

Risk Management Toolbox™ Release Notes



MATLAB®



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Risk Management Toolbox™ Release Notes

© COPYRIGHT 2016–2020 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2020a

Consumer Credit Risk: Screen credit scorecard predictors on data that is too big to fit in memory using tall Arrays	1-2
Random number generation for credit copula classes has changed	1-2

R2019b

Market Risk: Backtest expected shortfall (ES) models using Du and Escanciano tests	2-2
Consumer Credit Risk: Validation of compact credit scorecards using validateModel	2-2
Credit Scorecard: Example comparing credit score using logistic regression and decision trees	2-2
Credit Scorecard: Example using reject inference to incorporate credit rejection data into creditscorecard workflow	2-2
Consumer Credit Risk: Example comparing probability of default using through-the-cycle and point-in-time models	2-2
Consumer Credit Risk: Example fitting different types of models to loss given default (LGD) data	2-3
Calculation of p-value for bin has changed	2-3

R2019a

Consumer Credit Risk: Predictor screening for credit scorecards	3-2
Consumer Credit Risk: Support for compact credit scorecards for easier deployment and reduced memory usage	3-2

R2018b

Binning Explorer: Bin data automatically using merge and split algorithms	4-2
Binning Explorer: Bin missing data in a separate bin	4-2
Binning Explorer: Load data from the command line	4-2
Corporate Credit Risk: Perform parallel simulations using creditDefaultCopula and creditMigrationCopula	4-2
creditCopula object removed	4-2

R2018a

Corporate Credit Risk: Calculate standard deviation and value-at-risk contributions for each counterparty in a credit portfolio	5-2
--	-----

R2017b

Corporate Credit Risk: Compute regulatory capital and value-at-risk using an asymptotic single risk factor (ASRF) model	6-2
Corporate Credit Risk: Perform credit portfolio simulation with random loss given default (LGD)	6-2
Market Risk: Backtest expected shortfall models	6-2
Consumer Credit Risk: Specify weights in credit scorecards using Binning Explorer	6-2
creditCopula object renamed	6-3

R2017a

Corporate Credit Risk: Estimate the probability of credit rating migration based on multifactor copula model	7-2
Corporate Credit Risk: Quantify credit concentration risk by Herfindahl index and other concentration measures	7-2

Corporate Credit Risk: Model corporate default risk using Merton model	7-2
creditCopula object renamed	7-2

R2016b

Consumer Credit Risk: Binning Explorer for Credit Scorecards	8-2
Corporate Credit Risk: Copula-based simulation framework	8-2
Market Risk: Value-at-Risk Backtesting Tools	8-2

R2020a

Version: 1.7

New Features

Bug Fixes

Compatibility Considerations

Consumer Credit Risk: Screen credit scorecard predictors on data that is too big to fit in memory using tall Arrays

Tall variable support for screenpredictors.

Random number generation for credit copula classes has changed

In R2019b and previous releases, when using the `creditDefaultCopula` and `creditMigrationCopula` classes to perform nonparallel simulations, the MATLAB® global random number generator was used to generate scenarios.

Compatibility Considerations

In R2020a, the random number generator for the `creditDefaultCopula` and `creditMigrationCopula` classes is set to `Threefry` for both parallel and nonparallel code paths.

R2019b

Version: 1.6

New Features

Bug Fixes

Compatibility Considerations

Market Risk: Backtest expected shortfall (ES) models using Du and Escanciano tests

The following new functions provide support for performing expected shortfall (ES) backtests by Du and Escanciano:

- `esbacktestbyte`
- `summary`
- `runtests`
- `unconditionalDE`
- `conditionalDE`
- `simulate`

New examples demonstrating workflows using expected shortfall (ES) backtests by Du and Escanciano:

- Workflow for Expected Shortfall (ES) Backtesting by Du and Escanciano
- Rolling Windows and Multiple Models for Expected Shortfall (ES) Backtesting by Du and Escanciano

Consumer Credit Risk: Validation of compact credit scorecards using `validateModel`

Work with compact credit scorecards using the `compactCreditScorecard` class and then use `validateModel` to validate a compact credit scorecard.

Credit Scorecard: Example comparing credit score using logistic regression and decision trees

A new example compares credit score values using logistic regression and decision trees (Comparison of Credit Scoring Using Logistic Regression and Decision Trees).

Credit Scorecard: Example using reject inference to incorporate credit rejection data into `creditscorecard` workflow

A new example demonstrates two approaches for using reject inference techniques to incorporate credit rejection data as part of the `creditscorecard` modeling workflow (Use Reject Inference Techniques with Credit Scorecards).

Consumer Credit Risk: Example comparing probability of default using through-the-cycle and point-in-time models

A new example compares the probability of default using through-the-cycle (TTC) and point-in-time (PIT) models (Comparison of Probability of Default Using Through-the-Cycle and Point-in-Time Models).

Consumer Credit Risk: Example fitting different types of models to loss given default (LGD) data

A new example demonstrates fitting different types of models to loss given default (LGD) data (Model Loss Given Default).

Calculation of p-value for bin has changed

In R2019a and previous releases, the binomial VaR backtest reports the tail probability as the p -value and compares the reported p -value to half of the $alpha$ ($1 - \text{TestLevel}$) of the bin test.

In R2019b, the binomial VaR backtest `bin` calculates the p -value contained in the `TestResults` output using the 2*tail probability convention and the p -value can be compared to $alpha$. For more information, see Algorithms.

R2019a

Version: 1.5

New Features

Bug Fixes

Consumer Credit Risk: Predictor screening for credit scorecards

Perform predictor screening for credit scorecards using the `screenpredictors` function. For more information, see [Feature Screening with `screenpredictors`](#).

Consumer Credit Risk: Support for compact credit scorecards for easier deployment and reduced memory usage

Work with compact credit scorecards using the `compactCreditScorecard` class along with associated functions for `displaypoints`, `score`, and `probdefault`. In addition, you can create a compact credit scorecard by using the `compact` function from [Financial Toolbox™](#) with a `creditscorecard` object. For more information, see [compactCreditScorecard Object Workflow and Validate the Quality of a Compact Credit Scorecard Model](#).

R2018b

Version: 1.4

New Features

Bug Fixes

Compatibility Considerations

Binning Explorer: Bin data automatically using merge and split algorithms

The Binning Explorer app supports the merge and split algorithms. For more information, see **Binning Explorer**.

Binning Explorer: Bin missing data in a separate bin

The **Binning Explorer** app supports binning missing data for a predictor in a separate <missing> bin. For more information, see **Binning Explorer**.

Binning Explorer: Load data from the command line

The Binning Explorer app supports loading data or a `creditscorecard` object from the command line. For more information, see **Binning Explorer**.

Corporate Credit Risk: Perform parallel simulations using `creditDefaultCopula` and `creditMigrationCopula`

The `creditDefaultCopula` and `creditMigrationCopula` classes support the 'UseParallel' property for parallel simulations when using the `simulate` and `riskContributions` functions. You can set the 'UseParallel' property when creating `creditDefaultCopula` or `creditMigrationCopula` objects only if you have Parallel Computing Toolbox™.

`creditCopula` object removed

The `creditCopula` object is removed.

Compatibility Considerations

Object Name	What Happens When You Use It	Use This Instead	Compatibility Considerations
<code>creditCopula</code>	Removed	<code>creditDefaultCopula</code>	Replace all instances of <code>creditCopula</code> object with a <code>creditDefaultCopula</code> object using <code>creditDefaultCopula</code> .

R2018a

Version: 1.3

New Features

Bug Fixes

Corporate Credit Risk: Calculate standard deviation and value-at-risk contributions for each counterparty in a credit portfolio

The `riskContribution` function for `creditDefaultCopula` and the `riskContribution` function for `creditMigrationCopula` support returned information for counterparty contributions for standard deviation of the losses (Std) and value at risk (VaR) at the threshold `VaRLevel`.

R2017b

Version: 1.2

New Features

Bug Fixes

Compatibility Considerations

Corporate Credit Risk: Compute regulatory capital and value-at-risk using an asymptotic single risk factor (ASRF) model

The `asrf` function provides an ASRF model for credit risk analysis. `asrf` accepts the risk characteristics of a portfolio of credit-sensitive instruments as input and computes the necessary capital using an ASRF model.

Corporate Credit Risk: Perform credit portfolio simulation with random loss given default (LGD)

Support is provided for specifying random LGD (loss given default) for `creditDefaultCopula` and `creditMigrationCopula` objects. You can now specify the LGD input argument as a `NumCounterparties-by-2` matrix, where the first column contains the LGD mean values and the second column contains the LGD standard deviations. In this case, LGD values are drawn randomly from a beta distribution with the parameters provided for the defaulting counterparty.

Market Risk: Backtest expected shortfall models

The following tools support expected shortfall (ES) backtesting for table-based tests for the unconditional Acerbi-Szekely test.

- `esbacktest`
- `summary`
- `runtests`
- `unconditionalNormal`
- `unconditionalT`

The following tools support expected shortfall (ES) backtesting for distribution tests for normal and `t` distributions.

- `esbacktestbysim`
- `summary`
- `runtests`
- `conditional`
- `unconditional`
- `quantile`
- `simulate`

Consumer Credit Risk: Specify weights in credit scorecards using Binning Explorer

Specify weights in a credit scorecard when using the Binning Explorer app. For more information on defining weights for a `creditscorecard` object, see the optional name-value pair argument `WeightsVar` for `creditscorecard`.

creditCopula object renamed

The `creditCopula` object is renamed to the `creditDefaultCopula` object.

Compatibility Considerations

Object Name	What Happens When You Use This Object	Use This Object Instead	Compatibility Considerations
<code>creditCopula</code>	Errors	<code>creditDefaultCopula</code>	<p>Replace all instances of <code>creditCopula</code> object with <code>creditDefaultCopula</code> object using <code>creditDefaultCopula</code>.</p> <hr/> <p>Note The <code>CounterpartyLosses</code> property of the <code>creditCopula</code> object is removed in the <code>creditDefaultCopula</code> object. To obtain counterparty losses, use the <code>getScenarios</code> function.</p>

R2017a

Version: 2.5

New Features

Bug Fixes

Compatibility Considerations

Corporate Credit Risk: Estimate the probability of credit rating migration based on multifactor copula model

The following tools support corporate credit portfolio analysis for credit migration simulation using a `creditMigrationCopula` object for copula-based simulations:

- `creditMigrationCopula`
- `simulate`
- `portfolioRisk`
- `riskContribution`
- `confidenceBands`
- `getScenarios`

Corporate Credit Risk: Quantify credit concentration risk by Herfindahl index and other concentration measures

The `concentrationIndices` function supports the following concentration indices:

- CR — Concentration ratio
- Deciles — Deciles of the portfolio weights distribution
- Gini — Gini coefficient
- HH — Herfindahl-Hirschman index
- HK — Hannah-Kay index
- HT — Hall-Tideman index
- TE — Theil entropy index

Corporate Credit Risk: Model corporate default risk using Merton model

The `mertonmodel` and `mertonByTimeSeries` functions estimate the default probability using Merton's model.

creditCopula object renamed

The `creditCopula` object is renamed to the `creditDefaultCopula` object.

Compatibility Considerations

Object Name	What Happens When You Use This Object	Use This Object Instead	Compatibility Considerations
creditCopula	Warns	creditDefaultCopula	<p>Replace all instances of <code>creditCopula</code> object with <code>creditDefaultCopula</code> object using the <code>creditDefaultCopula</code> constructor.</p> <hr/> <p>Note The <code>CounterpartyLosses</code> property of <code>creditCopula</code> object is removed in the <code>creditDefaultCopula</code> object. To obtain counterparty losses, use the <code>getScenarios</code> function.</p>

R2016b

Version: 2.4

New Features

Consumer Credit Risk: Binning Explorer for Credit Scorecards

Binning Explorer is an app for developing and modifying binning assignments for a `creditscorecard` object. For more information, see [Binning Explorer](#).

Corporate Credit Risk: Copula-based simulation framework

The following tools support corporate credit portfolio analysis using a `creditCopula` object for copula-based simulations:

- `creditCopula` — Creates a `creditCopula` object.
- `simulate` — Simulates credit defaults using a `creditCopula` object.
- `portfolioRisk` — Generates portfolio-level risk measurements for a `creditCopula` object.
- `confidenceBands` — Generates confidence interval bands for a `creditCopula` object.
- `riskContribution` — Generates risk contributions for each counterparty in the `creditCopula` object.

Market Risk: Value-at-Risk Backtesting Tools

Value-at-risk (VaR) is an important measure of financial risk. VaR is an estimate of how much value a portfolio can lose in a given time period with a given confidence level. VaR backtesting tools assess the accuracy of VaR models. The following VaR backtesting tools are supported:

- `varbacktest` — Creates a `varbacktest` object using portfolio outcomes data and corresponding value-at-risk (VaR) data.
- `bin` — Binomial test.
- `cc` — Christoffersen's conditional coverage mixed test.
- `cci` — Christoffersen's conditional coverage independence test.
- `pof` — Kupiec's proportion of failures test.
- `tbf` — Haas's time between exceptions independence test.
- `tbf_i` — Haas's mixed time between exceptions (independence and frequency) test.
- `tl` — Traffic light test.
- `tuff` — Kupiec's time until the first failure test.
- `summary` — Summary report on the given `varbacktest` data.
- `runtests` — Runs all tests and reports the final test results.