Package 'CerioliOutlierDetection'

June 23, 2024

Type Package

Title Outlier Detection Using the Iterated RMCD Method of Cerioli (2010)

Version 1.1.15

Date 2024-06-22

Maintainer Christopher G. Green <christopher.g.green@gmail.com>

Depends R (>= 4.0.0)

Imports robustbase (>= 0.91-1)

Description Implements the iterated RMCD method of Cerioli (2010) for multivariate outlier detection via robust Mahalanobis distances. Also provides the finite-sample RMCD method discussed in the paper, as well as the methods provided in Hardin and Rocke (2005) <doi:10.1198/106186005X77685> and Green and Martin (2017) <https://christopherggreen.github.io/papers/hr05_extension.pdf>. See also Chapter 2 of Green (2017) <https://digital.lib.washington.edu/researchworks/handle/1773/40304>.

License GPL (>= 2)

Suggests rrcov, mvtnorm, mclust

URL https://christopherggreen.github.io/CerioliOutlierDetection/

NeedsCompilation no

Author Christopher G. Green [aut, cre] (<https://orcid.org/0000-0003-1277-6597>), R. Doug Martin [ths]

Repository CRAN

Date/Publication 2024-06-23 06:10:02 UTC

Contents

`cerioli2010.fsrmcd.test` *Finite-Sample Reweighted MCD Outlier Detection Test of Cerioli (2010)*

Description

Given a set of observations, this function tests whether there are outliers in the data set and identifies outlying points. Outlier testing/identification is done using the Mahalanobis-distances based on the MCD dispersion estimate. The finite-sample reweighted MCD method of Cerioli (2010) is used to test for unusually large distances, which indicate possible outliers.

Usage

```
cerioli2010.fsrmcd.test(datamat,
  mcd.alpha = max.bdp.mcd.alpha(n,v),
  signif.alpha = 0.05, nsamp = 500,
  nmini = 300, trace = FALSE,
  delta = 0.025, hrdf.method=c("GM14","HR05"))
```

Arguments

`datamat`	(Data Frame or Matrix) Data set to test for outliers (rows = observations, columns = variables). `datamat` cannot have missing values; please deal with them prior to calling this function. `datamat` will be converted to a matrix.
`mcd.alpha`	(Numeric) Value to control the fraction of observations used to compute the covariance matrices in the MCD calculation. Default value is corresponds to the maximum breakpoint case of the MCD; valid values are between 0.5 and 1. See the `covMcd` documentation in the `robustbase` library for further details.
`signif.alpha`	(Numeric) Desired nominal size α of the *individual* outlier test (default value is 0.05). Equivalently, significance level at which to test individual observations for outlyingness. (This is the α parameter in Cerioli (2010).) To test the intersection hypothesis of no outliers in the data, specify $$\alpha = 1 - (1 - \gamma)^{(1/n)},$$ where γ is the nominal size of the intersection test and n is the number of observations.
`nsamp`	(Integer) Number of subsamples to use in computing the MCD. See the `covMcd` documentation in the `robustbase` library.

nmini	(Integer) See the covMcd documentation in the robustbase library.
trace	(Logical) See the covMcd documentation in the robustbase library.
delta	(Numeric) False-positive rate to use in the reweighting step (Step 2). Defaults to 0.025 as used in Cerioli (2010). When the ratio n/ν of sample size to dimension is very small, using a smaller delta can improve the accuracy of the method.
hrdf.method	(String) Method to use for computing degrees of freedom and cutoff values for the non-MCD subset. The original method of Hardin and Rocke (2005) and the expanded method of Green and Martin (2017) are available as the options "HR05" and "GM14", respectively. "GM14" is the default, as it is more accurate across a wider range of mcd.alpha values.

Value

mu.hat	Location estimate from the MCD calculation
sigma.hat	Dispersion estimate from the MCD calculation
mahdist	Mahalanobis distances calculated using the MCD estimate
DD	Hardin-Rocke or Green-Martin critical values for testing MCD distances. Used to produce weights for reweighted MCD. See Equation (16) in Cerioli (2010).
weights	Weights used in the reweighted MCD. See Equation (16) in Cerioli (2010).
mu.hat.rw	Location estimate from the reweighted MCD calculation
sigma.hat.rw	Dispersion estimate from the reweighted MCD calculation
mahdist.rw	a matrix of dimension nrow(datamat) by length(signif.alpha) of Mahalanobis distances computed using the finite-sample reweighted MCD methodology in Cerioli (2010). Even though the distances do not depend on signif.alpha, there is one column per entry in signif.alpha for user convenience.
critvalfcn	Function to compute critical values for Mahalanobis distances based on the reweighted MCD; see Equations (18) and (19) in Cerioli (2010). The function takes a signifance level as its only argument, and provides a critical value for each of the original observations (though there will only be two unique values, one for points included in the reweighted MCD (weights == 1) and one for points excluded from the reweighted MCD (weights == 0)).
signif.alpha	Significance levels used in testing.
mcd.alpha	Fraction of the observations used to compute the MCD estimate
outliers	A matrix of dimension nrow(datamat) by length(signif.alpha) indicating whether each row of datamat is an outlier. The i-th column corresponds to the result of testing observations for outlyingness at significance level signif.alpha[i].

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

Andrea Cerioli. Multivariate outlier detection with high-breakdown estimators. Journal of the American Statistical Association, 105(489):147-156, 2010. doi:10.1198/jasa.2009.tm09147

Andrea Cerioli, Marco Riani, and Anthony C. Atkinson. Controlling the size of multivariate outlier tests with the MCD estimator of scatter. Statistical Computing, 19:341-353, 2009. doi:10.1007/s1122200890965

See Also

cerioli2010.irmcd.test

Examples

```
require(mvtnorm, quiet=TRUE)

############################################
# dimension v, number of observations n
v <- 5
n <- 200
simdata <- array( rmvnorm(n*v, mean=rep(0,v),
    sigma = diag(rep(1,v))), c(n,v) )
#
# detect outliers with nominal sizes
# c(0.05,0.01,0.001)
#
sa <- 1. - ((1. - c(0.05,0.01,0.001))^(1./n))
results <- cerioli2010.fsrmcd.test( simdata,
    signif.alpha=sa )
# count number of outliers detected for each
# significance level
colSums( results$outliers )


############################################
# add some contamination to illustrate how to
# detect outliers using the fsrmcd test
# 10/200 = 5% contamination
simdata[ sample(n,10), ] <- array(
  rmvnorm( 10*v, mean=rep(2,v), sigma = diag(rep(1,v))),
  c(10,v)
)
results <- cerioli2010.fsrmcd.test( simdata,
  signif.alpha=sa )
colMeans( results$outliers )


## Not run:
############################################
# example of how to ensure the size of the intersection test is correct

  n.sim <- 5000
```

```
  simdata <- array(
    rmvnorm(n*v*n.sim, mean=rep(0,v), sigma=diag(rep(1,v))),
    c(n,v,n.sim)
  )
  # in practice we'd do this using one of the parallel processing
  # methods out there
  sa <- 1. - ((1. - 0.01)^(1./n))
  results <- apply( simdata, 3, function(dm) {
    z <- cerioli2010.fsrmcd.test( dm,
      signif.alpha=sa )
    # true if outliers were detected in the data, false otherwise
    any(z$outliers[,1,drop=TRUE])
  })
  # count the percentage of samples where outliers were detected;
  # should be close to the significance level value used (0.01) in these
  # samples for the intersection test.
  mean(results)


## End(Not run)
```

`cerioli2010.irmcd.test`

Iterated RMCD test of Cerioli (2010)

Description

Given a set of observations, this function tests whether there are outliers in the data set and identifies outlying points. Outlier testing/identification is done using the Mahalanobis-distances based on the MCD dispersion estimate. The iterated reweighted MCD method of Cerioli (2010) is used to ensure the intersection test has the specified nominal size (Type I error rate).

Usage

```
cerioli2010.irmcd.test(datamat,
  mcd.alpha = max.bdp.mcd.alpha(n,v),
  signif.gamma = 0.05, nsamp = 500,
  nmini = 300, trace = FALSE,
  delta = 0.025, hrdf.method=c("GM14","HR05"))
```

Arguments

`datamat`	(Data Frame or Matrix) Data set to test for outliers (rows = observations, columns = variables). `datamat` cannot have missing values; please deal with them prior to calling this function. `datamat` will be converted to a matrix.
`mcd.alpha`	(Numeric) Value to control the fraction of observations used to compute the covariance matrices in the MCD calculation. Default value is corresponds to the maximum breakpoint case of the MCD; valid values are between 0.5 and 1. See the `covMcd` documentation in the `robustbase` library for further details.

`signif.gamma` (Numeric) Desired nominal size of the *intersection* outlier test (e.g., 0.05), i.e., a test that there are no outliers in the data. (This is the γ parameter in Cerioli (2010).) The corresponding α parameter for testing individual observations for outlyingness will be calculated from γ as

$$\alpha = 1 - (1 - \gamma)^{(1/n)}.$$

`nsamp` (Integer) Number of subsamples to use in computing the MCD. See the `covMcd` documentation in the `robustbase` library.

`nmini` (Integer) See the `covMcd` documentation in the `robustbase` library.

`trace` (Logical) See the `covMcd` documentation in the `robustbase` library.

`delta` (Numeric) False-positive rate to use in the reweighting step (Step 2). Defaults to 0.025 as used in Cerioli (2010). When the ratio n/ν of sample size to dimension is very small, using a smaller delta can improve the accuracy of the method.

`hrdf.method` (String) Method to use for computing degrees of freedom and cutoff values for the non-MCD subset. The original method of Hardin and Rocke (2005) and the expanded method of Green and Martin (2017) are available as the options "HR05" and "GM14", respectively. "GM14" is the default, as it is more accurate across a wider range of `mcd.alpha` values.

Details

Calls the finite-sample reweighted MCD (FSRMCD) outlier detection function `cerioli2010.fsrmcd.test` first to test for the existence of any outliers in the data. If the FSRMCD method rejects the null hypothesis of no outliers in the data, individual observations are then tested for outlyingness using the critical value function returned by `cerioli2010.fsrmcd.test` with significance γ.

Value

`outliers` A matrix of dimension `nrow(datamat)` by `length(signif.gamma)` indicating whether each row of `datamat` is an outlier. The `i-th` column corresponds to the result of testing observations for outlyingness at significance level `signif.gamma[i]`.

`mahdist.rw` a matrix of dimension `nrow(datamat)` by `length(signif.gamma)` of Mahalanobis distances computed using the finite-sample reweighted MCD methodology in Cerioli (2010). Even though the distances do not depend on `signif.gamma`, there is one column per entry in `signif.gamma` for user convenience.

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

Andrea Cerioli. Multivariate outlier detection with high-breakdown estimators. Journal of the American Statistical Association, 105(489):147-156, 2010. doi:10.1198/jasa.2009.tm09147

Andrea Cerioli, Marco Riani, and Anthony C. Atkinson. Controlling the size of multivariate outlier tests with the MCD estimator of scatter. Statistical Computing, 19:341-353, 2009. doi:10.1007/s1122200890965

See Also

cerioli2010.fsrmcd.test

Examples

```
require(mvtnorm, quiet=TRUE)

##########################################
# dimension v, number of observations n
v <- 5
n <- 200
simdata <- array( rmvnorm(n*v, mean=rep(0,v),
    sigma = diag(rep(1,v))), c(n,v) )
# detect outliers
results <- cerioli2010.irmcd.test( simdata,
    signif.gamma=c(0.05,0.01,0.001) )
# count number of outliers detected for each
# significance level
colSums( results$outliers )


##########################################
# add some contamination to illustrate how to
# detect outliers using the irmcd test
# 10/200 = 5% contamination
simdata[ sample(n,10), ] <- array(
  rmvnorm( 10*v, mean=rep(2,v), sigma = diag(rep(1,v))),
  c(10,v)
)
results <- cerioli2010.irmcd.test( simdata,
  signif.gamma=0.01 )
mean( results$outliers[,1,drop=TRUE] )

##########################################
# banknote example from Cerioli (2010)
## Not run:

  require(rrcov) # for CovMcd
  require(mclust)  # banknote data set lives here
  data(banknote, package="mclust")
  # length, width of left edge, width of right edge,
  # width of bottom edge, width of top edge, length
  # of image diagonal, counterfeit (1=counterfeit)

  bnk.gamma <- 0.01
  # genuine banknotes
  # classical mean and covariance
  banknote.real <- banknote[ banknote[,"Status"]=="genuine", 2:7 ]
  cov.cls <- CovClassic( banknote.real  )
  # 1 - (1 - 0.01)^(1/100) quantile of scaled-Beta distribution
  # with m=100 and v=6
  bnk.m <- nrow( banknote.real )
```

```
bnk.v <- ncol( banknote.real )
bnk.alpha <- 1. - ((1. - bnk.gamma)^(1./bnk.m))
cutoff.cls <- (bnk.m-1.)*(bnk.m-1.)*qbeta( 1. - bnk.alpha, bnk.v/2.,
(bnk.m - bnk.v - 1.)/2.)/bnk.m
# Figure 4 (left) in Cerioli (2010)
plot( getDistance( cov.cls ), xlab="Index number",
ylab="Squared Mahalanobis Distance", type="p",
ylim=c(0,45)
)
abline( h=cutoff.cls )

# reweighted MCD, maximum breakdown point case
cov.rob <- CovMcd( banknote.real,
alpha=floor((bnk.m + bnk.v + 1.)/2.)/bnk.m, nsamp="best" )
# cutoff using chi-squared individually
cutoff.rmcdind <- qchisq(1. - bnk.gamma, df=bnk.v)
# cufoff using simultaneous chi-square
cutoff.rmcdsim <- qchisq(1. - bnk.alpha, df=bnk.v)
# scaled-F cutoff using FSRMCD
# cutoff value is returned by critvalfcn for observations
# with weight=0
tmp.fsrmcd <- cerioli2010.fsrmcd.test( banknote.real,
signif.alpha=bnk.alpha )
cutoff.fsrmcd <- unique(tmp.fsrmcd$critvalfcn( bnk.alpha )[tmp.fsrmcd$weights==0])
# Figure 4 (right)
plot( getDistance( cov.rob ), xlab="Index number",
ylab="Squared Robust Reweighted Distance", type="p",
ylim=c(0,45)
)
abline( h=cutoff.rmcdind, lty="dotted" )
abline( h=cutoff.rmcdsim, lty="dashed" )
abline( h=cutoff.fsrmcd, lty="solid" )
legend( "topright", c("RMCD_ind","RMCD","FSRMCD"),
  lty=c("dotted","dashed","solid") )

# forged banknotes
# classical mean and covariance
banknote.fake <- banknote[ banknote[,"Status"]=="counterfeit", 2:7 ]
cov.cls <- CovClassic( banknote.fake  )
# 1 - (1 - 0.01)^(1/100) quantile of scaled-Beta distribution
# with m=100 and v=6
bnk.m <- nrow( banknote.fake )
bnk.v <- ncol( banknote.fake )
bnk.alpha <- 1. - ((1. - bnk.gamma)^(1./bnk.m))
cutoff.cls <- (bnk.m-1.)*(bnk.m-1.)*qbeta( 1. - bnk.alpha, bnk.v/2.,
  (bnk.m - bnk.v - 1.)/2.)/bnk.m
# Figure 5 (left) in Cerioli (2010)
plot( getDistance( cov.cls ), xlab="Index number",
  ylab="Squared Mahalanobis Distance", type="p",
  ylim=c(0,45)
)
abline( h=cutoff.cls )
```

```
  # reweighted MCD, maximum breakdown point case
  cov.rob <- CovMcd( banknote.fake,
    alpha=floor((bnk.m + bnk.v + 1.)/2.)/bnk.m, nsamp="best" )
  # cutoff using chi-squared individually
  cutoff.rmcdind <- qchisq(1. - bnk.gamma, df=bnk.v)
  # scaled-F cutoff using FSRMCD
  # cutoff value is returned by critvalfcn for observations
  # with weight=0
  tmp.fsrmcd <- cerioli2010.fsrmcd.test( banknote.fake,
    signif.alpha=bnk.alpha )
  cutoff.fsrmcd <- unique(tmp.fsrmcd$critvalfcn( bnk.alpha )[tmp.fsrmcd$weights==0])
  cutoff.irmcd <- unique(tmp.fsrmcd$critvalfcn( bnk.gamma )[tmp.fsrmcd$weights==0])
  # Figure 5 (right) in Cerioli (2010)
  plot( getDistance( cov.rob ), xlab="Index number",
    ylab="Squared robust reweighted Distance", type="p",
    ylim=c(0,150)
  )
  abline( h=cutoff.rmcdind, lty="dotted" )
  abline( h=cutoff.fsrmcd, lty="dashed" )
  abline( h=cutoff.irmcd, lty="solid" )
  legend( "topright", c("RMCD_ind","FSRMCD","IRMCD"),
    lty=c("dotted","dashed","solid") )


## End(Not run)

##############################################
# example of how to ensure the size of the intersection test is correct
## Not run:
  n.sim <- 5000
  simdata <- array(
    rmvnorm(n*v*n.sim, mean=rep(0,v), sigma=diag(rep(1,v))),
    c(n,v,n.sim)
  )
  # in practice we'd do this using one of the parallel processing
  # methods out there
  results <- apply( simdata, 3, function(dm) {
    z <- cerioli2010.irmcd.test( dm,
      signif.gamma=0.01 )
    # true if outliers were detected in the data, false otherwise
    any(z$outliers[,1,drop=TRUE])
  })
  # count the percentage of samples where outliers were detected;
  # should be close to the significance level value used (0.01) in these
  # samples for the intersection test
  mean(results)

## End(Not run)
```

CerioliOutlierDetection

CerioliOutlierDetection: package for implementing the Iterated Reweighted MCD outlier detection method of Cerioli (2010)

Description

Implements the outlier detection methodology of Cerioli (2010) based on Mahalanobis distances and the minimum covariance determinant (MCD) estimate of dispersion. Also provides critical values for testing outlyingness of MCD-based Mahalanobis distances using the distribution approximations developed by Hardin and Rocke (2005), Chapter 2 of Green (2017), and Green and Martin (2017).

Details

The function `cerioli2010.irmcd.test()` provides the outlier detection methodology of Cerioli (2010), and is probably the best place for a new user of this package to start. See the documentation for that function for examples.

This package was also used to produce the results presented in Chapter 2 of Green (2017) and Green and Martin (2017). There is a companion `R` package, `HardinRockeExtension`, that provides code that can be used to replicate the results of that paper. The package `HardinRockeExtension` is available from Christopher G. Green's GitHub: https://github.com/christopherggreen/HardinRockeExtensionSimulations .

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>, with advice and support from Doug Martin.

References

Andrea Cerioli. Multivariate outlier detection with high-breakdown estimators. Journal of the American Statistical Association, 105(489):147-156, 2010. doi:10.1198/jasa.2009.tm09147

C. G. Green. Applications of Robust Statistical Methods in Quantitative Finance. Dissertation, 2017. Available from https://digital.lib.washington.edu/researchworks/handle/1773/40304

C. G. Green and R. Douglas Martin. An extension of a method of Hardin and Rocke, with an application to multivariate outlier detection via the IRMCD method of Cerioli. Working Paper, 2017. Available from https://christopherggreen.github.io/papers/hr05_extension.pdf

J. Hardin and D. M. Rocke. The distribution of robust distances. Journal of Computational and Graphical Statistics, 14:928-946, 2005. doi:10.1198/106186005X77685

`ch99AsymptoticDF` *Croux and Haesbroeck (1999) finite-sample asymptotic approximation parameters for the MCD estimate*

Description

Computes the asymptotic Wishart degrees of freedom and consistency constant for the MCD robust dispersion estimate (for data with a model normal distribution) as described in Hardin and Rocke (2005) and using the formulas described in Croux and Haesbroeck (1999).

Usage

```
ch99AsymptoticDF(n.obs, p.dim, mcd.alpha)
```

Arguments

`n.obs`	(Integer) Number of observations
`p.dim`	(Integer) Dimension of the data, i.e., number of variables.
`mcd.alpha`	(Numeric) Value that determines the fraction of the sample used to compute the MCD estimate. $1 - mcd.alpha$ will be the fraction of observations that are omitted in computing the MCD estimate. The default value is $$\lfloor (n.obs + p.dim + 1)/2 \rfloor / n.obs,$$ which yields the MCD estimate with the maximum possible breakdown point.

Details

The consistency factor `c.alpha` is already available in the `robustbase` library as the function `.MCDcons`. (See the code for `covMcd`.) `ch99AsymptoticDF` uses the result of `.MCDcons` for consistency.

The computation of the asymptotic Wishart degrees of freedom parameter `m` follows the Appendix of Hardin and Rocke (2005).

Value

`c.alpha`	the asymptotic consistency coefficient for the MCD estimate of the dispersion matrix
`m.hat.asy`	the asymptotic degrees of freedom for the Wishart distribution approximation to the distribution of the MCD dispersion estimate

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

Christopher Croux and Gentiane Haesbroeck. Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. Journal of Multivariate Analysis, 71:161-190, 1999. doi:10.1006/jmva.1999.1839

J. Hardin and D. M. Rocke. The distribution of robust distances. Journal of Computational and Graphical Statistics, 14:928-946, 2005. doi:10.1198/106186005X77685

Examples

```
# compare to table from p941 of Hardin and Rocke (2005)
ch99AsymptoticDF(  50, 5)
ch99AsymptoticDF( 100,10)
ch99AsymptoticDF( 500,10)
ch99AsymptoticDF(1000,20)
```

`hr05AdjustedDF` *Adjusted Degrees of Freedom for Testing Robust Mahalanobis Distances for Outlyingness*

Description

Computes the degrees of freedom for the adjusted F distribution for testing Mahalanobis distances calculated with the minimum covariance determinant (MCD) robust dispersion estimate (for data with a model normal distribution) as described in Hardin and Rocke (2005) or in Green and Martin (2017).

Usage

```
hr05AdjustedDF( n.obs, p.dim, mcd.alpha, m.asy, method = c("HR05", "GM14"))
```

Arguments

`n.obs`	(Integer) Number of observations
`p.dim`	(Integer) Dimension of the data, i.e., number of variables.
`mcd.alpha`	(Numeric) Value that determines the fraction of the sample used to compute the MCD estimate. Default value corresponds to the maximum breakdown point case of the MCD.
`m.asy`	(Numeric) Asymptotic Wishart degrees of freedom. The default value uses `ch99AsymptoticDF` to obtain the the finite-sample asymptotic value, but the user can also provide a pre-computed value.
`method`	Either "HR05" to use the method of Hardin and Rocke (2005), or "GM14" to use the method of Green and Martin (2017).

Details

Hardin and Rocke (2005) derived an approximate F distribution for testing robust Mahalanobis distances, computed using the MCD estimate of dispersion, for outlyingness. This distribution improves upon the standard χ^2 distribution for identifying outlying points in data set. The method of Hardin and Rocke was designed to work for the maximum breakdown point case of the MCD, where

$$\alpha = \lfloor (n.obs + p.dim + 1)/2 \rfloor / n.obs.$$

Green and Martin (2017) extended this result to MCD(α), where α controls the size of the sample used to compute the MCD estimate, as well as the breakdown point of the estimator.

With argument `method = "HR05"` the function returns m_{pred} as given in Equation 3.4 of Hardin and Rocke (2005). The Hardin and Rocke method is only supported for the maximum breakdown point case; an error will be generated for other values of `mcd.alpha`.

The argument `method = "GM14"` uses the extended methodology described in Green and Martin (2017) and is available for all values of `mcd.alpha`.

Value

Returns the adjusted F degrees of freedom based on the asymptotic value, the dimension of the data, and the sample size.

Note

This function is typically not called directly by users; rather it is used in the construction of other functions.

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

C. G. Green and R. Douglas Martin. An extension of a method of Hardin and Rocke, with an application to multivariate outlier detection via the IRMCD method of Cerioli. Working Paper, 2017. Available from https://christopherggreen.github.io/papers/hr05_extension.pdf

J. Hardin and D. M. Rocke. The distribution of robust distances. Journal of Computational and Graphical Statistics, 14:928-946, 2005. doi:10.1198/106186005X77685

See Also

ch99AsymptoticDF

Examples

```
hr05tester <- function(n,p) {
a <- floor( (n+p+1)/2 )/n
hr05AdjustedDF( n, p, a, ch99AsymptoticDF(n,p,a)$m.hat.asy, method="HR05" )
}
# compare to m_pred in table on page 941 of Hardin and Rocke (2005)
```

```
hr05tester(  50, 5)
hr05tester( 100,10)
hr05tester( 500,10)
hr05tester(1000,20)

# using default arguments
hr05tester <- function(n,p) {
hr05AdjustedDF( n, p, method="HR05" )
}
# compare to m_pred in table on page 941 of Hardin and Rocke (2005)
hr05tester(  50, 5)
hr05tester( 100,10)
hr05tester( 500,10)
hr05tester(1000,20)

# Green and Martin (2017) improved method
hr05tester <- function(n,p) {
hr05AdjustedDF( n, p, method="GM14" )
}
# compare to m_sim in table on page 941 of Hardin and Rocke (2005)
hr05tester(  50, 5)
hr05tester( 100,10)
hr05tester( 500,10)
hr05tester(1000,20)
```

`hr05CriticalValue` *Hardin and Rocke (2005) Critical Value for Testing MCD-based Mahalanobis Distances*

Description

Hardin and Rocke (2005) provide an approximate F distribution for testing whether Mahalanobis distances calculated using the MCD dispersion estimate are unusually large, and hence, indicative of outliers in the data.

Usage

```
hr05CriticalValue(em, p.dim, signif.alpha)
```

Arguments

`em`	(Numeric) Degrees of freedom for Wishart distribution approximation to the MCD scatter matrix.
`p.dim`	(Integer) Dimension of the data, i.e., number of variables.
`signif.alpha`	(Numeric) Significance level for testing the null hypothesis

Details

Hardin and Rocke (2005) derived an F distributional approximation for the Mahalanobis distances of the observations that were excluded from the MCD calculation; see equation 3.2 on page 938 of the paper.

It is assumed here that the MCD covariance estimate used in the Mahalanobis distance calculation was adjusted by the consistency factor, so it is not included in the calculation here. (If one needs the consistency factor it is returned by the function `ch99AsymptoticDF` in this package or by the function `.MCDcons` in the `robustbase` package.)

Value

The appropriate cutoff value (from the F distributional approximation) for testing whether a Mahalanobis distance is unusually large at the specified significance level.

Note

It can happen that one of the F distribution paramaters, $m - p + 1$, is non-positive, in which case `qf` will return NaN. `hr05CriticalValue` will issue a warning in this case, and return NA.

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

J. Hardin and D. M. Rocke. The distribution of robust distances. Journal of Computational and Graphical Statistics, 14:928-946, 2005. doi:10.1198/106186005X77685

See Also

`hr05AdjustedDF`, `hr05CutoffMvnormal`

Examples

```
hr05CriticalValue( hr05AdjustedDF( 1000, 20 ), 20, 0.05 )
```

`hr05CutoffMvnormal` *Corrected Critical Values for Testing MCD-based Mahalanobis Distances*

Description

Provides critical values for testing for outlyingness using MCD-based Mahalanobis distances and the F distributional approximation developed by Hardin and Rocke (2005) or the enhancement by Green and Martin (2017).

Usage

```
hr05CutoffMvnormal(n.obs, p.dim, mcd.alpha, signif.alpha,
 method = c("GM14", "HR05"), use.consistency.correction = FALSE)
```

Arguments

n.obs	(Integer) Number of observations
p.dim	(Integer) Dimension of the data, i.e., number of variables.
mcd.alpha	(Numeric) Value that determines the fraction of the sample used to compute the MCD estimate. Defaults to the value used in the maximum breakdown point case of the MCD.
signif.alpha	(Numeric) Significance level for testing the null hypothesis. Default value is 0.05.
method	Either "HR05" to use the method of Hardin and Rocke (2005), or "GM14" to use the method of Green and Martin (2017).
use.consistency.correction	(Logical) By default, the method does not multiply the cutoff values by the consistency correction for the MCD, under the assumption that the correction was applied during the calculation of the MCD-based Mahalanobis distances. Specify TRUE to add the correction factor if you need it for your application.

Details

hr05CutoffMvnormal is the typical way in which a user will calculate critical values for testing outlyingness via MCD-based Mahalanobis distances. The critical values come from the F distributional approximation derived by Hardin and Rocke (2005). One can use either the corrected degrees of freedom parameter derived in that paper (which was only shown to work for the maximum breakdown point case of MCD), or the correction derived in Green and Martin (2017) for arbitrary values of mcd.alpha.

Value

cutoff.pred	Critical value based on the predicted Wishart degrees of freedom m.pred
cutoff.asy	Critical value based on the asymptotic Wishart degrees of freedom m.asy
c.alpha	The value of the consistency correction factor, c_α
m.asy	Asymptotic Wishart degrees of freedom parameter
m.pred	Predicted Wishart degrees of freedom (using the method specified in method)
n.obs	Number of observations
p.dim	Number of variables

Author(s)

Written and maintained by Christopher G. Green <christopher.g.green@gmail.com>

References

C. G. Green and R. Douglas Martin. An extension of a method of Hardin and Rocke, with an application to multivariate outlier detection via the IRMCD method of Cerioli. Working Paper, 2017. Available from https://christopherggreen.github.io/papers/hr05_extension.pdf

J. Hardin and D. M. Rocke. The distribution of robust distances. Journal of Computational and Graphical Statistics, 14:928-946, 2005. doi:10.1198/106186005X77685

See Also

hr05CriticalValue, hr05AdjustedDF

Examples

```
# examples from page 941 of Hardin and Rocke
hr05CutoffMvnormal(n.obs=50  , p.dim=5 , signif.alpha=0.05)
hr05CutoffMvnormal(n.obs=100 , p.dim=10, signif.alpha=0.05)
hr05CutoffMvnormal(n.obs=500 , p.dim=10, signif.alpha=0.05)
hr05CutoffMvnormal(n.obs=1000, p.dim=20, signif.alpha=0.05)
```

Index