

# Package ‘maxstat’

May 8, 2026

**Version** 0.7-26

**Title** Maximally Selected Rank Statistics

**Date** 2025-05-02

**Description** Maximally selected rank statistics with  
several p-value approximations.

**Depends** R ( $\geq 1.7.0$ )

**Imports** exactRankTests( $\geq 0.8-23$ ), mvtnorm( $\geq 0.5-10$ ), stats, graphics

**Suggests** TH.data, survival

**License** GPL ( $\geq 2$ )

**LazyData** yes

**NeedsCompilation** yes

**Author** Torsten Hothorn [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-8301-0471>>)

**Maintainer** Torsten Hothorn <Torsten.Hothorn@R-project.org>

**Repository** CRAN

**Date/Publication** 2025-05-02 14:30:10 UTC

## Contents

corrmsrs . . . . .	2
DLBCL . . . . .	3
hohnloser . . . . .	6
maxstat.test . . . . .	8
pexactgauss . . . . .	11
pLausen92 . . . . .	12
pLausen94 . . . . .	13
plot.maxtest . . . . .	14
pmaxstat . . . . .	15
treepipit . . . . .	16

<b>Index</b>	<b>18</b>
--------------	-----------

---

`corrmsrs`*Correlation Matrix*

---

**Description**

Correlation matrix of maximally selected rank statistics.

**Usage**

```
corrmsrs(X, minprop=0.1, maxprop=0.9)
```

**Arguments**

<code>X</code>	the vector, matrix or data.frame of prognostic factors under test.
<code>minprop</code>	at least <code>minprop*100%</code> of the observations in the first group.
<code>maxprop</code>	not more than <code>minprop*100%</code> of the observations in the first group.

**Details**

The correlations between all two-sample rank statistics induced by all possible cutpoints in `X` are computed.

**Value**

The correlation matrix with dimension depending on ties in `X` is returned.

**References**

Hothorn, T. and Lausen, B. (2003). On the Exact Distribution of Maximally Selected Rank Statistics. *Computational Statistics & Data Analysis*, **43**, 121–137.

Lausen, B., Hothorn, T., Bretz, F. and Schmacher, M. (2004). Assessment of Optimally Selected Prognostic Factors. *Biometrical Journal*, **46**(3), 364–374.

**Examples**

```
set.seed(29)

# matrix of hypothetical prognostic factors
X <- matrix(rnorm(30), ncol=3)

# this function
a <- corrmsrs(X, minprop=0, maxprop=0.999)

# coded by just typing the definition of the correlation
testcorr <- function(X) {
```

```

wh <- function(cut, x)
  which(x <= cut)
index <- function(x) {
  ux <- unique(x)
  ux <- ux[ux < max(ux)]
  lapply(ux, wh, x = x)
}
a <- unlist(test <- apply(X, 2, index), recursive=FALSE)
cnull <- rep(0, nrow(X))
mycorr <- diag(length(a))
for (i in 1:(length(a)-1)) {
  for (j in (i+1):length(a)) {
    cone <- cnull
    cone[a[[i]]] <- 1
    ctwo <- cnull
    ctwo[a[[j]]] <- 1
    sone <- sqrt(sum((cone - mean(cone))^2))
    stwo <- sqrt(sum((ctwo - mean(ctwo))^2))
    tcorr <- sum((cone - mean(cone))*(ctwo - mean(ctwo)))
    tcorr <- tcorr/(sone * stwo)
    mycorr[i,j] <- tcorr
  }
}
mycorr
}

tc <- testcorr(X)
tc <- tc + t(tc)
diag(tc) <- 1
stopifnot(all.equal(tc, a))

```

DLBCL

*Diffuse large B-cell lymphoma***Description**

A data frame with gene expression data from DLBCL (diffuse large B-cell lymphoma) patients.

**Usage**

```
data("DLBCL")
```

**Format**

DLCLid DLBCL identifier  
 GEG Gene Expression Group  
 time survival time in month  
 cens censoring: 0 censored, 1 dead

IPI International Prognostic Index

MGE Mean Gene Expression

### Source

Except of MGE, the data is published at <http://llmpp.nih.gov/lymphoma/data.shtml>. MGE is the mean of the gene expression.

### References

Ash A. Alizadeh et. al (2000), Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 504–509

### Examples

```
library("survival")

set.seed(29)

# compute the cutpoint and plot the empirical process

mod <- maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL, smethod="LogRank")

print(mod)

## Not run:
# postscript("statDLBCL.ps", horizontal=F, width=8, height=8)
# pdf("statDLBCL.pdf", width=8, height=8)

## End(Not run)
par(mai=c(1.0196235, 1.0196235, 0.8196973, 0.4198450))
plot(mod, cex.lab=1.6, cex.axis=1.6, xlab="Mean gene expression", lwd=2)
## Not run:
# dev.off()

## End(Not run)

# significance of the cutpoint
# limiting distribution

maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL,
             smethod="LogRank", pmethod="Lau92", iscores=TRUE)

# improved Bonferroni inequality, plot with significance bound

maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL,
             smethod="LogRank", pmethod="Lau94", iscores=TRUE)

mod <- maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL, smethod="LogRank",
                  pmethod="Lau94", alpha=0.05)
plot(mod, xlab="Mean gene expression")
```

```
## Not run:
# postscript(file="RNewsStat.ps",horizontal=F, width=8, height=8)
  pdf("RNewsStat.pdf", width=8, height=8)

## End(Not run)
par(mai=c(1.0196235, 1.0196235, 0.8196973, 0.4198450))
plot(mod, xlab="Mean gene expression", cex.lab=1.6, cex.axis=1.6)
## Not run:
  dev.off()

## End(Not run)

# small sample solution Hothorn & Lausen

maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL,
              smethod="LogRank", pmethod="HL")

# normal approximation

maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL,
              smethod="LogRank", pmethod="exactGauss", iscores=TRUE,
              abseps=0.01)

# conditional Monte-Carlo
maxstat.test(Surv(time, cens) ~ MGE, data=DLBCL,
              smethod="LogRank", pmethod="condMC", B = 9999)

# survival analysis and plotting like in Alizadeh et al. (2000)

splitGEG <- rep(1, nrow(DLBCL))
DLBCL <- cbind(DLBCL, splitGEG)
DLBCL$splitGEG[DLBCL$GEG == "Activated B-like"] <- 0

plot(survfit(Surv(time, cens) ~ splitGEG, data=DLBCL),
     xlab="Survival time in month", ylab="Probability")

text(90, 0.7, "GC B-like")
text(60, 0.3, "Activated B-like")

splitIPI <- rep(1, nrow(DLBCL))
DLBCL <- cbind(DLBCL, splitIPI)
DLBCL$splitIPI[DLBCL$IPI <= 2] <- 0

plot(survfit(Surv(time, cens) ~ splitIPI, data=DLBCL),
     xlab="Survival time in month", ylab="Probability")

text(90, 0.7, "Low clinical risk")
text(60, 0.25, "High clinical risk")

# survival analysis using the cutpoint

splitMGE <- rep(1, nrow(DLBCL))
```

```

DLBCL <- cbind(DLBCL, splitMGE)
DLBCL$splitMGE[DLBCL$MGE <= mod$estimate] <- 0

## Not run:
# postscript("survDLBCL.ps",horizontal=F, width=8, height=8)
# pdf("survDLBCL.pdf", width=8, height=8)

## End(Not run)
par(mai=c(1.0196235, 1.0196235, 0.8196973, 0.4198450))

plot(survfit(Surv(time, cens) ~ splitMGE, data=DLBCL),
     xlab = "Survival time in month",
     ylab="Probability", cex.lab=1.6, cex.axis=1.6, lwd=2)

text(90, 0.9, expression("Mean gene expression" > 0.186), cex=1.6)
text(90, 0.45, expression("Mean gene expression" <= 0.186 ), cex=1.6)

## Not run:
# dev.off()

## End(Not run)

```

---

hohnloser

*Left ventricular ejection fraction of patients with malignant ventricular tachyarrhythmias.*

---

## Description

A data frame with the left ventricular ejection fraction of patients with malignant ventricular tachyarrhythmias including recurrence-free month and censoring.

## Usage

```
data("hohnloser")
```

## Format

EF left ventricular ejection in percent  
 month recurrence-free month  
 cens censoring: 0 censored, 1 not censored

The data used here is published in Table 1 of Lausen and Schumacher (1992).

## Source

The data was first published by Hohnloser et al. (1987), the data used here is published in Table 1 of Lausen and Schumacher (1992).

## References

Hohnloser, S.H., Raeder, E.A., Podrid, P.J., Graboys, T.B. and Lown, B. (1987), Predictors of antiarrhythmic drug efficacy in patients with malignant ventricular tachyarrhythmias. *American Heart Journal* **114**, 1–7

Lausen, B. and Schumacher, M. (1992), Maximally Selected Rank Statistics. *Biometrics* **48**, 73–85

## Examples

```
set.seed(29)

library("survival")

# limiting distribution

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="Lau92")

# with integer valued scores for comparison

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="Lau92", iscores=TRUE)

# improved Bonferroni inequality

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="Lau94")

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="Lau94", iscores=TRUE)

# small sample solution by Hothorn & Lausen

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="HL")

# normal approximation

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="exactGauss")

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="exactGauss", iscores=TRUE)

# conditional Monte-Carlo

maxstat.test(Surv(month, cens) ~ EF, data=hohnloser,
             smethod="LogRank", pmethod="condMC", B = 9999)
```

maxstat.test

*Maximally Selected Rank and Statistics***Description**

Performs a test of independence of a response and one or more covariables using maximally selected rank statistics.

**Usage**

```
## S3 method for class 'data.frame'
maxstat.test(formula, data, subset, na.action, ...)
maxstat(y, x=NULL, weights = NULL, smethod=c("Wilcoxon", "Median",
      "NormalQuantil", "LogRank", "Data"), pmethod=c("none", "Lau92",
      "Lau94", "exactGauss", "HL", "condMC", "min"), iscores=(pmethod=="HL"),
      minprop = 0.1, maxprop=0.9, alpha = NULL, keepxy=TRUE, ...)
```

**Arguments**

y	numeric vector of data values, dependent variable.
x	numeric vector of data values, independent variable.
weights	an optional numeric vector of non-negative weights, summing to the number of observations.
smethod	kind of statistic to be computed, i.e. defines the scores to be used for computing the statistic.
pmethod	kind of p-value approximation to be used.
iscores	logical: should the scores be mapped into integers 1:length(x)? This is TRUE by default for pmethod=="HL" and FALSE otherwise.
minprop	at least minprop*100% of the observations in the first group.
maxprop	not more than minprop*100% of the observations in the first group.
alpha	significance niveau, the appropriate quantile is computed if alpha is specified. Used for plotting within <a href="#">plot.maxtest</a> .
keepxy	logical: return y and x as elements of the maxtest object.
formula	a formula describing the model to be tested of the form lhs ~ rhs where lhs is the response variable. For survival problems, i.e. using the log-rank statistic, the formula is of the form <code>Surv(time, event) ~ rhs</code> , see above.
data	an data frame containing the variables in the model formula. data is required.
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to <code>getOption("na.action")</code> .
...	additional parameters to be passed to <a href="#">pmvnorm</a> or B, an integer defining the number of Monte-Carlo replications.

## Details

The assessment of the predictive power of a variable  $x$  for a dependent variable  $y$  can be determined by a maximally selected rank statistic.

`smethod` determines the kind of statistic to be used. `Wilcoxon` and `Median` denote maximally selected Wilcoxon and Median statistics. `NormalQuantile` and `LogRank` denote v.d. Waerden and log-rank scores.

`pmethod` specifies which kind of approximation of the p-value should be used. `Lau92` is the limiting distribution by a Brownian bridge (see Lausen and Schumacher, 1992, and [pLausen92](#)), `Lau94` the approximation based on an improved Bonferroni inequality (see Lausen, Sauerbrei and Schumacher, 1994, and [pLausen94](#)).

`exactGauss` returns the exact p-value for a maximally selected Gauss statistic, see Hothorn and Lausen (2003).

`HL` is a small sample approximation based on the Streitberg-Röhmel algorithm (see [pperm](#)) introduced by Hothorn and Lausen (2003). This requires integer valued scores. For v. d. Waerden and Log-rank scores we try to find integer valued scores having the same shape. This results in slightly different scores (and a different test), the procedure is described in Hothorn (2001) and Hothorn and Lausen (2003).

All the approximations are known to be conservative, so `min` gives the minimum p-value of all procedures.

`condMC` simulates the distribution via conditional Monte-Carlo.

For survival problems, i.e. using a maximally selected log-rank statistic, the interface is similar to [survfit](#). The depended variable is a survival object `Surv(time, event)`. The argument `event` may be a numeric vector of 0 (alive) and 1 (dead) or a vector of logicals with `TRUE` indicating death. If more than one covariable is specified in the right hand side of `formula` (or if  $x$  is a matrix or data frame), the variable with smallest p-value is selected and the p-value for the global test problem of independence of  $y$  and every variable on the right hand side is returned (see Lausen et al., 2002).

## Value

An object of class `maxtest` or `mmaxtest` (if more than one covariable was specified) containing the following components is returned:

<code>statistic</code>	the value of the test statistic.
<code>p.value</code>	the p-value for the test.
<code>smethod</code>	the type of test applied.
<code>pmethod</code>	the type of p-value approximation applied.
<code>estimate</code>	the estimated cutpoint (of $x$ ) which separates $y$ best. For numeric data, the groups are defined by $x$ less or equal to <code>estimate</code> and $x$ greater estimate.
<code>maxstats</code>	a list of <code>maxtest</code> objects, one for each covariable.
<code>whichmin</code>	an integer specifying the element of <code>maxstats</code> with smallest p-value.
<code>p.value</code>	the p-value of the global test.
<code>univp.values</code>	the p-values for each of the variables under test.
<code>cm</code>	the correlation matrix the p-value is based on.

`plot.maxtest` and `print.maxtest` can be used for plotting and printing. If `keepxy = TRUE`, there are elements `y` and `x` giving the response and independent variable.

## References

- Hothorn, T. and Lausen, B. (2003). On the Exact Distribution of Maximally Selected Rank Statistics. *Computational Statistics & Data Analysis*, **43**, 121–137.
- Lausen, B. and Schumacher, M. (1992). Maximally Selected Rank Statistics. *Biometrics*, **48**, 73–85
- Lausen, B., Sauerbrei, W. and Schumacher, M. (1994). Classification and Regression Trees (CART) used for the exploration of prognostic factors measured on different scales. in: P. Dirschedl and R. Ostermann (Eds), *Computational Statistics*, Heidelberg, Physica-Verlag, 483–496
- Hothorn, T. (2001). On Exact Rank Tests in R. *R News*, **1**, 11–12
- Lausen, B., Hothorn, T., Bretz, F. and Schumacher, M. (2004). Assessment of Optimally Selected Prognostic Factors. *Biometrical Journal*, **46**(3), 364–374.

## Examples

```
set.seed(29)

x <- sort(runif(20))
y <- c(rnorm(10), rnorm(10, 2))
mydata <- data.frame(cbind(x,y))

mod <- maxstat.test(y ~ x, data=mydata, smethod="Wilcoxon", pmethod="HL",
                    minprop=0.25, maxprop=0.75, alpha=0.05)

print(mod)
plot(mod)

# adjusted for more than one prognostic factor.
library("survival")
mstat <- maxstat.test(Surv(time, cens) ~ IPI + MGE, data=DLBCL,
                     smethod="LogRank", pmethod="exactGauss",
                     abseps=0.01)

plot(mstat)

### sphase
set.seed(29)
data("sphase", package = "TH.data")

maxstat.test(Surv(RFS, event) ~ SPF, data=sphase, smethod="LogRank",
             pmethod="Lau94")

maxstat.test(Surv(RFS, event) ~ SPF, data=sphase, smethod="LogRank",
             pmethod="Lau94", iscores=TRUE)

maxstat.test(Surv(RFS, event) ~ SPF, data=sphase, smethod="LogRank",
             pmethod="HL")

maxstat.test(Surv(RFS, event) ~ SPF, data=sphase, smethod="LogRank",
             pmethod="condMC", B = 9999)

plot(maxstat.test(Surv(RFS, event) ~ SPF, data=sphase, smethod="LogRank"))
```

**Description**

Computes the exact probability that a maximally selected gauss statistic is greater or equal to b.

**Usage**

```
pexactgauss(b, x, minprop=0.1, maxprop=0.9, ...)  
qexactgauss(p, x, minprop=0.1, maxprop=0.9, ...)
```

**Arguments**

b	quantile.
p	probability.
x	the prognostic factor(s) under test.
minprop	at least minprop*100% of the observations in the first group.
maxprop	not more than minprop*100% of the observations in the first group.
...	additional parameters to be passed to <a href="#">pmvnorm</a> .

**Details**

This is the exact distribution of a maximally selected Gauss statistic and the asymptotic distribution for maximally selected rank statistics. Normal probabilities are derived from the procedures by Genz/Bretz (see [pmvnorm](#) for details).

**Value**

The probability that, under the hypothesis of independence, a maximally selected gauss statistic greater equal b is observed.

**References**

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, **1**, 141–150

Genz, A. (1993). Comparison of methods for the computation of multivariate normal probabilities. *Computing Science and Statistics*, **25**, 400–405

**Examples**

```
set.seed(29)  
  
x <- rnorm(20)  
  
pexact <- pexactgauss(2.5, x, abseps=0.01)
```

**Description**

Approximates the probability that a maximally selected rank statistic is greater or equal to  $b$ .

**Usage**

```
pLausen92(b, minprop=0.1, maxprop=0.9)
qLausen92(p, minprop=0.1, maxprop=0.9)
```

**Arguments**

<code>b</code>	quantile.
<code>p</code>	probability.
<code>minprop</code>	at least <code>minprop*100%</code> of the observations in the first group.
<code>maxprop</code>	not more than <code>minprop*100%</code> of the observations in the first group.

**Details**

Large sample approximation by Miller and Siegmund (1982) based on a Brownian bridge, cf. Lausen and Schumacher (1992).

**Value**

The probability that, under the hypothesis of independence, a maximally selected statistic greater equal  $b$  is observed.

**References**

Miller, R. and Siegmund, D. (1982), Maximally Selected Chi Square Statistics. *Biometrics*, **38**, 1011–1016

Lausen, B. and Schumacher, M. (1992), Maximally Selected Rank Statistics. *Biometrics*, **48**, 73–85

**Examples**

```
# Compute quantiles. Should be equal to Table 2 in Lausen and Schumacher

load(file.path(system.file(package = "maxstat"), "results", "LausenTab2.rda"))

a <- rev(c(0.01, 0.025, 0.05, 0.1))
prop <- rbind(c(0.25, 0.75), c(0.4, 0.6), c(0.1, 0.9), c(0.4, 0.9))
Quant <- matrix(rep(0, length(a)*nrow(prop)), nrow=length(a))

for (i in 1:length(a)) {
  for (j in 1:nrow(prop)) {
```

```

    Quant[i,j] <- qLausen92(a[i], minprop=prop[j,1], maxprop=prop[j,2])
  }
}

Quant <- round(Quant, 3)
rownames(Quant) <- a
colnames(Quant) <- c("A2575", "A46", "A19", "A49")
Quant <- as.data.frame(Quant)
rownames(LausenTab2) <- a

Quant

LausenTab2

if(!all.equal(LausenTab2, Quant)) stop("error checking pLausen92")

```

---

pLausen94

*Approximating Maximally Selected Statistics*


---

## Description

Approximates the probability that a maximally selected rank statistic is greater or equal to b.

## Usage

```

pLausen94(b, N, minprop=0.1, maxprop=0.9, m=NULL)
qLausen94(p, N, minprop=0.1, maxprop=0.9, m=NULL)

```

## Arguments

b	quantile.
p	probability.
N	number of observations.
minprop	at least minprop*100% of the observations in the first group.
maxprop	not more than minprop*100% of the observations in the first group.
m	a integer vector containing the sample sizes in the first groups for each cutpoint considered. If is.null(m) a continuous predictor is assumed.

## Details

Approximation based on an improved Bonferroni inequality.

## Value

The probability that, under the hypothesis of independence, a maximally selected statistic greater equal b is observed.

## References

- Worsley, K.J. (1982), An Improved Bonferroni Inequality and Applications. *Biometrika*, **69**, 297–302
- Lausen, B. (1990), Maximal Selektierte Rangstatistiken. Dissertation. Universität Dortmund
- Lausen, B., Sauerbrei, W. & Schumacher, M. (1994). Classification and Regression Trees (CART) used for the exploration of prognostic factors measured on different scales. in: P. Dirschedl & R. Ostermann (Eds), *Computational Statistics*, Heidelberg, Physica-Verlag, 483–496

## Examples

```
p <- pLausen94(2.5, 20, 0.25, 0.75)

# Lausen 94, page 489

if (round(p, 3) != 0.073) stop("error checking pLausen94")

# the same

p2 <- pLausen94(2.5, 200, 0.25, 0.75, m=seq(from=50, to=150, by=10))

stopifnot(all.equal(round(p,3), round(p2,3)))
```

---

plot.maxtest

*Print and Plot Standardized Statistics*

---

## Description

Printing and plotting method of objects of class maxtest

## Usage

```
## S3 method for class 'maxtest'
plot(x, xlab=NULL, ylab=NULL, ...)
## S3 method for class 'maxtest'
print(x, digits = getOption("digits"), ...)
## S3 method for class 'mmaxtest'
plot(x, xlab=NULL, ylab=NULL, nrow=2, ...)
## S3 method for class 'mmaxtest'
print(x, digits = getOption("digits"), ...)
```

## Arguments

x	an object of class maxtest or mmaxtest.
xlab	label of x-axis.
ylab	label of y-axis.

nrow            number of rows for multiple plots at one page.  
 digits         number of significant digits to be printed.  
 ...             additional arguments to plot or print.htest.

### Details

The standardized statistics are plotted. If alpha was given in `maxstat.test` the appropriate significance bound is plotted as a red line. `print.maxtest` is just a wrapper to `print.htest`.

### Examples

```
set.seed(29)

x <- sort(runif(20))
y <- rbinom(20, 1, 0.5)
mydata <- data.frame(c(x,y))

mod <- maxstat.test(y ~ x, data=mydata, smethod="Median",
                    pmethod="HL", alpha=0.05)
print(mod)
plot(mod)
```

---

pmaxstat

*Approximating Maximally Selected Statistics*

---

### Description

Approximates the probability that a maximally selected rank statistic is greater or equal to b.

### Usage

```
pmaxstat(b, scores, msample, quant=FALSE)
qmaxstat(p, scores, msample)
```

### Arguments

b                quantile.  
 p                propability.  
 scores         integer valued scores assigned to the observations.  
 msample        all possible splitpoints.  
 quant         logical. Returns the results of SR instead of P-values. Only to be used in qmaxstat.

### Details

Small sample approximation by Hothorn and Lausen (2003).

**Value**

An upper limit for the probability that, under the hypothesis of independence, a maximally selected statistic greater equal  $b$  is observed. `qmaxstat` needs optimization.

**References**

Hothorn, T. and Lausen, B. (2003). On the Exact Distribution of Maximally Selected Rank Statistics. *Computational Statistics & Data Analysis*, **43**, 121–137.

**Examples**

```
pmaxstat(2.5, 1:20, 5:15)
```

---

treepipit

*Tree Pipit Data*

---

**Description**

Counts of tree pipits at 86 raster points in oak forests.

**Usage**

```
data("treepipit")
```

**Format**

A data frame with 86 observations on the following 2 variables.

**counts** number of tree pipits counted.

**coverstorey** canopy overstorey in percent.

**Details**

The influence of canopy overstorey on the number of bird individuals is of special interest.

**Source**

Data collected and kindly provided by Joerg Mueller <mue@lwf.uni-muenchen.de>.

**References**

Mueller J. and Hothorn T. (2004), Maximally Selected Two-sample Statistics as a New Tool for the Identification and Assessment of Habitat Factors with an Application to Breeding-bird Communities in Oak Forests, *European Journal of Forest Research*, **123**(3), 219–228.

**Examples**

```
mod <- maxstat.test(counts ~ coverstorey, data = treepipit,  
                    smethod = "Data", pmethod = "HL", minprop = 0.2,  
                    maxprop = 0.8)  
  
print(mod)  
plot(mod)
```

# Index

- \* **datasets**
  - DLBCL, [3](#)
  - hohnloser, [6](#)
  - treepipit, [16](#)
- \* **distribution**
  - pexactgauss, [11](#)
  - pLausen92, [12](#)
  - pLausen94, [13](#)
  - pmaxstat, [15](#)
- \* **htest**
  - maxstat.test, [8](#)
  - plot.maxtest, [14](#)
- \* **misc**
  - corrmsrs, [2](#)

corrmsrs, [2](#)

DLBCL, [3](#)

hohnloser, [6](#)

maxstat (maxstat.test), [8](#)  
maxstat.test, [8](#), [15](#)

pexactgauss, [11](#)  
pLausen92, [9](#), [12](#)  
pLausen94, [9](#), [13](#)  
plot.maxtest, [8](#), [9](#), [14](#)  
plot.mmaxtest (plot.maxtest), [14](#)  
pmaxstat, [15](#)  
pmvnorm, [8](#), [11](#)  
pperm, [9](#)  
print.maxtest, [9](#)  
print.maxtest (plot.maxtest), [14](#)  
print.mmaxtest (plot.maxtest), [14](#)

qexactgauss (pexactgauss), [11](#)  
qLausen92 (pLausen92), [12](#)  
qLausen94 (pLausen94), [13](#)  
qmaxstat (pmaxstat), [15](#)

survfit, [9](#)

treepipit, [16](#)