Package ‘st’
July 8, 2015

Version 1.2.5
Date 2015-07-08
Title Shrinkage t Statistic and Correlation-Adjusted t-Score
Author Rainer Opgen-Rhein, Verena Zuber, and Korbinian Strimmer.
Maintainer Korbinian Strimmer <strimmerlab@gmail.com>
Depends R (>= 3.0.2), sda (>= 1.3.7), fdrtool (>= 1.2.15), corpcor (>= 1.6.8)
Suggests limma, samr
Imports graphics, stats
Description Implements the \"shrinkage t\" statistic
    introduced in Opgen-Rhein and Strimmer (2007) and a shrinkage estimate
    of the \"correlation-adjusted t-score\" (CAT score) described in
    Zuber and Strimmer (2009). It also offers a convenient interface
    to a number of other regularized t-statistics commonly
    employed in high-dimensional case-control studies.
License GPL (>= 3)
URL http://strimmerlab.org/software/st/
NeedsCompilation no
Repository CRAN
Date/Publication 2015-07-08 22:37:28

R topics documented:

- st-package ......................................................... 2
- choedata ........................................................... 2
- cst.stat ............................................................ 3
- diffmean.stat ...................................................... 5
- lait.stat ............................................................ 6
- regularizedt ....................................................... 8
- shrinkcat.stat ..................................................... 11
- shrinkt.stat ...................................................... 14
- studentt.stat ...................................................... 16
The st package

Description

This package implements the "shrinkage t" statistic described in Opgen-Rhein and Strimmer (2007) and a shrinkage estimate of the "correlation-adjusted t-score" (cat score) introduced in Zuber and Strimmer (2009). It also offers a convenient interface to a number of other regularized t-statistics commonly employed in high-dimensional case-control studies.

Author(s)

Rainer Opgen-Rhein, Verena Zuber, and Korbinian Strimmer (http://strimmerlab.org/)

References

See website: http://strimmerlab.org/software/st/

See Also

shrinkt.stat, shrinkcat.stat, studentt.stat, modt.stat, cst.stat, lait.stat.

choedata A Subset of the Choe et al. (2005) “Golden Spike” Experiment

Description

These data are expression levels for a subset of the genes investigated in the Choe et al. (2005) “Golden Spike” Affymetrix case-control experiment.

From the original data the 2,535 probe sets for spike-ins with ratio 1:1 were removed, leaving in total 11,475 genes with 3 replicates per group, and 1,331 known differentially expressed genes.

Usage

data(choedata)

Format

choe2.mat is a matrix of dimension 6 times 11,475. It contains the samples in its rows and the genes in its columns.

choe2.L describes the case control-structure of the experiment, and choe2.degenes indicates the known differentially expressed genes. choe2.symbol.name, choe2.probe.name, and choe2.mapping provide additional information on the investigated genes.
References


Examples

```r
# load st library
library("st")

# load data set
data(choedata)

# 6 samples, 11,475 genes
dim(choeRNmat)

# two groups (case vs. control
choeRNl

# 1,331 differentially expressed genes
sum(choeRNdegenes)

# further information on genes
choeRNsymbol
choeRNname
choeRNprobe
choeRNmapping
```

cst.stat  

**Correlation-Shared t-Statistic**

Description

shrinkcat.stat and shrinkcat.fun compute the “correlation-shared” t-statistic of Tibshirani and Wassermann (2006).

Usage

```r
cst.stat(X, L, verbose=TRUE)
cst.fun(L, verbose=TRUE)
```

Arguments

- **X**  
data matrix. Note that the *columns* correspond to variables (“genes”) and the *rows* to samples.

- **L**  
vector with class labels for the two groups.

- **verbose**  
print out some (more or less useful) information during computation.
Details

The correlation-shared t-statistic for a gene is computed as the average of t-scores correlated with that gene. For mathematical details see Tibshirani and Wasserman (2006).

Value

cst.stat returns a vector containing correlation-shared t-statistic for each variable/gene.
The corresponding cst.fun functions return a function that computes the correlation-shared t-statistic when applied to a data matrix (this is very useful for simulations).

Author(s)

Korbinian Strimmer (http://strimmerlab.org).

References


See Also

shrinkcat.stat, lait.stat.

Examples

```r
# load st library
library("st")

# prostate data set
data(singh2002)
X = singh2002$x
L = singh2002$y

dim(X) # 102 6033
length(L) # 102

# correlation shared t statistic
## Not run:
score = cst.stat(X, L)
idx = order(abs(score), decreasing=TRUE)
idx[1:10]
# [1] 610 1720 364 332 914 3940 4546 1068 579 4331

## End(Not run)

# compared with:

# Student t statistic
score = studentt.stat(X, L)
idx = order(abs(score), decreasing=TRUE)
idx[1:10]
```
diffmean.stat  Difference of Means ("Fold Change") and Rank Products Statistic

Description
These function compute the difference of group means ("fold change") and the related rank products statistic of Breitling et al. (2004).

Usage
```
diffmean.stat(X, L)
diffmean.fun(L)
rankprod.stat(X, L)
rankprod.fun(L)
```

Arguments
- `X` data matrix. Note that the columns correspond to variables ("genes") and the rows to samples.
- `L` factor containing class labels for the two groups.

Details
- `diffmean.*` computes the difference of means (i.e. the fold-change for log-transformed data).
- `rankprod.*` computes the two-sided rank products statistic, i.e. the geometric mean of the ranks of the pairwise absolute mean differences (Breitling et al. 2004). Note that for consistency with the other functions in this package the complement of the averaged ranks is returned (i.e. rank 1 becomes `ncol(X)`, rank 2 becomes `ncol(X)-1`, etc.).

Value
The *.stat functions directly return the respective statistic for each variable.
The corresponding *.fun functions return a function that produces the respective statistics when applied to a data matrix (this is very useful for simulations).

Author(s)
Korbinian Strimmer [http://strimmerlab.org].
This function is in part based on code from Henry Wirth.
References


See Also

studentt.stat, shrinkt.stat.

Examples

# load st library
library("st")

# load Choe et al. (2005) data
data(choedata)
X <- choeRNmat
dim(X) # 6 11475
L <- choeRNlabels
L

# L may also contain some real labels
L = c("group 1", "group 1", "group 1", "group 2", "group 2", "group 2")

# difference of means resp. fold change statistic
score = diffmean.stat(X, L)
order(abs(score), decreasing=TRUE)[1:10]
# [1] 4790 6620 1022 10979 970 35 2693 5762 5885 2

# two-sided rank products statistic
score = rankprod.stat(X, L)
order(score, decreasing=TRUE)[1:10]
# [1] 4790 1022 10979 6620 35 2693 970 5762 5885 2

---

lait.stat  Correlation-Predicted t-Statistic

Description

lait.stat, laicat.fun, and lai.tscore compute the “correlation-predicted” t-statistic of Lai (2008).

Usage

lait.stat(X, L, f=0.2, verbose=TRUE)
lait.fun(L, f=0.2, verbose=TRUE)
lai.tscore(gene, tscore, corr, f=0.2, plot=FALSE)
Arguments

- **X**: data matrix. Note that the columns correspond to variables ("genes") and the rows to samples.
- **L**: vector with class labels for the two groups.
- **verbose**: print out some (more or less useful) information during computation.
- **f**: smoother span used in `lowess` (default value: 0.2)
- **gene**: the gene for which the Lai t-score is computed
- **tscore**: a vector with t-scores
- **corr**: a matrix containing correlations
- **plot**: show scatter plot correlations versus t-scores with predicted t-score

Details

The correlation-predicted t-statistic for a gene is the t-score predicted by local linear regression using all other genes. For mathematical details see Lai (2008).

Value

- `lait.stat` returns a vector containing correlation-predicted t-statistic for each variable/gene.
- The corresponding `lait.fun` functions return a function that computes the correlation-shared t-statistic when applied to a data matrix (this is very useful for simulations).
- The function `lai.tscore` allows to compute the correlation-predicted t-statistic for a gene given a correlation matrix and a vector of t-statistics.

Author(s)


References


See Also

`shrinkcat.stat`, `cst.stat`.

Examples

```r
# load stat library
go
library("st")

# prostate data set
go
data(singh2002)
X = singh2002$x
g = singh2002$y
```
dim(x)  # 102 6033
length(L)  # 102

# compute correlation-predicted t-score for various choices
# of smoothing span

## Not run:

score1 = lait.stat(X, L, f=0.1)
idx1 = order(abs(score1), decreasing=TRUE)
idx1[1:10]
# 1072 297 1130 4495 4523 4401 1089 955 373 3848

score3 = lait.stat(X, L, f=0.3)
idx3 = order(abs(score3), decreasing=TRUE)
idx3[1:10]
# 1130 962 1688 1223 583 1118 955 297 698 1219

score5 = lait.stat(X, L, f=0.5)
idx5 = order(abs(score5), decreasing=TRUE)
idx5[1:10]
# 698 962 1223 1219 739 1172 583 694 3785 3370

score7 = lait.stat(X, L, f=0.7)
idx7 = order(abs(score7), decreasing=TRUE)
idx7[1:10]
# 698 739 1219 962 3785 725 694 735 3370 1172

# pick the one with highest correlation to Student t score
t = studentt.stat(X, L)
cor(t, score1, method="spearman")  # 0.4265832
cor(t, score3, method="spearman")  # 0.471273
cor(t, score5, method="spearman")  # 0.4750564
cor(t, score7, method="spearman")  # 0.4666669

# focus on gene 19
t = studentt.stat(X, L)
R = cor(centroids(X, L, lambda.var=0, centered.data=TRUE,
        verbose=TRUE)$centered.data)
lai.tscore(gene=19, t, R, f=0.5, plot=TRUE)

## End(Not run)
Description

These functions provide a simple interface to a variety of (regularized) t statistics that are commonly used in the analysis of high-dimensional case-control studies.

Usage

```r
efront.stat(X, L, verbose=TRUE)
efront.fun(L, verbose=TRUE)
sam.stat(X, L)
sam.fun(L)
samL1.stat(X, L, method=c("lowess", "cor"), plot=FALSE, verbose=TRUE)
samL1.fun(L, method=c("lowess", "cor"), plot=FALSE, verbose=TRUE)
modt.stat(X, L)
modt.fun(L)
```

Arguments

- **X**: data matrix. Note that the columns correspond to variables ("genes") and the rows to samples.
- **L**: factor containing class labels for the two groups.
- **method**: determines how the smoothing parameter is estimated (applies only to improved SAM statistic `samL1`).
- **plot**: output diagnostic plot (applies only to improved SAM statistic `samL1`).
- **verbose**: print out some (more or less useful) information during computation.

Details

`efront.*` computes the t statistic using the 90 % rule of Efron et al. (2001).

`sam.*` computes the SAM t statistic of Tusher et al. (2001). Note that this requires the additional installation of the "samr" package.

`samL1.*` computes the improved SAM t statistic of Wu (2005). Note that part of the code in this function is based on the R code provided by B. Wu.

`modt.*` computes the moderated t statistic of Smyth (2004). Note that this requires the additional installation of the "limma" package.

All the above statistics are compared relative to each other and relative to the shrinkage t statistic in Opgen-Rhein and Strimmer (2007).

Value

The *.stat functions directly return the respective statistic for each variable.

The corresponding *.fun functions return a function that produces the respective statistics when applied to a data matrix (this is very useful for simulations).

Author(s)

References


See Also
diffmean.stat, studentt.stat, shrinkt.stat, shrinkcat.stat.

Examples

# load st library
library("st")

# load Choe et al. (2005) data
data(choedata)
X <- choe2.mat
dim(X) # 6 11475
L <- choe2.L
L

# L may also contain some real labels
L = c("group 1", "group 1", "group 1", "group 2", "group 2", "group 2")

# Efron t statistic (90 % rule)
score = efront.stat(X, L)
order(score^2, decreasing=TRUE)[1:10]
# [1] 4790 10979 11068 1022 50 724 5762 43 10936 9939

# sam statistic
# (requires "samr" package)
#score = sam.stat(X, L)
#order(score^2, decreasing=TRUE)[1:10]
# [1] 4790 10979 1022 5762 35 970 50 11068 10905 2693

# improved sam statistic
#score = samI1.stat(X, L)
#order(score^2, decreasing=TRUE)[1:10]
# [1] 1 2 3 4 5 6 7 8 9 10
# here all scores are zero!

# moderated t statistic
# (requires "limma" package)
#score = modt.stat(X, L)
#order(score^2, decreasing=TRUE)[1:10]
# [1] 4790 10979 1022 5762 35 50 11068 970 10905 43

# shrinkage t statistic
score = shrinkt.stat(X, L)
order(score^2, decreasing=TRUE)[1:10]
# [1] 10979 11068 50 1022 724 5762 43 10979 10936 9939
**Description**

`shrinkcat.stat` and `shrinkcat.fun` compute a shrinkage estimate of the “correlation-adjusted t score” of Zuber and Strimmer (2009).

**Usage**

```r
shrinkcat.stat(x, L, lambda, lambda.var, lambda.freqs, var.equal=TRUE, paired=FALSE, verbose=TRUE)
shrinkcat.fun(L, lambda, lambda.var, lambda.freqs, var.equal=TRUE, verbose=TRUE)
```

**Arguments**

- `x` data matrix. Note that the columns correspond to variables (“genes”) and the rows to samples.
- `L` factor with class labels for the two groups. If only a single label is given then a one-sample CAT score against 0 is computed.
- `lambda` Shrinkage intensity for the correlation matrix. If not specified it is estimated from the data. `lambda=0` implies no shrinkage and `lambda=1` complete shrinkage.
- `lambda.var` Shrinkage intensity for the variances. If not specified it is estimated from the data. `lambda.var=0` implies no shrinkage and `lambda.var=1` complete shrinkage.
- `lambda.freqs` Shrinkage intensity for the frequencies. If not specified it is estimated from the data. `lambda.freqs=0` implies no shrinkage (i.e. empirical frequencies).
- `var.equal` assume equal (default) or unequal variances in each group.
- `paired` compute paired CAT score (default is to use unpaired CAT score).
- `verbose` print out some (more or less useful) information during computation.

**Details**

The CAT (“correlation-adjusted t”) score is the product of the square root of the inverse correlation matrix with a vector of t scores. The CAT score thus describes the contribution of each individual feature in separating the two groups, after removing the effect of all other features.

In Zuber and Strimmer (2009) it is shown that the CAT score is a natural criterion to rank features in the presence of correlation. If there is no correlation, the CAT score reduces to the usual t score (hence in this case the estimate from `shrinkcat.stat` equals that from `shrinkt.stat`).

The function `catscore` implements multi-class CAT scores.
shrinkcat.stat

Value

shrinkcat.stat returns a vector containing a shrinkage estimate of the "CAT score" for each variable/gene.

The corresponding shrinkcat.fun functions return a function that computes the cat score when applied to a data matrix (this is very useful for simulations).

The scale factor in the "shrinkage CAT" statistic is computed from the estimated frequencies (to use the standard empirical scale factor set lambda.freqs=0).

Author(s)


References


See Also
catscore, shrinkt.stat, cst.stat, lait.stat.

Examples

# load st library
library("st")

# prostate data set
data(singh2002)
X = singh2002$x
L = singh2002$y

dim(X)  # 102 6033
length(L)  # 102

# shrinkage cat statistic
score = shrinkcat.stat(X, L)
idx = order(score^2, decreasing=TRUE)
idx[1:10]
# 610 364 1720 3647 3375 332 3282 3991 1557 914

# compute q-values and local false discovery rates
library("fdrtool")
fdr.out = fdrtool(as.vector(score))
sum(fdr.out$qval < 0.05)
sum(fdr.out$lfdr < 0.2)

# compared with:
shrinkcat.stat

# shrinkage t statistic
score = shrinkt.stat(X, L)
idx = order(score^2, decreasing=TRUE)
idx[1:10]
# 610 1720 3940 914 364 332 3647 4331 579 1068

# shrinkage CAT score with zero correlation among predictors
# is the same as shrinkage t
score2 = shrinkcat.stat(X, L, lambda=1)
sum((score-score)^2)

# Student t statistic
score = studentt.stat(X, L)
idx = order(score^2, decreasing=TRUE)
idx[1:10]
# 610 1720 364 332 914 3940 4546 1068 579 4331

# shrinkage CAT score with zero correlation and no shrinkage
# is the same as student t
score2 = shrinkcat.stat(X, L, lambda=1, lambda.var=0, lambda.freqs=0, verbose=FALSE)
sum((score-score)^2)

# difference of means ("Fold Change")
score = diffmean.stat(X, L)
idx = order(abs(score), decreasing=TRUE)
idx[1:10]
# 735 610 694 298 698 292 739 3940 702 721

## paired CAT score

# we drop two cancer cases to make samples size equal in
# the two groups to allow to compute paired statistic
X = X[1:100,]
L = L[1:100]
sum(L="cancer") # 50
sum(L="healthy") # 50

# paired shrinkage CAT score
scat.paired = shrinkcat.stat(X, L, paired=TRUE)

# for zero correlation the paired shrinkage CAT score
# reduces to the paired shrinkage t score
score = shrinkt.stat(X, L, paired=TRUE, verbose=FALSE)
score2 = shrinkcat.stat(X, L, lambda=1, paired=TRUE, verbose=FALSE)
sum((score-score2)^2)
shrinkt.stat  The Shrinkage t Statistic

Description

shrinkt.stat and shrinkt.fun compute the “shrinkage t” statistic of Opgen-Rhein and Strimmer (2007).

Usage

shrinkt.stat(X, L, lambda.var, lambda.freqs, var.equal=TRUE, paired=FALSE, verbose=TRUE)
shrinkt.fun(L, lambda.var, lambda.freqs, var.equal=TRUE, verbose=TRUE)

Arguments

X    data matrix. Note that the columns correspond to variables (“genes”) and the rows to samples.
L    factor with class labels for the two groups. If only a single label is given then a one-sample t-score against 0 is computed.
lambda.var    Shrinkage intensity for the variances. If not specified it is estimated from the data. lambda.var=0 implies no shrinkage and lambda.var=1 complete shrinkage.
lambda.freqs    Shrinkage intensity for the frequencies. If not specified it is estimated from the data. lambda.freqs=0 implies no shrinkage (i.e. empirical frequencies).
var.equal    assume equal (default) or unequal variances in each group.
paired    compute paired t-score (default is to use unpaired t-score).
verbose    print out some (more or less useful) information during computation.

Details

The “shrinkage t” statistic is similar to the usual t statistic, with the replacement of the sample variances by corresponding shrinkage estimates. These are derived in a distribution-free fashion and with little a priori assumptions. Using the “shrinkage t” statistic produces highly accurate rankings - see Opgen-Rhein and Strimmer (2007).

The “shrinkage t” statistic can be generalized to include gene-wise correlation, see shrinkcat.stat.

The scale factor in the ”shrinkage t” statistic is computed from the estimated frequencies (to use the standard empirical scale factor set lambda.freqs=0).

Value

shrinkt.stat returns a vector containing the “shrinkage t” statistic for each variable/gene.

The corresponding shrinkt.fun functions return a function that produces the “shrinkage t” statistics when applied to a data matrix (this is very useful for simulations).
Author(s)


References

Opgen-Rhein, R., and K. Strimmer. 2007. Accurate ranking of differentially expressed genes by a
10.2202/1544-6115.1252)

See Also

studentt.stat, diffmean.stat, shrinkcat.stat.

Examples

# load st library
library("st")

# load Choe et al. (2005) data
data(choedata)
X <- choeRNAmat
dim(X) # 6 11475
L <- choeRNl
L

# L may also contain some real labels
L = c("group 1", "group 1", "group 1", "group 2", "group 2", "group 2")

# shrinkage t statistic (equal variances)
score = shrinkt.stat(X, L)
order(score, decreasing=TRUE)[1:10]

# [1] 10979 11068 50 1022 724 5762 43 4790 10936 9939
# lambda.var (variances): 0.3882
# lambda.freqs (frequencies): 1

# shrinkage t statistic (unequal variances)
score = shrinkt.stat(X, L, var.equal=FALSE)
order(score^2, decreasing=TRUE)[1:10]

# [1] 11068 50 10979 724 43 1022 5762 10936 9939 9769
# lambda.var class #1 and class #2 (variances): 0.3673 0.3362
# lambda.freqs (frequencies): 1

# compute q-values and local false discovery rates
library("fdrtool")
fdr.out = fdrtool(score)
sum( fdr.out$qval < 0.05 )
sum( fdr.out$lfdr < 0.2 )
fdr.out$param
# computation of paired t-score

# paired shrinkage t statistic
score = shrinkt.stat(X, L, paired=TRUE)
order(score*2, decreasing=TRUE)[1:10]
# [1] 50 4790 5393 11868 5762 18238 9939 708 728 68

# if there is no shrinkage the paired shrinkage t score reduces
# to the conventional paired student t statistic
score = studentt.stat(X, L, paired=TRUE)
score2 = shrinkt.stat(X, L, lambda.var=0, lambda.freqs=0, paired=TRUE, verbose=FALSE)
sum((score-score2)^2)

---

**studentt.stat**

(Paired) Student t Statistic

**Description**

These functions provide a simple interface to compute (paired) Student t statistics in the analysis of high-dimensional case-control studies.

**Usage**

```
studentt.stat(X, L, var.equal=TRUE, paired=FALSE)
studentt.fun(L, var.equal=TRUE)
```

**Arguments**

- **X** data matrix. Note that the *columns* correspond to variables (“genes”) and the *rows* to samples.
- **L** factor with class labels for the two groups. If only a single label is given then a one-sample t score against 0 is computed.
- **var.equal** assume equal (default) or unequal variances in each group.
- **paired** compute paired t-score (default is to use unpaired t-score).

**Value**

The `studentt.stat` function returns a vector containing the t-statistic for each variable. It can be specified whether the variances in the two groups are equal. A paired t-score can also be computed.

The `studentt.fun` function returns a function that computes the t-score statistics when applied to a data matrix (useful for simulations).

**Author(s)**

See Also

shrinkt.stat, shrinkcat.stat.

Examples

```r
# load st library
library("st")

# load Choe et al. (2005) data
data(choedata)
X <- choe2.mat
dim(X) # 6 11475
L <- choe2.L
L

# L may also contain some real labels
L = c("group 1", "group 1", "group 1", "group 2", "group 2", "group 2")

# student t statistic (equal variances)
score = studentt.stat(X, L)
order(score^2, decreasing=TRUE)[1:10]
# [1] 11068 724 9990 11387 11310 9985 9996 11046 43 50

# the same computed with standard R methods (slower!)
#score2 = apply(X, 2, function(x) t.test(x ~ L, var.equal=TRUE)$statistic)
#sum((score-score2)^2)

# student t statistic (unequal variances)
score = studentt.stat(X, L, var.equal=FALSE)
order(score^2, decreasing=TRUE)[1:10]
# [1] 11068 724 9990 11387 11310 9985 9996 11046 43 50

# the same computed with standard R methods (slower!)
#score2 = apply(X, 2, function(x) t.test(x ~ L, var.equal=FALSE)$statistic)
#sum((score-score2)^2)

# paired student t statistic
score = studentt.stat(X, L, paired=TRUE)
order(score^2, decreasing=TRUE)[1:10]
# [1] 9985 7239 5393 11387 11310 9942 10238 9996 11015 11276

# the same computed with standard R methods (slower!)
#score2 = apply(X, 2, function(x) t.test(x ~ L, paired=TRUE)$statistic)
#sum((score-score2)^2)
```
Index

*Topic **datasets**
chaodata, 2

*Topic **multivar**
  shrinkcat.stat, 11

*Topic **univar**
  cst.stat, 3
diffmean.stat, 5
lait.stat, 6
regularizedt, 8
shrinkt.stat, 14
st-package, 2
studentt.stat, 16

choe2.dogenes (choedata), 2
choe2.mapping (choedata), 2
choe2.probe.name (choedata), 2

catscore, 11, 12
choe2.L (choedata), 2
choe2.mat (choedata), 2
choe2.symbol.name (choedata), 2
choedata, 2
cst.fun (cst.stat), 3
cst.stat, 2, 3, 7, 12
diffmean.fun (diffmean.stat), 5
diffmean.stat, 5, 10, 15
efront.fun (regularizedt), 8
efront.stat (regularizedt), 8
lait.tscore (lait.stat), 6
lait.fun (lait.stat), 6
lait.stat, 2, 4, 6, 12
lowess, 7

modt.fun (regularizedt), 8
modt.stat, 2
modt.stat (regularizedt), 8

rankprod.fun (diffmean.stat), 5
rankprod.stat (diffmean.stat), 5
regularizedt, 8

sam.fun (regularizedt), 8
sam.stat (regularizedt), 8
samL1.fun (regularizedt), 8
samL1.stat (regularizedt), 8
shrinkcat.fun (shrinkcat.stat), 11
shrinkcat.stat, 2, 4, 7, 10, 11, 14, 15, 17
shrinkt.fun (shrinkt.stat), 14
shrinkt.stat, 2, 6, 10–12, 14, 17
st-package, 2
studentt.fun (studentt.stat), 16
studentt.stat, 2, 6, 10, 15, 16