Statistics Toolbox Release Notes

The “Statistics Toolbox 4.0 Release Notes” on page 1-1 describe the changes introduced in the latest version of the Statistics Toolbox.

The following topics are discussed in these Release Notes:

- “New Features” on page 1-2
- “Major Bug Fixes” on page 1-9
- “Upgrading from an Earlier Release” on page 1-10
- “Known Software and Documentation Problems” on page 1-11

If you are upgrading from a release earlier than Release 12, you should also see “Statistics Toolbox 3.0 Release Notes” on page 2-1.

If you are upgrading from a release prior to Release 11.1, see the Release 11 New Features Guide. Note that this is a PDF document.

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

Statistics Toolbox 4.0 Release Notes

1

Statistics Toolbox 3.0 Release Notes

2

1

Statistics Toolbox 4.0 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Statistics Toolbox 4.0.

If you are upgrading from a release earlier than Release 12.0, then you should also see "New Features" on page 2-2 in the Statistics Toolbox 3.0 Release Notes.

Multivariate Analysis

Cluster Analysis

The new `kmeans` function performs K-means clustering and supports five different distance measures. The new function `silhouette` plots silhouettes of clusters created using either K-means or hierarchical clustering methods. The `pdist` function now allows several new distance measures and is more efficient for large datasets.

Factor Analysis

The new `factoran` function fits a Common Factor Analysis model using maximum likelihood, including rotation of the estimated factor loadings and estimation of factor scores.

Multidimensional Scaling and Procrustes Analysis

The new `cmdscale` function performs classical (metric) Multidimensional Scaling, to create a configuration of points in Euclidean space solely from distance data. The new function `procrustes` performs orthogonal Procrustes rotations to match one set of points onto another.

Canonical Correlation Analysis

The new function `canoncorr` performs Canonical Correlation Analysis, to find the subsets of variables in two datasets that best correlate with each other.

Discriminant Analysis

The `classify` function now supports three types of discrimination (linear, quadratic, and Mahalanobis) and allows specification of prior probabilities.

`'linear'` is now the default, and you must specify `'mahalanobis'` to duplicate the behavior of the previous version.

Nonlinear Regression Models

Classification and Regression Trees

A collection of new functions (`treefit`, `treeprune`, `treedisp`, `treetest`, `treeval`) performs classification and regression using decision trees. These functions fit trees to data, display them, prune them, compute error rates for them using test data or cross-validation, and apply them to new data.

Probability Distributions

Several new functions support the generation of random samples from multivariate distributions. There are functions for generating random matrices from the Wishart (`wishrnd`) or inverse Wishart (`iwishrnd`) distributions. Other functions (`lhsdesign`, `lhsnorm`) use latin hypercube sampling methods to generate samples from the multivariate uniform and normal distributions. In addition there have been improvements in other probability functions, particularly those for the negative binomial distribution. Finally, a new function (`mvnpdf`) computes the probability density function for the multivariate Normal distribution.

Descriptive Statistics

Density Estimation

The new `ksdensity` function produces a nonparametric density estimate using a kernel smoothing technique.

Empirical Cumulative Distribution

The new `ecdf` function computes the empirical cumulative distribution function (cdf) and confidence bounds for it. For censored data (common in survival analysis), it computes the Kaplan-Meier estimate of the cdf.

Design of Experiments

Response Surface Designs

New functions support two commonly used designs: central composite designs (`ccdesign`) and Box-Behnken designs (`bbdesign`). Central composite designs fit a full quadratic model and can have three or five levels of each factor. `ccdesign` supports the three types, circumscribed, inscribed and faced.

Box-Behnken designs are rotatable designs that also fit a full quadratic model but use just three levels of each factor.

D-Optimal Designs

The D-optimal design generation functions are faster than in the past. In addition, the two new functions `candgen` and `candexch` provide more control over the row-exchange algorithm for design generation.

Function Summary

Version 4.0 of the Statistics Toolbox provides the following:

- New functions
- Functions with new or changed capabilities

New Functions

Function	Purpose
`bbdesign`	Generate Box-Behnken design
`candexch`	D-optimal design from candidate set using row exchanges
`candgen`	Generate candidate set for D-optimal design
`canoncorr`	Canonical correlation analysis
`ccdesign`	Generate central composite design
`cmdscale`	Classical multidimensional scaling
`ecdf`	Empirical (Kaplan-Meier) cumulative distribution function

Function	Purpose
`factoran`	Perform Factor Analysis by maximum likelihood
`iwishrnd`	Generate inverse Wishart random matrix
`kmeans`	K-means clustering
`ksdensity`	Compute a probability density estimate using a kernel smoothing method
`lhsdesign`	Generate a latin hypercube sample
`lhsnorm`	Generate a multivariate normal random matrix using latin hypercube sampling
`mvnpdf`	Multivariate normal probability density function (pdf)
`nbinfit`	Parameter estimates and confidence intervals for negative binomial data
`procrustes`	Procrustes Analysis
`silhouette`	Silhouette plot for clustered data
`treefit`	Fit a tree-based model for classification or regression.
`treeprune`	Produce a sequence of subtrees by pruning.
`treedisp`	Show classification or regression tree graphically
`treetest`	Compute error rate for tree
`treeval`	Compute fitted value for decision tree applied to data
`wishrnd`	Generate Wishart random matrix

Statistics Functions with New or Changed Capabilities

<table>
<tr><th>Function</th><th colspan="2">Enhancement or Change</th></tr>
<tr><td>classify</td><td colspan="2">A new syntax lets you specify the type of discriminant function as 'linear' (default), 'quadratic', or 'mahalanobis'. Specify 'mahalanobis' to duplicate the behavior of the previous version.
Another new syntax enables you to specify prior probabilities for the groups.
A new output returns an estimate of the misclassification error rate.</td></tr>
<tr><td rowspan="6">cluster</td><td colspan="2">Now also allows clustering based on distance measures. A new syntax also enables you to specify values for these parameters:</td></tr>
<tr><td>'cutoff'</td><td>Cutoff for inconsistent and distance measure</td></tr>
<tr><td>'maxclust'</td><td>Maximum number of clusters to form</td></tr>
<tr><td>'criterion'</td><td>Either 'inconsistent' or 'distance'</td></tr>
<tr><td>'depth'</td><td>Depth for computing inconsistent values</td></tr>
<tr><td colspan="2">The old syntax still works but is undocumented.</td></tr>
</table>

Function	Enhancement or Change	
`clusterdata`	`clusterdata(Z,'param1',val1,'param2',val2,...)` now enables you to specify parameters that `clusterdata` uses in calling `pdist`, `linkage`, and `cluster`:	
	`'distance'`	Any of the distance metric names allowed by `pdist`
	`'linkage'`	Any of the linkage methods allowed by `linkage`
	`'cutoff'`	Cutoff for inconsistent and distance measure
	`'maxclust'`	Maximum number of clusters to form
	`'criterion'`	Either `'inconsistent'` or `'distance'`
	`'depth'`	Depth for computing inconsistent values

`cordexch`
`daugment`
`dcovary`
`rowexch`
	A new syntax provides more control over design generation through a set of parameter-value pairs.
	`function``(...,'param1',value1,'param2',value2,...)`
	Valid parameters are:	

	`'display'`	Controls display of iteration counter.
	`'init'`	Specifies an initial design. The default is a randomly selected set of points.
	`'maxiter'`	Specifies the maximum number of iterations. The default is 10.

Function	Enhancement or Change

`corrcoef`
`(MATLAB)`
	Provides three new syntaxes:
	`[R,P] = corrcoef(...)` returns `P`, a matrix of p-values for testing the hypothesis of no correlation.
	`[R,P,RLO,RUP] = corrcoef(...)` returns matrices `RLO` and `RUP` which contain lower and upper bounds for a 95% confidence interval for each coefficient.
	`[...]=corrcoef(...,'param1',val1,'param2',val2,...)` accepts parameter-value pairs that enable you to override the default confidence interval, and specify how to treat rows of `X` that contain `NaN`s.

`nbincdf`, `nbininv`, `nbinpdf`, `nbinrnd`, `nbinstat`	Consistent with a more general interpretation of the negative binomial, these functions now accept any positive value, including nonintegers, for the size parameter `R`.
`pdist`	Provides four new metrics for calculating the pairwise distance between observations: `'cosine'`, `'correlation'`, `'hamming'`, and `'jaccard'`. It now also accepts a function handle to a user-defined distance function.
`regstats`	A new syntax `stats = regstats(responses,DATA,model,whichstats)` creates an output structure `stats` containing the statistics listed in `whichstats`. `whichstats` can be a single name or a cell array of names. The list of available statistics remains the same.

Major Bug Fixes

The Statistics Toolbox 4.0 includes several bug fixes made since Version 3.0. This section describes the particularly important Version 4.0 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving from the Statistics Toolbox 3.0 to Version 4.0.

Linear and Quadratic Discriminant Analysis Added to classify

The algorithm that was previously implemented in `classify` used the Mahalanobis distance between sample points and training groups, with stratified estimates of covariance. The new implementation adds the standard algorithms for linear (default) and quadratic discriminant analysis. Set `'type'` to `'mahalanobis'` in Version 4.0 (R13) to duplicate the behavior of the previous version.

Use playshow Command to Run glmdemo

Starting in Release 13, to run slideshow style demos such as `glmdemo` from the command line, you must use the `playshow` command. For example,

```
playshow glmdemo
```

You can continue to run other styles of demos from the command line by typing just the demo name. `glmdemo` is the only slideshow style demo in the Statistics Toolbox Version 4.0.

Known Software and Documentation Problems

There is a known software problem in Version 4.0.

Spurious Warning Message from factoran

The first time `factoran` is called (or the first time it is called after a `clear functions` command), you will see the warning message

```
Warning: File: MATLAB\toolbox\stats\factoran.m Line: 333 Column: 1
Variable 'lower' has been previously used as a function name.
```

This warning does not indicate a problem with the results, and can safely be ignored.

2

Statistics Toolbox 3.0 Release Notes

New Features

This section introduces the new features and enhancements added in the Statistics Toolbox 3.0 since the Statistics Toolbox 2.2 (Release 11.0).

Summary of Enhancements

Expanded Support for Linear Models

Version 3.0 expands the Statistics Toolbox support for linear models in general, and analysis of variance in particular. The following are the major changes for the Statistics Toolbox 3.0:

- Improvements in one-way analysis of variance (`anova1`)
- Higher-way analysis of variance (`anovan`)
- Analysis of covariance (`aoctool`)
- Multiple comparisons of means or other estimates (`multcompare`)
- Multivariate analysis of variance (`manova1`, `manovacluster`)
- Graphics functions useful for examining data used for multivariate analysis of variance (`gscatter`, `gplotmatrix`, `gname`)
- Response surface fitting with multiple responses (`rstool`)
- Nonparametric analysis of variance (`friedman`, `kruskalwallis`)
- More flexible calculation of confidence bounds (`polytool`, `nlintool`, `nlpredci`)

Other Enhancements

In addition, the following changes do not involve linear models:

- Generalized linear models (`glmfit`, `glmval`)
- Robust regression (`robustfit`, `polytool`)
- Distribution testing and plotting (`cdfplot`, `lillietest`, `kstest`, `kstest2`)
- Fractional factorial design generation (`fracfact`)
- Importing numeric and text data from tab-delimited files (`tdfread`)
- More flexible handling of grouping variables (`boxplot`, `grpstats`)
- Multivariate t random number generation (`mvtrnd`) and improvements to other t distribution functions

Numerous other functions received enhancements, as described in the following sections:

- "New Functions" on page 2-3
- "New Demos" on page 2-4
- "New Sample Data Files" on page 2-5
- "Updated Functions for ANOVA-Type Tables" on page 2-5
- "Other Updated Functions" on page 2-6

New Functions

The following functions have been added to the Statistics Toolbox 3.0.

Function	Description
`anovan`	N-way Analysis of Variance (ANOVA)
`aoctool`	Interactive plot for fitting and predicting analysis of covariance models
`cdfplot`	Plot of empirical cumulative distribution function
`fracfact`	Generate fractional factorial design from generators
`friedman`	Friedman's nonparametric two-way Analysis of Variance (ANOVA)
`glmfit`	Generalized linear model fitting
`glmval`	Compute predictions for generalized linear model
`gplotmatrix`	Plot matrix of scatter plots by group
`gscatter`	Scatter plot by group
`jbtest`	Jarque-Bera test for goodness-of-fit to a normal distribution
`kruskalwallis`	Kruskal-Wallis nonparametric one-way Analysis of Variance (ANOVA)

Function	Description
`kstest`	Kolmogorov-Smirnov test of the distribution of one sample
`kstest2`	Kolmogorov-Smirnov test to compare the distribution of two samples
`lillietest`	Lilliefors test for goodness-of-fit to a normal distribution
`manova1`	One-way Multivariate Analysis of Variance (MANOVA)
`manovacluster`	Plot dendrogram showing group mean clusters after MANOVA
`multcompare`	Multiple comparison test of means or other estimates
`mvtrnd`	Random matrices from the multivariate t distribution
`robustfit`	Robust regression
`tdfread`	Read file containing tab-delimited numeric and text values

New Demos

glmdemo

The `glmdemo` function is a slideshow-style demo of generalized linear model fitting.

Note To run `glmdemo` from the command line in Version 4.0, Release 13, you must type `playshow glmdemo`. In Version 3.0, Release 12, you need only type `glmdemo`.)

robustdemo

The `robustdemo` function demonstrates robust fitting. The function graphs (*x*,*y*) data with an outlier, and shows how the least squares and robust fits differ. You can move points with the mouse, and see how the two fits change. You can also display the least squares leverage and the robust weight for each point. You can also provide input data instead of using the built-in example.

New Sample Data Files

carbig

The `carbig` data file is a large dataset on cars from the 70s and 80s.

carsmall

The `carsmall` data file is a subset of `carbig`, containing cars from just three model years.

Updated Functions for ANOVA-Type Tables

Several functions display tables, such as ANOVA tables, in a figure window. These figure windows now have a new **Copy Text** option on the **Edit** menu. You can use this option to copy the table as tab-delimited text into Microsoft Excel, Microsoft Word, or other applications. These two functions that produce such tables have been updated:

- `anova1`
- `anova2`

The changes to each of these functions are described below.

Note The `aoctool`, `friedman`, and `kruskalwallis` functions are new functions added in the Statistics Toolbox 3.0; these functions also display ANOVA-type tables.

anova1

```
[p,table,stats] = anova1(x,group,'displayopt')
```

- New output `table` is a cell array of the ANOVA table values, including row and column labels.
- Now returns a `stats` output structure useful for performing multiple comparisons (see `multcompare` for more information).
- New input `'displayopt'` is `'off'` to omit the table and boxplot display, or `'on'` (the default) to display the table and boxplot.
- If `x` is a matrix, `group` can now be a character array or cell array of strings with one row for each column of `x`. The boxes in the boxplot are then labeled using the rows of `group`.
- If `x` is a vector, `group` can be a vector of integers or a character array or cell array of strings with one row for each element of `x`. The boxes in the boxplot are labeled with values from `group`.
- P-value added to both `table` and to the table display.
- Now accepts group numbers that are not of the form 1, ..., *g*.

anova2

```
[p,table,stats] = anova2(x,reps,'displayopt')
```

- Now returns `table` as a second output.
- The additional input `'displayopt'` can be used to suppress the ANOVA table display.
- The additional output `stats` can be used as input to `multcompare` to perform multiple comparisons of row or column means.

Other Updated Functions

Linear Model Functions

Linear model functions (e.g., `anova1`, `polyval`, etc.) ignore observations with `NaN` value in the `X` or `Y` input.

betafit

```
[phat,pci] = betafit(x,alpha)
```

The `betafit` function now:

- Removes `NaN` data before fitting
- Issues an error message if there are any `0` or `1` values
- Issues an error message if `x` is constant

boxplot

```
boxplot(x,notch,sym,vert,whis)
boxplot(x,g,notch,sym,vert,whis)
```

The second syntax for `boxplot` above is new. The first syntax displays a box for each column of the `x` matrix. The second syntax displays a box for each level of the grouping variable `g`. In addition, `g` can be a cell array of grouping variables to produce a separate box for each unique combination of grouping variable levels. See `grpstats`.

cluster

```
T = cluster(Z,cutoff,depth,flag)
```

The `cluster` function adds a flag argument which overrides the default meaning of the `cutoff` argument. If `flag` is `'inconsistent'`, then `cutoff` is interpreted as a threshold for the inconsistency coefficient. If `flag` is `'clusters'`, then `cutoff` is the maximum number of clusters.

crosstab

```
[table,chi2,p,labels] = crosstab(col1,col2,...)
```

The `crosstab` function now accepts any number of inputs, not just two. Each input can be a numeric vector, a string array, or a cell array of strings. (In the previous release each input had to be a vector of positive integers taking values 1, ..., g for some g.) If there are v input variables, the output `table` is a v-dimensional array, with `table(i,j,k,...)` counting the number of times that the first argument takes its ith value, that the second argument takes its jth value, that the third argument takes its kth value, and so on.

For the case of two positive integer input arguments, the function yields the same results as the previous release unless there are missing integers (i.e., not all of 1, ..., *g* appear in the input). In that case, the previous release would have produced a divide-by-zero warning and would have generated a row or column of zeros in table. The new version simply does not consider that category, so it does not reserve zeros for it.

As in the previous release, chi2 is a chi-square statistic for testing independence, and p is its p-value. In this release, table can be other than a two-dimensional table, and the test is that all dimensions are independent.

The labels output is a cell array with one column for each input argument. The column lists the values of that input. Revisiting the example above, table(i,j,k,...) counts the number of times that the first argument takes the value labels{i,1}, that the second argument takes the value labels{j,2}, that the third argument takes the value labels{k,3}, and so on.

ewmaplot

```
h = ewmaplot(data,lambda,alpha,specs)
```

The ewmaplot default for alpha changed to 0.27% to conform to the standard ewma chart definition.

grpstats

```
[means,sem,counts,gname] = grpstats(x,group)
```

The grpstats argument group is no longer restricted to be a vector of integers. It can be a grouping variable that is a numeric vector, a string matrix, or a cell array of strings. In addition it can be a cell array containing multiple group vectors. The function computes statistics on groups defined by unique combinations of levels of the grouping variables. The new output gname is a cell array with one row per group and one column per grouping variable. Elements of means, sem, and counts are statistics calculated for the group defined by values in the corresponding row of gname. Examples include

```
[m,s,c] = grpstats(x,g1);
[m,s,c,gnames] = grpstats(x,{g1 g2});
```

nlinfit

```
[beta,r,J] = nlinfit(X,y,fun,beta0)
```

The nlinfit function now accepts inline functions and function handles (@FF) in addition to the text strings ('FF') accepted in the past for input fun.

nlintool

```
nlintool(x,y,fun,beta0,alpha,'xname','yname')
```

The interface invoked with the `nlintool` function now:

- Adds a new menu option to compute different types of confidence intervals. Intervals can be simultaneous (provide a specified confidence level over all `x` values simultaneously) or nonsimultaneous (provide that level for a single predetermined `x` value). They can apply to the estimated regression function only (not taking account any variability from a new observation) or to a prediction for a new observation (taking its variability into account).
- Accepts inline functions and function handles (`@FF`) in addition to the text strings (`'FF'`) accepted in the past for input `fun`.

nlpredci

```
ypred = nlpredci(fun,inputs,beta,r,J,alpha,'simopt','predopt')
```

The `nlpredci` function has new arguments that allow the same types of confidence intervals produced by `nlintool`.

norminv

```
x = norminv(p,mu,sigma)
```

The `norminv` function now returns `NaN` for each element of `p` that is `NaN`.

normplot

```
h = normplot(x)
```

The `normplot` function now strips `NaN` values individually from each column of `x`.

normrnd

```
r = normrnd(mu,sigma,m,n)
```

The `normrnd` function now returns the mean if `sigma` is `0`.

polytool

```
h = polytool(x,y,n,alpha,xname,yname)
```

The interface invoked by the polytool function has the following enhancements:

- Removes x(j) and y(j) if either is NaN, and display a warning when doing so.
- The **Method** menu provides the option of using robust (bisquare) fitting in place of least squares.
- The new **Bounds** menu option computes different types of confidence intervals. Intervals can be simultaneous (provide a specified confidence level over all x values simultaneously) or nonsimultaneous (provide that level for a single predetermined x value). They can apply to the estimated regression function only (not taking account any variability from a new observation) or to a prediction for a new observation (taking its variability into account).

prctile

```
y = prctile(x,p)
```

The prctile function now strips NaN values individually from each column of x.

qqplot

```
h = qqplot(x,y,pvec)
```

The qqplot function now:

- Strips NaN values individually from each column of x and y.
- If y is omitted, uses standard normal quantiles.

ranksum

```
[p,h,stats] = ranksum(x,y,alpha)
```

New ranksum output stats is a structure that always contains a field named ranksum whose value is the value of the rank sum statistic, and that for large samples contains a field named zval that is the value of the normal (Z) statistic used to compute the p-value p.

schart

```
[outliers,h] = schart(data,conf,specs)
```

The schart default for conf changed to 99.73% to conform to the standard s-chart definition.

signrank

```
[p,h] = signrank(x,y,alpha)
```

The signrank function has the following enhancements:

- If y is a scalar, extend it to the same length as x. This facilitates comparison of the median of one sample to a constant value.
- If x and y are the same, return p=1 and h=0.
- If p=alpha, now h=1 rather than h=0 (rejects hypothesis).
- New return value stats is a structure that always contains a field named signed rank whose value is the value of the signed rank statistic, and that for large samples contains a field named zval that is the value of the normal (Z) statistic used to compute the p-value p.

signtest

```
[p,h,stats] = signtest(x,y,alpha)
```

The signtest function has the following enhancements:

- If p=alpha, now h=1 rather than h=0 (rejects hypothesis)
- New return value stats is a structure that always contains a field named sign whose value is the value of the sign statistic, and that for large samples contains a field named zval that is the value of the normal (Z) statistic used to compute the p-value p. Z is signed, not an absolute value.

tcdf, tinv, tpdf, trnd, tstat

The tcdf, tinv, tpdf, trnd, and tstat functions now accept noninteger degrees of freedom.

ttest

```
[h,sig,ci,stats] = ttest(x,m,alpha,tail)
```

The ttest function has these enhancements:

- Added new output stats, the value of the *t* statistic and its degrees of freedom.
- Removes NaN from x before starting test.

ttest2

```
[h,sig,ci,stats] = ttest2(x,y,alpha,tail)
```

The ttest2 function has these enhancements:

- Adds new output stats, the value of the *t* statistic and its degrees of freedom.
- If tail is 1 or -1, now ci has one endpoint set to Inf or -Inf.
- Removes NaN from x and y before starting test.

weibplot

```
h = weibplot(x)
```

The weibplot function strips NaN values individually from each column of x.

xbarplot

```
[outliers,h] = xbarplot(data,conf,specs,'sigmaest')
```

The xbarplot function has these enhancements:

- Changed default conf to 99.73%, to conform to the standard x-bar chart definition.
- Corrected calculation of control limits. With the new default conf the control limits are three-sigma limits.
- New input sigmaest specifies how to estimate sigma in the control limit calculation. The default is 'std', meaning estimate using the average of the subgroup standard deviations. The value 'variance' uses a pooled variance estimate; this was the value used in previous releases. The value 'range' uses the average of the subgroup ranges, and requires subgroups with no more than 25 observations.

ztest

```
[h,sig,ci,zval] = ztest(x,m,sigma,alpha,tail)
```

The ztest function has these enhancements:

- Adds new output zval, the value of the test statistic.
- If tail is 1 or -1, now ci has one endpoint set to Inf or -Inf.
- Removes NaN from x before starting test.