] is the scenario A in the reference paper,
## i.e. when AD is perfectly within IPD
md <- maicMD(eIPD, eAD[1,2:3])
md ## a dot-plot of IPD Mahalanobis distances along with AD in the same metric.

## End(Not run)
```

maicPCA

*Checks whether AD is outside IPD in PC coordinates***Description**

Checks whether AD is outside IPD in principal component (PC) coordinates

Usage

```
maicPCA(ipd, ad)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects in IPD set and p is the number of variables used in matching.
ad	a dataframe with 1 row and p column. The matching variables should be in the same order as that in ipd. The function does not check this.

Details

When AD is within the IPD PC ranges, AD can still be outside the IPD convex hull in the original scale. On the other hand, if AD is outside the IPD PC ranges, in the original scale AD is for sure outside the IPD convex hull.

Value

Prints a message whether AD is inside or outside IPD PC coordinates. Also returns a ggplot object to be plotted.

pc.dplot	dot-plot of AD and IPD both in IPD's PC coordinates
pca.check	0 = AD within the ranges of IPD's PC coordinates; 2 = otherwise

References

Glimm & Yau (2021). "Geometric approaches to assessing the numerical feasibility for conducting matching-adjusted indirect comparisons", *Pharmaceutical Statistics*, 21(5):974-987. doi:10.1002/pst.2210.

Examples

```
## Not run:
## eAD[1,] is the scenario A in the reference paper,
## i.e. when AD is perfectly within IPD
a1 <- maicPCA(eIPD, eAD[1,2:3])
a1 ## the dot plots of PC's for IPD and AD

## eAD[3,] is the scenario C in the reference paper,
## i.e. when AD is outside IPD
a3 <- maicPCA(eIPD, eAD[3,2:3])
```

```
a3 ## the dot plots of PC's for IPD and AD
## End(Not run)
```

maicT2Test

Hotelling's T-square test to check whether maic is needed

Description

Hotelling's T-square test to check whether maic is needed

Usage

```
maicT2Test(ipd, ad, n.ad = Inf)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects and p is the number of variables used in matching.
ad	a dataframe with 1 row and p column. The matching variables should be in the same order as that in ipd. The function does not check this.
n.ad	default is Inf assuming ad is a fixed (known) quantity with infinit accuracy. In most MAIC applications ad is the sample statistics and n.ad is known.

Details

When n.ad is not Inf, the covariance matrix is adjusted by the factor $n.ad/(n.ipd + n.ad)$, where n.ipd is nrow(ipd), the sample size of ipd.

Value

T.sq.f	the value of the T^2 test statistic
p.val	the p-value corresponding to the test statistic. When the p-value is small, matching is necessary.

References

Glimm & Yau (2021). "Geometric approaches to assessing the numerical feasibility for conducting matching-adjusted indirect comparisons", *Pharmaceutical Statistics*, 21(5):974-987. doi:10.1002/pst.2210.

Examples

```
## eAD[1,] is the scenario A in the reference paper,
## i.e. when AD is perfectly within IPD
maicT2Test(eIPD, eAD[1,2:3])
```

maicWt

*Estimates the MAIC weights***Description**

Estimates the MAIC weights for each individual in the IPD. Should only be used after it is ascertained that AD is indeed within the convex hull of IPD.

Usage

```
maicWt(ipd, ad, max.it = 25)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects and p is the number of variables used in matching.
ad	a dataframe with 1 row and p coln. The matching variables should be in the same order as that in ipd. The function does not check this.
max.it	maximum iteration passed to optim(). if ad is within ipd convex hull, then the default 25 iterations of optim() should be enough.

Value

The main code are taken from Filippo (2016). It returns the following:

optim.out	results of optim()
maic.wt	MAIC un-scaled weights for each subject in the IPD set
maic.wt.rs	re-scaled weights which add up to the original total sample size, i.e. nrow(ipd)
ipd.ess	effective sample size
ipd.wtsumm	weighted summary statistics of the matching variables after matching. they should be identical to the input AD when AD is within the IPD convex hull.

References

Phillippo DM, Ades AE, Dias S, et al. (2016). Methods for population-adjusted indirect comparisons in submissions to NICE. NICE Decision Support Unit Technical Support Document 18.

Examples

```
## eAD[1,] is scenario A in the reference manuscript
m1 <- maicWt(eIPD, eAD[1,2:3])
```

maxessWt	<i>Maximum ESS Weights</i>
----------	----------------------------

Description

Estimates an alternative set of weights which maximizes effective sample size (ESS) for a given set of variates used in the matching. Should only be used after it is ascertained that AD is indeed within the convex hull of IPD.

Usage

```
maxessWt(ipd, ad)
```

Arguments

ipd	a dataframe with n row and p column, where n is number of subjects and p is the number of variables used in matching.
ad	a dataframe with 1 row and p column. The matching variables should be in the same order as that in ipd. The function does not check this.

Details

The weights maximize the ESS subject to the set of baseline covariates used in the matching.

Value

maxess.wt	maximum ESS weights. Scaled to sum up to the total IPD sample size, i.e. nrow(ipd)
ipd.ess	effective sample size. It is no smaller than the ESS given by the MAIC weights.
ipd.wtsumm	weighted summary statistics of the matching variables after matching. they should be identical to the input AD when AD is within the IPD convex hull.

References

Glimm & Yau (2021). "Geometric approaches to assessing the numerical feasibility for conducting matching-adjusted indirect comparisons", *Pharmaceutical Statistics*, 21(5):974-987. doi:10.1002/pst.2210.

Examples

```
## eAD[1,] is scenario A in the reference manuscript  
m0 <- maxessWt(eIPD, eAD[1,2:3])
```

`sim110`*Simulated data used for exact matching*

Description

sim110 is one of the simulated data presented in the simulation study in Glimm & Yau (2025). The covariates used in matching are X1 to X15. A response variable Y is simulated to depend on 6 of the 15 covariates.

Usage

```
data(sim110)
```

Format

X1 to X15 Covariates used in matching

Y Response variable

study IPD A and IPD B

References

Glimm & Yau (2025)

Examples

```
data(sim110)
```

Index

* datasets

eAD, [2](#)

eIPD, [3](#)

sim110, [12](#)

eAD, [2](#)

eIPD, [3](#)

exmLP.2ipd, [3](#)

exmWt.2ipd, [4](#)

maicLP, [6](#)

maicMD, [6](#)

maicPCA, [8](#)

maicT2Test, [9](#)

maicWt, [10](#)

maxessWt, [11](#)

sim110, [12](#)