

Package ‘validateRS’

December 26, 2015

Type Package

Title One-Sided Multivariate Testing Procedures for Rating Systems

Version 1.0.0

Date 2015-12-24

Author Coppens F., Mayer M., Millischer L., Resch F., Sauer S., Schulze K.

Maintainer Coppens F. <francois.coppens@nbb.be>

Description An implementation of statistical tests for the validation of rating systems as described in the ECB Working paper "Advances in multivariate back-testing for credit risk underestimation", by F. Coppens, M. Mayer, L. Millischer, F. Resch, S. Sauer, K. Schulze (ECB WP series, forthcoming).

Depends truncnorm, triangle, reshape2, data.table

License EUPL

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-26 10:04:35

R topics documented:

validateRS-package	2
minP.adj.pvalue	4
par.dist.default	5
power.target.Nclasses	6
ratingData	7
region.acceptance	8
region.power	9
sample.knowledge.H1	10
simul.scenario.rs	11

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

validateRS-package *Multiple testing procedures for validation of rating systems.*

Description

This package implements statistical tests described in the ECB Working paper "Advances in multivariate back-testing for credit risk underestimation", by F. Coppens, M. Mayer, L. Millischer, F. Resch, S. Sauer, K. Schulze (ECB WP series, forthcoming). The empirical results in the paper were obtained using the package's functions.

The data on the rating classes, i.e. the PDs (table 1 of the ECB Working paper) and the scenarios for the class sizes (table 3 of the ECB Working paper), can be found in the object `ratingData`.

The power of the test can be computed for different "scenarios" for the alternative hypothesis H1. There are three scenario's for H1 (see ECB Working paper, table 2): (1) H1A where H1 is obtained by increasing all the PDs under the null hypothesis towards 1, (2) H1B where H1 is obtained by increasing one PD at a time towards 1 and (3) H1C where the PDs under H1 are drawn from a multivariate normal distribution.

H1A and H1B can be computed using the function `power.target.Nclasses` and H1C with the function `sample.knowledge.H1`. Examples are given below.

Questions on the package can be asked via the 'stackexchange' network, in particular the 'crossvalidated' community.

Author(s)

Coppens F., Mayer M., Millischer L., Resch F., Sauer S., Schulze K.

References

F. Coppens, M. Mayer, L. Millischer, F. Resch, S. Sauer, K. Schulze, "Advances in multivariate back-testing for credit risk underestimation", ECB WP series, forthcoming

Examples

```
## Not run:
data(ratingData)
rc.selected<-1:5
si.selected<-1:2
p.0<-ratingData$p.0[rc.selected]
sizes<-ratingData$sizes[si.selected,rc.selected]

# =====
# compute alternative hypothesis H1A
# =====
H1A<-power.target.Nclasses(p.0=p.0, size=sizes[1,],
                           N=length(p.0),
                           target=0.50)

# compute minP's region and its power under H1A (the latter must be close to 0.5)
```

```

r.mp<-region.acceptance(hypo.test="minP", p.0=p.0, size=sizes[1,], alpha=0.05)
region.power(region=r.mp, p.1=H1A[1,])

# compute enhanced test's region and its power under H1A
# (result in table 4 of the ECB Working paper)
r.mpp<-region.acceptance(hypo.test="minPp", p.0=p.0, size=sizes[1,], alpha=0.05)
region.power(region=r.mpp, p.1=H1A[1,])

# compute envelope test's region and its power under H1A
# (result in table 4 of the ECB Working paper)
r.sh<-region.acceptance(hypo.test="sterneHull", p.0=p.0, size=sizes[1,], alpha=0.05)
region.power(region=r.sh, p.1=H1A[1,])

# =====
# compute alternative hypothesis H1B
# =====
H1B<-power.target.Nclasses(p.0=p.0, size=sizes[1,],
                           N=1,
                           target=0.30)

# compute minP's region and its power under H1B (the latter must be close to 0.3)
r.mp<-region.acceptance(hypo.test="minP", p.0=p.0, size=sizes[1,], alpha=0.05)
power.vec<-vector(mode="numeric", length=nrow(H1B))
for ( i in 1:nrow(H1B) ) {
  power.vec[i]<-region.power(region=r.mp, p.1=H1B[i,])
}
cat(paste("mean :", mean(power.vec), "\n"))

# compute enhanced test's region and its power under H1B
# (result in table 4 of the ECB Working paper)
r.mpp<-region.acceptance(hypo.test="minPp", p.0=p.0, size=sizes[1,], alpha=0.05)
power.vec<-vector(mode="numeric", length=nrow(H1B))
for ( i in 1:nrow(H1B) ) {
  power.vec[i]<-region.power(region=r.mpp, p.1=H1B[i,])
}
cat(paste("mean :", mean(power.vec), "\n"))

# compute envelope test's region and its power under H1B
# (result in table 4 of the ECB Working paper)
r.sh<-region.acceptance(hypo.test="sterneHull", p.0=p.0, size=sizes[1,], alpha=0.05)
power.vec<-vector(mode="numeric", length=nrow(H1B))
for ( i in 1:nrow(H1B) ) {
  power.vec[i]<-region.power(region=r.sh, p.1=H1B[i,])
}
cat(paste("mean :", mean(power.vec), "\n"))

# =====
# compute alternative hypothesis H1C
# =====
param<-par.dist.default(dist="tr.normal", p.0=p.0)
set.seed(1)
sample.h1<-sample.knowledge.H1(n=10000, par=param, p.0=p.0)

```

```

sim.mp<-simul.scenario.rs(hypo.test="minP", p.0=p.0, sampleH1=sample.h1, sizes=sizes, alpha=0.05)
sim.mp$power$weighted.average

sim.mpp<-simul.scenario.rs(hypo.test="minPp", p.0=p.0, sampleH1=sample.h1, sizes=sizes, alpha=0.05)
sim.mpp$power$weighted.average

## End(Not run)

```

minP.adj.pvalue	<i>Adjusted p-values for the multiple test.</i>
-----------------	---

Description

Computes the adjusted p-values for multiple test (minP).

Usage

```
minP.adj.pvalue(p.0, size, defaultPattern)
```

Arguments

p.0	A vector with the PD values under H0
size	A vector with the sizes of the rating classes
defaultPattern	A vector with the observed default pattern

Details

Compute the adjusted p-values for the multiple test.

Value

A list with

par	The parameters used
raw.p.value	The raw p-values
minp.adj.p.value	The adjusted p-value

Examples

```
mpa<-minP.adj.pvalue(p.0=c(0.001, 0.004), size=c(1000, 1000),
  defaultPattern=c(0, 0))
```

par.dist.default *Create default parameters for a distribution*

Description

A distribution for knowledge about the values under H1 is assumed. The function *par.dist.default* creates default values for the parameters of the distribution.

Usage

```
par.dist.default(dist, p.0)
```

Arguments

dist	The name of the distribution. Possible values are “triangular” (triangular distribution) and “tr.normal” (truncated normal distribution). The value “usersupplied” generates a parameter structure that the caller must first fill in with his own list of p.1s (see the examples below).
p.0	The PD values under H0. The default parameter values are derived from p.0

Details

The structure of the parameters depends on the distribution. The function *par.dist.default* creates default values for the parameters. These defaults can be changed but a call to the function is required such that the structure of the parameters is known.

The return value is to be used in a call to [sample.knowledge.H1](#) (also for the “usersupplied”) distribution.

Value

The function returns a list with elements

dist	the distribution chosen
p.0	the value of p.0 that was used
param	For a triangular distribution these are a (The lower value of the base of the triangle), b (The higher value of the base of the triangle) and mode (The mode of the triangle). There is a value for a, b and mode for each rating class. For a truncated normal distribution the lowest value for the truncation (for each class), the highest value for truncation (for each class), the mean and the standard deviations (both for each class). For a “usersupplied” distribution only the parameter structure is created, param is left NA. The user should fill in an array of H1 values in param.

See Also

[sample.knowledge.H1](#)

Examples

```

par.trian<-par.dist.default(dist="triangular", p.0=c(0.001, 0.004))

par.trnormal<-par.dist.default(dist="tr.normal", p.0=c(0.001, 0.004))

par.ownsample<-par.dist.default(dist="usersupplied", p.0=c(0.001, 0.004))
#first fill in your own list of p1.s
par.ownsample$param<-rbind(c(0.001, 0.01), c(0.004, 0.004), c(0.004, 0.01))
# see also \code{simulate.scenario}

```

power.target.Nclasses *Compute the alternative hypothesis' PDs by increasing the PDs under the null hypothesis towards 1.*

Description

Compute the alternative hypothesis' PDs by increasing the PDs under the null hypothesis towards 1.

One can either increase all the PDs (i.e. alternative H1A in the ECB Working series' paper) or one at a time (scenario H1B).

The alternative hypothesis H1C can be computed using [sample.knowledge.H1](#).

Usage

```
power.target.Nclasses(p.0, size, N , target = 0.5, alpha = 0.05, precision = 1e-07)
```

Arguments

p.0	A Vector with the PD values under H0
size	A Vector with the size of the rating classes
N	The number of PDs in which H1 should be different from the PDs under H0. For the alternative hypothesis H1A of the ECB Working paper, N should be set to length(p.0), for the alternative hypothesis H1B N should be 1.
target	The target power for the benchmark test
alpha	The significance level chosen
precision	The precision for the PDs under H1

Value

A vector with PD values under H1, such that the power of the benchmark equals the target power.

See Also

[sample.knowledge.H1](#)

Examples

```
data(ratingData)

p.0<-ratingData$p.0
sizes<-ratingData$sizes

h1A<-power.target.Nclasses(p.0=p.0, size=sizes[1,], N=length(p.0),
                           target=0.50)
h1A

h1B<-power.target.Nclasses(p.0=p.0, size=sizes[,], N=1,
                           target=0.30)
h1B
```

ratingData	<i>Data for rating classes of S&P</i>
------------	---

Description

Data for rating classes of S&P

Usage

```
data(ratingData)
```

Format

A list with:

- SP.p.0: vector with PDs of S&P classes
- SP.sizes: vector with sizes of S&P classes

Details

Data that can be used for the function `simul.scenario.rs`. For a large number of classes a lot of memory is required. For computers with a limited amount of memory lesser classes can be obtained by indexing.

The number of rated entities and the PDs over a one-year horizon in the six S&P classes reflect the statistics for S&P's global corporate ratings available at https://www.nact.org/resources/NACT_2012_Global_Corporate_Def. S&P's rating grades are assigned to six classes according to the mapping report of the European Supervisory Authorities for the implementation of the Basel III banking regulation in the European Union (Capital Requirements Regulation (EU) No 575/2013) (see <https://www.eba.europa.eu/regulation-and-policy/external-credit-assessment-institutions-ecai/draft-implementing-technical-standards-on-the-mapping-of-ecais-credit-assessments>).

See Also

[simul.scenario.rs](#)

Examples

```
data(ratingData)
```

```
region.acceptance      Determine acceptance region for a test
```

Description

The acceptance region for a particular test is determined.

Usage

```
region.acceptance(hypo.test, p.0, size, alpha)
```

Arguments

hypo.test	A string containing the name of the test "minP" for the multiple test, "minPp" for the enhanced multiple test or "sterneHull" for the envelope test.
p.0	A vector containing the PD values under H0
size	A vector with the sizes of the rating classes
alpha	The desired significance level

Details

The function computes the acceptance region, the observed significance level and the cardinality of the acceptance region.

Value

A list containing

par	The parameters that were passed to the function
acc.reg	The name of the test and a description of the acceptance region. For the multiple test this is the lower left point of the critical region, for the enhanced multiple test this is the lower left point of the critical region and the equation of the cutting hyperplane. For the envelope test it contains the coordinates of all the points.
optim.data	Data needed for optimal computations in other functions
observedAlpha	The observed significance level
cardinality	The cardinality of the acceptance region

Examples

```
mp<-region.acceptance(hypo.test="minP", p.0=c(0.001, 0.004), size=c(1000, 1000), alpha=0.05)
mpp<-region.acceptance(hypo.test="minPp", p.0=c(0.001, 0.004), size=c(1000, 1000), alpha=0.05)
```

region.power	<i>Compute the power for a region and a given simple alternative hypothesis.</i>
--------------	--

Description

The power of a region is computed for a simple alternative hypothesis. The region should be created by a call to `region.acceptance`.

Usage

```
region.power(region, p.1)
```

Arguments

region	The region for which the power should be computed. Should be the return value of a call to <code>region.acceptance</code> .
p.1	A vector with the values for the simple alternative hypothesis.

Details

The power of a region is computed for a simple alternative hypothesis. The region should be created by a call to `region.acceptance`.

Value

The power of the region for the given simple alternative hypothesis.

See Also

[region.acceptance](#)

Examples

```
mp<-region.acceptance(hypo.test="minP", p.0=c(0.001, 0.004), size=c(1000, 1000), alpha=0.5)
region.power(mp, p.1=c(0.001, 0.01))
```

sample.knowledge.H1 *Draw a random sample from assumed distribution of H1 values. This function can be used to compute alternative hypothesis H1C from the ECB Working paper.*

Description

A sample of a certain size is drawn from a chosen distribution. The parameters for the distribution are obtained through a call to the function *par.dist.default*. The parameters may be changed, although the structure should not be changed.

For alternative hypothesis H1C one should use the truncated normal distribution for the parameter "par".

Usage

```
sample.knowledge.H1(n, par, p.0=NULL)
```

Arguments

n	The size of the sample to be drawn
par	The distribution and its parameters.
p.0	A vector with the PD values under H0. The default value is NULLL. If a vector is supplied, then, for each draw, only the p.1 rows where at least one pd is higher than the corresponding p.0 pd are kept.

Details

A sample of size n is drawn from the distribution.

Value

A list with elements:

dist	the name of the distribution that was used
par	the parameters of the distribution that was used
p.1	the sample drawn
stats.wrong.pds	The p.1 are randomly drawn from the distribution given in the parameter par. By the random drawing it might be that the outcome does not fulfill H1. In that case the draw is excluded from the list-element p.1 and added to the list element stats.wrong.pds

See Also

[par.dist.default](#), [simul.scenario.rs](#)

Examples

```
## Not run:
s.nrm<-sample.knowledge.H1(n=100, par=par.dist.default(dist="tr.normal",
  p.0=c(0.001, 0.004)))

## End(Not run)
```

simul.scenario.rs *Determine cardinality and power for a statistical test.*

Description

For a sample drawn with *sample.knowledge.H1* and for a statistical test, the acceptance region is determined and the power for each value of H1 in the sample is computed.

Usage

```
simul.scenario.rs(hypo.test, p.0, sampleH1, sizes, alpha)
```

Arguments

hypo.test	The name of the hypothesis test: “minP” for the multiple test , “minPp” for the enhanced multiple test or “sterneHull” for the envelope test.
p.0	A vector containing the PDs under H0
sampleH1	The sample drawn using the function <i>sample.knowledge.H1</i> .
sizes	A matrix with different scenario’s for the sizes of the rating classes. There should be at least two rows and the number of columns must equal the length of the parameter p.0.
alpha	The significance level

Value

A list with elements

par	The parameters passed to the function
power	Power for each value of H1, the average power and the standard error on the average
cardinality.AR	The cardinality of the acceptance region
executionTime	The time it took to compute the acceptance region and to compute the power
knowledgeH1	The characteristics of the distribution and the sample drawn

See Also

[sample.knowledge.H1](#),

Examples

```
p.0<-c(0.001, 0.004)
sizes<-rbind( c(500, 500),
              c(1000, 5000))
alpha<-0.05
## Not run:
s.nrm<-sample.knowledge.H1(n=10,
                          par=par.dist.default(dist="tr.normal", p.0=c(0.001, 0.004)))
scen.nrm.mp<-simul.scenario.rs(hypo.test="minP", p.0=p.0,
                              sampleH1=s.nrm, sizes=sizes, alpha=alpha)
scen.nrm.mpp<-simul.scenario.rs(hypo.test="minPp", p.0=p.0,
                                sampleH1=s.nrm, sizes=sizes, alpha=alpha)

par.ownsample<-par.dist.default(dist="usersupplied", p.0=c(0.001, 0.004))
par.ownsample$param<-rbind(c(0.001, 0.01), c(0.004, 0.004), c(0.004, 0.01))
s.own<-sample.knowledge.H1(n=10,
                          par=par.ownsample)

scen.ownsample.mp<-simul.scenario.rs(hypo.test="minP", p.0=p.0,
                                    sampleH1=s.own, sizes=sizes, alpha=alpha)

## End(Not run)
```

Index

`minP.adj.pvalue`, [4](#)

`par.dist.default`, [5](#), [10](#)

`power.target.Nclasses`, [6](#)

`ratingData`, [7](#)

`region.acceptance`, [8](#), [9](#)

`region.power`, [9](#)

`sample.knowledge.H1`, [5](#), [6](#), [10](#), [11](#)

`simul.scenario.rs`, [7](#), [10](#), [11](#)

`validateRS` (`validateRS`-package), [2](#)

`validateRS`-package, [2](#)