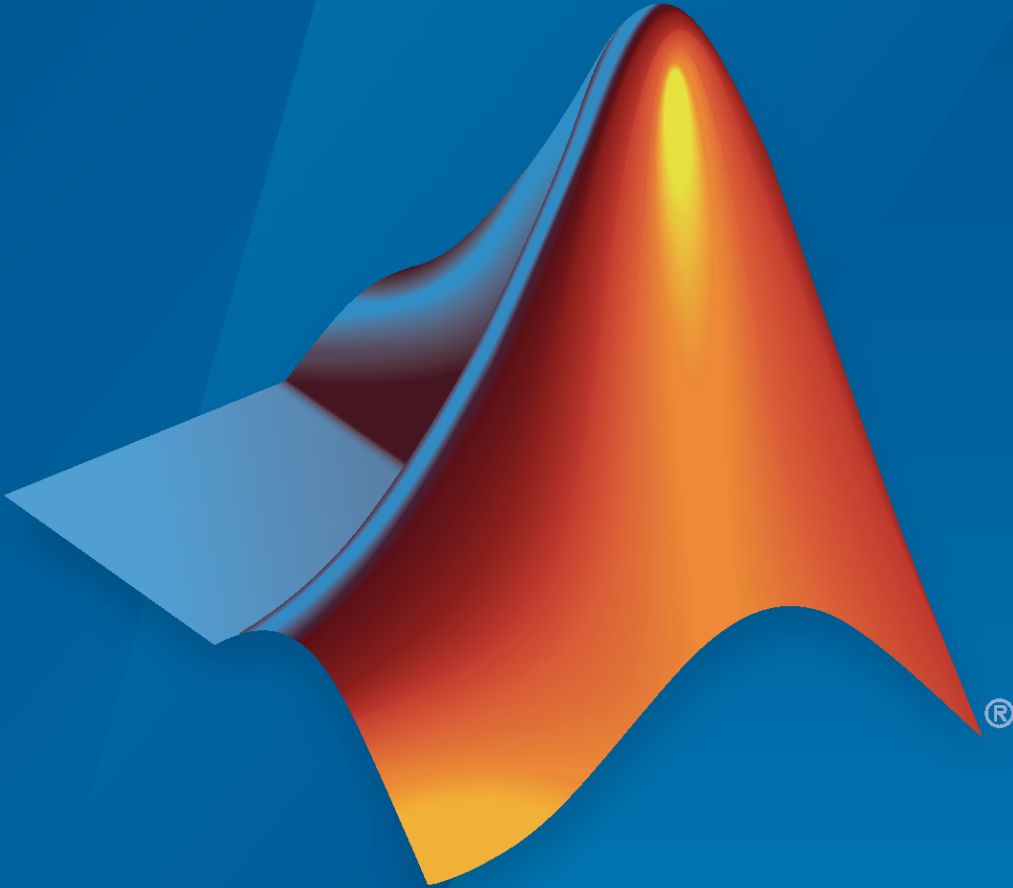


Financial Toolbox™ Release Notes



MATLAB®



## How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

### *Financial Toolbox™ Release Notes*

© COPYRIGHT 2005–2023 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](http://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](http://www.mathworks.com/patents) for more information.

## R2023a

<b>SDE Framework: Milstein scheme</b> .....	1-2
<b>Backtesting Framework: Data sharing in user-defined function handles</b> .....	1-2
<b>Portfolio Management: Covariance shrinkage and denoising</b> .....	1-2
<b>Portfolio Management: Set lower bound for trust-region radius</b> .....	1-2
<b>Portfolio Management: Plots for Brinson model performance attribution</b> .....	1-3
<b>Portfolio Management Example: Dynamic programming for wealth management</b> .....	1-3
<b>Example: Reinforcement learning for optimal trade execution</b> .....	1-3
<b>Portfolio Management: Name change for MasterSolverOptions and IntMasterSolverOptions</b> .....	1-3
<b>fints and related methods removal</b> .....	1-4
<b>fints support removed for input data for technical indicators</b> .....	1-9

## R2022b

<b>SDE Framework: Perform Brownian Bridge and Principal Component constructions</b> .....	2-2
<b>Backtesting Framework: Management and performance fees</b> .....	2-2
<b>Portfolio Management: Custom objective functions</b> .....	2-2
<b>Portfolio Management: Brinson model for performance attribution</b> ....	2-3
<b>Portfolio Management Example: Diversification of ESG portfolios</b> .....	2-3
<b>Technical Indicators and Charting: Support for negative prices</b> .....	2-3

<b>ftstool and ftsgui removal</b> .....	<b>2-4</b>
---	------------

## **R2022a**

<b>SDE Framework: Perform Quasi-Monte Carlo simulation</b> .....	<b>3-2</b>
<b>Backtesting Framework: Backtesting enhancements</b> .....	<b>3-2</b>
<b>Backtesting Framework Example: Backtest with Deep Learning strategies</b> .....	<b>3-3</b>
<b>Portfolio Management: Manage a risk parity portfolio</b> .....	<b>3-3</b>
<b>Portfolio Management: Portfolio enhancements</b> .....	<b>3-3</b>
<b>Data Transformation: Negative prices example</b> .....	<b>3-3</b>
<b>Date and Time: Move of date and time functions to MATLAB</b> .....	<b>3-3</b>

## **R2021b**

<b>Portfolio Management Example: Portfolio optimization using social criteria constraints</b> .....	<b>4-2</b>
<b>Portfolio Management Example: Portfolio diversification</b> .....	<b>4-2</b>
<b>Portfolio Management: Robustness improvements for MINLP solvers</b> ...	<b>4-2</b>
<b>Reinforcement Learning Example: Price and hedge instruments using Reinforcement Learning Toolbox</b> .....	<b>4-2</b>

## **R2021a**

<b>Backtesting Framework: Plot equity curves for each investment strategy</b> .....	<b>5-2</b>
<b>Backtesting Framework: Specify different frequencies for backtesting from datetime, duration, or calendarDuration arrays</b> .....	<b>5-2</b>
<b>Backtesting Framework: Define cash borrowing rate and risk free rate as an annualized rate</b> .....	<b>5-2</b>

<b>Portfolio Management Example: Hedging strategies using CVaR portfolio optimization</b> .....	5-2
<b>Financial Time Series: Aggregate timetable data for different periodicities ranging from daily to annually</b> .....	5-3

## R2020b

<b>Backtesting Workflow: Define investment strategies, run backtests, and summarize results</b> .....	6-2
<b>Stochastic Differential Equation Models: Transition density simulation for Heston and Bates models</b> .....	6-2
<b>Financial Time Series: Calculate period-over-period rolling returns with business calendars</b> .....	6-2

## R2020a

<b>Stochastic Differential Equation Models: Perform Monte Carlo simulation for the Bates and Merton jump diffusion models</b> .....	7-2
<b>Stochastic Differential Equation Models: Simulate Bates, Heston, CIR sample paths by Quadratic-Exponential discretization scheme</b> .....	7-2
<b>Credit Scorecards: Replace a missing value in credit scorecard predictors with mean, median, mode or a custom value</b> .....	7-2
<b>Machine Learning Examples: Series of examples on machine learning for statistical arbitrage</b> .....	7-2

## R2019b

<b>Portfolio Management: Perform CVaR and MAD portfolio optimization using nonlinear solvers (TrustRegionCP and ExtendedCP)</b> .....	8-2
<b>Portfolio Management Example: Optimize the portfolio using a factor model</b> .....	8-2
<b>Transition Probability Estimates: Exclude 'NR' rating from transition matrix</b> .....	8-2

<b>Functionality being removed or changed</b> .....	<b>8-2</b>
---	------------

**R2019a**

<b>Credit Scorecards: Specify equality, inequality, or bound constraints using fitConstrainedModel</b> .....	<b>9-2</b>
<b>Portfolio Management: Perform CVaR and MAD portfolio optimization with integrality constraints such as minimum and maximum number of assets</b> .....	<b>9-2</b>
<b>Portfolio Management: Configure solver options for the solver linprog for portfolio optimization</b> .....	<b>9-2</b>
<b>Portfolio Management Example: Optimize the portfolio using the Black-Litterman model</b> .....	<b>9-3</b>
<b>Functionality being removed or changed</b> .....	<b>9-3</b>

**R2018b**

<b>Credit Scorecards: Bin missing data in a separate bin</b> .....	<b>10-2</b>
<b>Portfolio Management: Perform mean-variance portfolio optimization with integrality constraints such as minimum and maximum number of assets</b> .....	<b>10-2</b>
<b>Implied Volatility: Improve performance of the blsimpv and blkimpv functions when using the Jäckel 2016 method</b> .....	<b>10-3</b>
<b>Stochastic Differential Equation Models: Perform Monte Carlo simulation for the Cox-Ingersoll-Ross model</b> .....	<b>10-3</b>
<b>Functionality being removed or changed</b> .....	<b>10-3</b>

**R2018a**

<b>Credit Scorecards: Bin creditscorecard data using supervised binning algorithms, including merge and split</b> .....	<b>11-2</b>
<b>Portfolio Management: Improve performance for estimations of efficient frontiers and portfolios</b> .....	<b>11-2</b>

<b>Sharpe Ratio of Portfolio: Estimate the Sharpe ratio of portfolio weights given custom portfolio weights</b> .....	<b>11-2</b>
<b>Portfolio Management: Input table and timetable objects for estimateAssetMoments, setScenarios, and simulateNormalScenariosByData</b> .....	<b>11-2</b>
<b>Financial Time Series: Input table and timetable objects for technical indicators and financial charts</b> .....	<b>11-2</b>
<b>Financial Time Series: fints object removed</b> .....	<b>11-3</b>
<b>Financial Time Series: fints-related functions removed</b> .....	<b>11-4</b>
fints object support removed from Portfolio, PortfolioCVaR, and PortfolioMAD objects .....	<b>11-15</b>
Technical indicators support for fints removed .....	<b>11-16</b>
User interface tools for data extraction supporting fints removed .....	<b>11-19</b>
Charting functions support removed for optional inputs for 'Dates' and 'Dateform' .....	<b>11-20</b>

## **R2017b**

<b>Credit Scorecards: Support weights in credit scorecards</b> .....	<b>12-2</b>
--	-------------

## **R2017a**

<b>Default Probability Modeling: Bootstrap default probabilities from bonds using Jarrow-Turnbull model</b> .....	<b>13-2</b>
<b>Credit Scorecards: Support strings in credit scorecard</b> .....	<b>13-2</b>
<b>Financial Time Series app removal for Database Toolbox support</b> .....	<b>13-2</b>

## **R2016b**

<b>Credit Scorecards: Last binning operation in creditscorecard</b> .....	<b>14-2</b>
<b>Functions moved to Financial Toolbox from Financial Instruments Toolbox</b> .....	<b>14-2</b>
<b>help findemos removal</b> .....	<b>14-2</b>

<b>Plots: Fan chart enhancements</b> .....	15-2
<b>Date and Time: datetime support for calendar functions</b> .....	15-2
<b>Date and Time: function to return the quarter of a given date</b> .....	15-5
<b>Functionality Removed</b> .....	15-5
<b>ugarch removal</b> .....	15-5
<b>ugarchllf removal</b> .....	15-6
<b>ugarchpred removal</b> .....	15-6
<b>ugarchsim removal</b> .....	15-7
<b>frontcon removal</b> .....	15-7
<b>portopt partial removal</b> .....	15-8

<b>Portfolio Optimization: Calculate mean-variance portfolios with tracking error constraint</b> .....	16-2
<b>Credit Scorecards: Set predictor types to numeric or categorical and view summary information</b> .....	16-2
<b>Transition Probability Estimates: Enter data using table input</b> .....	16-2
<b>Simple Interest Convention: Calculate zero, forward, and discount curves using simple interest</b> .....	16-2
<b>Functionality Being Changed for fwd2zero, zero2fwd, pyld2zero, and zero2pyld</b> .....	16-3
<b>ugarch removal</b> .....	16-3
<b>ugarchllf removal</b> .....	16-3
<b>ugarchpred removal</b> .....	16-4
<b>ugarchsim removal</b> .....	16-4
<b>frontcon removal</b> .....	16-5



<b>portopt partial removal</b> .....	<b>16-5</b>
--------------------------------------	-------------

## **R2015a**

<b>Credit scorecard enhancements for model validation, a binning algorithm, and probability of default computation</b> .....	<b>17-2</b>
<b>autobinning support for 'Monotone' has compatibility impact</b> .....	<b>17-2</b>
<b>Life table calibration and simulation for insurance</b> .....	<b>17-2</b>
<b>SDE suite parallel computing example</b> .....	<b>17-3</b>
<b>frontcon removal</b> .....	<b>17-3</b>
<b>portopt partial removal</b> .....	<b>17-3</b>

## **R2014b**

<b>Credit scorecard functionality</b> .....	<b>18-2</b>
<b>Performance improvements to CVaR portfolio optimization when using the fmincon function</b> .....	<b>18-2</b>
<b>Performance improvements to SDE Monte Carlo simulation for models with constant parameter or deterministic function of time</b> .....	<b>18-2</b>
<b>Fan chart visualization function</b> .....	<b>18-2</b>
<b>SDE functions accept parameters that can be specified as a single input argument</b> .....	<b>18-2</b>
<b>Default option for the cuttingplane solver for PortfolioCVaR optimization changed</b> .....	<b>18-3</b>

## **R2014a**

<b>SDE functions moved to Financial Toolbox from Econometrics Toolbox</b> .....	<b>19-2</b>
<b>Performance enhancements to SDE Monte Carlo simulation functions</b> .....	<b>19-2</b>

## R2013b

<b>Mean-absolute deviation (MAD) portfolio optimization</b> .....	20-2
<b>optimoptions support</b> .....	20-2
<b>Functions moved from Financial Instruments Toolbox to Financial Toolbox</b> .....	20-2

## R2013a

<b>Cash flow plot function</b> .....	21-2
<b>Financial Time Series Tool (ftstool) import of Excel XLSX files on Linux and Mac OS X</b> .....	21-2
<b>Cutting-plane solver added to PortfolioCVaR object</b> .....	21-2
<b>transprobbytots errors when using the algorithm input argument</b> ...	21-2
<b>Using datenum, datestr, datevec with dates in Financial products might produce inconsistent results</b> .....	21-2

## R2012b

<b>Conditional value at risk (CVaR) portfolio optimization</b> .....	22-2
<b>Margin and spread calculations for floating-rate bonds</b> .....	22-2
<b>Total (horizon) return calculation for fixed-coupon bonds</b> .....	22-2
<b>Performance improvements for cfamounts</b> .....	22-2

## R2012a

<b>xirr Update</b> .....	23-2
<b>Additional Support for Cash Flow Functions</b> .....	23-2
<b>New Demo for Portfolio Optimization Tools</b> .....	23-2

## R2011b

<b>One-Way Turnover Constraints Added to the Portfolio Object</b> .....	24-2
<b>Portfolio Optimization with Sharpe Ratio Maximization Using a Portfolio Object</b> .....	24-2
<b>Cash Flow and Time Mapping for Bond Portfolios with Variable Coupon Rates and Variable Face Values</b> .....	24-2
<b>Transition Probability Functions for Credit Quality Thresholds, Nonsquare Matrices, and User-Defined Ratings</b> .....	24-2
<b>New Demo for Forecasting Corporate Default Rates</b> .....	24-2
<b>Functionality Being Removed</b> .....	24-2
<b>Warning and Error ID Changes</b> .....	24-3
<b>transprobbytots Warns When Using the algorithm Input Argument</b> .....	24-3

## R2011a

<b>Portfolio Turnover and Transaction Costs</b> .....	25-2
<b>Updated showdemo Command for Credit Rating Demo</b> .....	25-2

## R2010b

<b>Estimation of Transition Probabilities for Credit Risk</b> .....	26-2
<b>Improved Performance in Portfolio Optimization Functions</b> .....	26-2
<b>New Demo for Credit Rating</b> .....	26-2
<b>New Input and Output Options for Swap Functionality</b> .....	26-2

---

**R2010a**

**No New Features or Changes**

---

**R2009b**

<b>Support for the BUS/252 Day-Count Convention</b> .....	<b>28-2</b>
<b>Extended Support for New York Stock Exchange Closures</b> .....	<b>28-2</b>
<b>Enhancements for Bond Pricing</b> .....	<b>28-2</b>

---

**R2009a**

<b>Support for Key Rate Duration</b> .....	<b>29-2</b>
--	-------------

---

**R2008b**

**No New Features or Changes**

---

**R2008a**

<b>Enhanced Mean-Variance Portfolio Optimization Based on Linear Complementarity Programming for Portfolio Optimization</b> .....	<b>31-2</b>
<b>Support for Actual/365 (ISDA)</b> .....	<b>31-2</b>
<b>Support for ret2tick and tick2ret Functions for Time Series Objects</b> ...	<b>31-3</b>
<b>Support for Additional Descriptive Statistics Functions Financial Time Series Objects</b> .....	<b>31-3</b>
<b>Added New Chart Types</b> .....	<b>31-3</b>

## R2007b

<b>ISMA Support for 30/360 Basis as a Variant of 30/360E with Annual Compounding</b> .....	32-2
<b>createholidays Function Added for Different Trading Calendars</b> .....	32-3
<b>Diagonal Covariance Matrix Support Added for Multivariate Normal Regression</b> .....	32-3
<b>arith2geom and geom2arith Functions Added for Portfolio Analysis</b> ...	32-3

## R2007a

<b>ISMA Support Added</b> .....	33-2
---------------------------------	------

## R2006b

<b>Investment Performance Metrics</b> .....	34-2
<b>Financial Time Series Tool</b> .....	34-2

## R2006a

<b>Financial Time Series Toolbox Incorporated</b> .....	35-2
<b>Financial Time Series Frequency Conversion Functions Modified</b> .....	35-2
<b>Continuous Compounding Option Removed from plyd2zero</b> .....	35-2
<b>New Statistical Functions</b> .....	35-2
Multivariate Normal Regression Without Missing Data .....	35-2
Multivariate Normal Regression With Missing Data (Expectation Conditional Maximization) .....	35-2
Least Squares Regression With Missing Data (Expectation Conditional Maximization) .....	35-3
Financial Model Transformation Function .....	35-3

---

<b>New Statistical Functions</b> .....	<b>36-2</b>
Expectation Conditional Maximization .....	<b>36-2</b>

# R2023a

---

**Version: 6.5**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## SDE Framework: Milstein scheme

You can now specify the Milstein scheme by using the `simByMilstein` method or `simulate` function with the following stochastic differential equation objects:

Class Names	Method Supporting Milstein Scheme
bates	<code>simByMilstein</code> , <code>simulate</code>
cir	<code>simByMilstein</code> , <code>simulate</code>
heston	<code>simByMilstein</code> , <code>simulate</code>
merton	<code>simByMilstein</code> , <code>simulate</code>
bm, gbm, cev, hmv, sdeddo, sdeId, and sdemrd	<code>simByMilstein</code> , <code>simulate</code>

## Backtesting Framework: Data sharing in user-defined function handles

Use the `backtestStrategy` object to implement a trading strategy that requires some persistent strategy data, or else a strategy that depends on some backtest engine state data. The `backtestStrategy` object supports these properties:

- `UserData` property — A struct to hold strategy-specific data used by the user-defined function handles (`rebalanceFcn` and `computeTransactionCostsFcn`). When you set the optional name-value argument `UserData`, you can use the `UserData` struct to read and write to a persistent data structure. When you run a backtest, the `UserData` struct is passed to your `rebalanceFcn` and `computeTransactionCostsFcn` function handles as an additional input argument. For an example of `backtestStrategy` using `UserData`, see “Backtest with Brinson Attribution to Evaluate Portfolio Performance”.
- `EngineDataList` property — A string or string array to specify a set of optional backtest state data that your strategy needs for the `rebalanceFcn` and `computeTransactionCostsFcn` function handles. When you set the optional name-value argument `EngineDataList`, you can access intermediate backtest values during the execution of the `rebalanceFcn` or `computeTransactionCostsFcn` function handles. For an example of `backtestStrategy` using `EngineDataList`, see “Use `EngineDataList` Property to Enforce Trading Strategy of Whole Shares”.

## Portfolio Management: Covariance shrinkage and denoising

In some situations, the traditional sample approximation of a covariance matrix performs poorly. You can use the `covarianceShrinkage` function to apply a shrinkage estimation and the `covarianceDenoising` function to reduce the noise when using a covariance matrix as an input to a `Portfolio` object. For more information on using covariance shrinkage and denoising, see “Comparison of Methods for Covariance Estimation”. For an example of `covarianceDenoising`, see “Compare Performance of Covariance Denoising with Factor Modeling Using Backtesting”.

## Portfolio Management: Set lower bound for trust-region radius

Use the `DeltaLowerBound` name-value argument to set the lower bound for the trust-region radius.

When using `setSolver` with a `PortfolioCVaR` or `PortfolioMAD` object, you can use `DeltaLowerBound` name-value argument with `solverType` set to 'TrustRegionCP'.



---

When using `setSolverMINLP` with a `Portfolio`, `PortfolioCVaR`, or `PortfolioMAD` object, you can use `DeltaLowerBound` with `solverTypeMINLP` set to `'TrustRegionCP'`.

## Portfolio Management: Plots for Brinson model performance attribution

The `brinsonAttribution` object implements the Brinson model for portfolio performance attribution. After creating a `brinsonAttribution` object, you can use the following functions to create plots:

- `categoryReturnsChart`
- `categoryWeightsChart`
- `attributionsChart`

The “Backtest with Brinson Attribution to Evaluate Portfolio Performance” example shows how to use the `brinsonAttribution` object and the backtesting framework to evaluate a portfolio's performance versus a benchmark.

## Portfolio Management Example: Dynamic programming for wealth management

The “Single Period Goal-Based Wealth Management” example shows a single-period approach to goal-based wealth management (GBWM). The “Dynamic Portfolio Allocation in Goal-Based Wealth Management for Multiple Time Periods” example shows a multi-period approach to dynamic programming for GBWM.

## Example: Reinforcement learning for optimal trade execution

The “Deep Reinforcement Learning for Optimal Trade Execution” example shows how to use Reinforcement Learning Toolbox™ and Deep Learning Toolbox™ to design agents for optimal trade execution.

## Portfolio Management: Name change for `MasterSolverOptions` and `IntMasterSolverOptions`

The `MasterSolverOptions` name-value argument is renamed to `MainSolverOptions`. The `IntMasterSolverOptions` name-value argument is renamed to `IntMainSolverOptions`. The use of `MasterSolverOptions` and `IntMasterSolverOptions` name-value arguments is discouraged.

## Compatibility Considerations

Function	Compatibility Considerations
<code>setSolver</code> using the name-value argument for <code>MasterSolverOptions</code>	<code>MasterSolverOptions</code> is renamed to <code>MainSolverOptions</code> . <code>MasterSolverOptions</code> is not recommended. Use <code>MainSolverOptions</code> instead.

Function	Compatibility Considerations
setSolverMINLP using the name-value argument for IntMasterSolverOptions	IntMasterSolverOptions is renamed to IntMainSolverOptions. IntMasterSolverOptions is not recommended. Use IntMainSolverOptions instead.

## fints and related methods removal

The `fints` function and the following related methods have been removed or changed to no longer support a `fints` object as an input.

### Compatibility Considerations

Function	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
<code>fints</code>	Errors	There is no replacement function. Use <code>timetable</code> instead. For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of <code>fints</code> .
<code>fts2timetable</code>	Removed	There is no replacement function. Use <code>timetable</code> instead. For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of <code>fts2timetable</code>
<code>ascii2fts</code>	Removed	There is no replacement function. Use <code>timetable</code> instead. For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of <code>ascii2fts</code>
<code>candle (fts)</code>	Removed	Use <code>candle</code> .	Remove all instances of <code>candle (fts)</code>
<code>chartfts</code>	Removed	There is no replacement function.	Remove all instances of <code>chartfts</code>
<code>chfield</code>	Removed	There is no replacement function.	Remove all instances of <code>chfield</code>
<code>convertto</code>	Removed	Use <code>timetable</code> and <code>retime</code> . For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of <code>convertto</code>
<code>corrcoef</code>	Removed	There is no replacement function.	Remove all instances of <code>corrcoef</code>
<code>cov</code>	Removed	Use <code>timetable</code> and MATLAB <sup>®</sup> <code>cov</code> . For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of Financial Toolbox <code>cov</code>

<b>Function</b>	<b>What Happens When You Use This Function?</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
cumsum	Removed	Use <code>timetable</code> and MATLAB <code>cumsum</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>cumsum</code>
diff	Removed	Use <code>timetable</code> and MATLAB <code>diff</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>diff</code>
display	Removed	There is no replacement function.	Remove all instances of <code>display</code>
end	Removed	There is no replacement function.	Remove all instances of <code>end</code>
eq (fts)	Removed	There is no replacement function.	Remove all instances of <code>eq (fts)</code>
exp	Removed	There is no replacement function.	Remove all instances of <code>exp</code>
extfield	Errors	There is no replacement function.	Remove all instances of <code>extfield</code>
fetch	Removed	There is no replacement function.	Remove all instances of <code>fetch</code>
fieldnames	Removed	There is no replacement function.	Remove all instances of <code>fieldnames</code>
fillts	Removed	Use <code>timetable</code> and <code>fillmissing</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>fillts</code>
filter	Removed	Use <code>timetable</code> and MATLAB <code>filter</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>filter</code>
freqnum	Removed	There is no replacement function.	Remove all instances of <code>freqnum</code>
freqstr	Removed	There is no replacement function.	Remove all instances of <code>freqstr</code>
fts2ascii	Removed	Use <code>timetable</code> and <code>writetable</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>fts2ascii</code>
fts2mat	Removed	There is no replacement function. Use <code>timetable</code> instead. For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>fts2mat</code>
ftsbound	Removed	There is no replacement function.	Remove all instances of <code>ftsbound</code>
ftsinfo	Removed	There is no replacement function.	Remove all instances of <code>ftsinfo</code>

Function	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
ftsuniq	Removed	There is no replacement function.	Remove all instances of <code>ftsuniq</code>
getfield	Removed	There is no replacement function.	Remove all instances of <code>getfield</code>
getnameidx	Removed	Use <code>contains</code> .	Remove all instances of <code>getnameidx</code>
highlow (fts)	Removed	Use <code>highlow</code> .	Remove all instances of <code>highlow fts</code>
hist	Removed	There is no replacement function.	Remove all instances of <code>hist</code>
horzcat	Removed	Use <code>timetable</code> and MATLAB <code>horzcat</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>horzcat</code>
iscompatible	Removed	There is no replacement function.	Remove all instances of <code>iscompatible</code>
isempty	Removed	There is no replacement function.	Remove all instances of <code>isempty</code>
isequal	Removed	There is no replacement function.	Remove all instances of <code>isequal</code>
isfield	Removed	There is no replacement function.	Remove all instances of <code>isfield</code>
issorted	Removed	Use <code>timetable</code> and MATLAB <code>issorted</code> .	Remove all instances of Financial Toolbox <code>issorted</code>
lagts	Removed	Use <code>timetable</code> and <code>lag</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>lagts</code>
leadts	Removed	Use <code>timetable</code> and <code>lag</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>leadts</code>
length	Removed	There is no replacement function.	Remove all instances of <code>length</code>
log	Removed	There is no replacement function.	Remove all instances of <code>log</code>
log10	Removed	There is no replacement function.	Remove all instances of <code>log10</code>
log2	Removed	There is no replacement function.	Remove all instances of <code>log2</code>
max	Removed	Use <code>timetable</code> and MATLAB <code>max</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>max</code>
mean	Removed	Use <code>timetable</code> and MATLAB <code>mean</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>mean</code>

Function	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
merge	Errors	Use <code>timetable</code> and <code>unique</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>merge</code>
min	Removed	Use <code>timetable</code> and MATLAB <code>min</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>min</code>
minus	Removed	There is no replacement function.	Remove all instances of <code>minus</code>
mrdivide	Removed	There is no replacement function.	Remove all instances of <code>mrdivide</code>
mtimes	Removed	There is no replacement function.	Remove all instances of <code>mtimes</code>
nancov	Removed	There is no replacement function.	Remove all instances of <code>nancov</code>
nanmax	Removed	There is no replacement function.	Remove all instances of <code>nanmax</code>
nanmean	Removed	There is no replacement function.	Remove all instances of <code>nanmean</code>
nanmedian	Removed	There is no replacement function.	Remove all instances of <code>nanmedian</code>
nanmin	Removed	There is no replacement function.	Remove all instances of <code>nanmin</code>
nanstd	Removed	There is no replacement function.	Remove all instances of <code>nanstd</code>
nansum	Removed	There is no replacement function.	Remove all instances of <code>nansum</code>
nanvar	Removed	There is no replacement function.	Remove all instances of <code>nanvar</code>
peravg	Removed	Use <code>timetable</code> and <code>retime</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>peravg</code>
plot	Removed	There is no replacement function.	Remove all instances of <code>plot</code>
plus	Removed	There is no replacement function.	Remove all instances of <code>plus</code>
power	Removed	There is no replacement function.	Remove all instances of <code>power</code>
resamplets	Removed	Use <code>timetable</code> and <code>retime</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>resamplets</code>
ret2tick (fts)	Removed	Use <code>ret2tick</code> .	Remove all instances of <code>ret2tick (fts)</code>
rmfield	Removed	There is no replacement function. Use <code>timetable</code> instead. For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>rmfield</code>
rsindex	Removed	There is no replacement function.	Remove all instances of <code>rsindex</code>

Function	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
<code>setfield</code>	Removed	There is no replacement function.	Remove all instances of <code>setfield</code>
<code>size</code>	Removed	There is no replacement function.	Remove all instances of <code>size</code>
<code>smoothts</code>	Removed	Use <code>timetable</code> and <code>smoothdata</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>smoothts</code>
<code>sortfts</code>	Removed	Use <code>timetable</code> and <code>sortrows</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of <code>sortfts</code>
<code>std</code>	Removed	Use <code>timetable</code> and MATLAB <code>std</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”.	Remove all instances of Financial Toolbox <code>std</code>
<code>subsasgn</code>	Removed	There is no replacement function.	Remove all instances of <code>subsasgn</code>
<code>subsref</code>	Removed	There is no replacement function.	Remove all instances of <code>subsref</code>
<code>tick2ret</code> (fts)	Removed	Use <code>tick2ret</code> .	Remove all instances of <code>tick2ret</code> (fts)
<code>times</code>	Removed	There is no replacement function.	Remove all instances of <code>times</code>
<code>toannual</code>	Removed	Use <code>convert2annual</code> .	Remove all instances of <code>to annual</code> and use <code>convert2annual</code>
<code>todayly</code>	Removed	Use <code>convert2daily</code>	Remove all instances of <code>todayly</code> and use <code>convert2daily</code>
<code>tomonthly</code>	Removed	Use <code>convert2monthly</code>	Remove all instances of <code>tomonthly</code> and use <code>convert2monthly</code>
<code>toquarterly</code>	Removed	Use <code>convert2quarterly</code>	Remove all instances of <code>toquarterly</code> and use <code>convert2quarterly</code>
<code>tosemi</code>	Removed	Use <code>convert2semiannual</code>	Remove all instances of <code>tosemi</code> and use <code>convert2semiannual</code>
<code>toweekly</code>	Removed	Use <code>convert2weekly</code>	Remove all instances of <code>toweekly</code> and use <code>convert2weekly</code>
<code>tsmovavg</code>	Removed	Use <code>movavg</code> instead.	Remove all instances of <code>tsmovavg</code>
<code>uminus</code>	Removed	There is no replacement function.	Remove all instances of <code>uminus</code>
<code>uplus</code>	Removed	There is no replacement function.	Remove all instances of <code>uplus</code>
<code>var</code>	Removed	Use <code>timetable</code> and MATLAB <code>var</code> . For more information, see “Convert Financial Time Series Objects (fints) to Timetables”	Remove all instances of Financial Toolbox <code>var</code>

Function	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
vertcat	Removed	Use <code>timetable</code> and MATLAB <code>vertcat</code> . For more information, see “Convert Financial Time Series Objects ( <code>fints</code> ) to Timetables”.	Remove all instances of Financial Toolbox <code>vertcat</code>

## fints support removed for input data for technical indicators

The following functions have changed and no longer support a `fints` object as an input.

### Compatibility Considerations

Function Name	What Happens When You Use This Function?	Use This Function Instead	Compatibility Considerations
<code>adline</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>adline</code> .
<code>adosc</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>adosc</code> .
<code>bar</code> , <code>barh</code>	Removed	Use MATLAB <code>bar</code> or <code>barh</code> .	Remove all instances of <code>bar</code> , <code>barh</code> .
<code>bar3</code> , <code>bar3h</code>	Removed	Use MATLAB <code>bar3</code> or <code>bar3h</code> .	Remove all instances of <code>bar3</code> , <code>bar3h</code> .
<code>bolling</code>	Removed	Use MATLAB <code>bollinger</code> .	Remove all instances of <code>bolling</code> .
<code>bollinger</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series	Remove all <code>fints</code> objects as data input to <code>bollinger</code> .
<code>boxcox</code>	Errors when using <code>fints</code> object as input	Use a <code>timetable</code> object and then use <code>timetable2table</code> and <code>table2array</code> .	Remove all <code>fints</code> objects as data input to <code>boxcox</code> .
<code>chaikosc</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>chaikosc</code> .
<code>chaikvolat</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>chaikvolat</code> .
<code>fpctkd</code>	Removed	Use <code>stochosc</code> .	Remove all instances of <code>fpctkd</code> .

<b>Function Name</b>	<b>What Happens When You Use This Function?</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
hhigh	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>hhigh</code> .
llow	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>llow</code> .
macd	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>macd</code> .
medprice	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>medprice</code> .
negvalidx	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>negvalidx</code> .
onbalvol	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>onbalvol</code> .
posvalidx	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>posvalidx</code> .
prcroc	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>prcroc</code> .
pvtrend	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>pvtrens</code> .
rsindex	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series	Remove all <code>fints</code> objects as data input to <code>rsindex</code> .
<code>simulateNormalScenariosByData</code>	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>simulateNormalScenariosByData</code> .
spctdk	Removed	Use <code>stochosc</code> .	Remove all instances of <code>spctdk</code> .
stochosc	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>stochosc</code> .
tsaccel	Errors when using <code>fints</code> object as input	Use a matrix, <code>timetable</code> , or <code>table</code> instead for data input for financial time series.	Remove all <code>fints</code> objects as data input to <code>tsaccel</code> .



<b>Function Name</b>	<b>What Happens When You Use This Function?</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
tsmom	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series	Remove all fints objects as data input to tsmom.
typprice	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series.	Remove all fints objects as data input to typprice.
volroc	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series.	Remove all fints objects as data input to volroc.
wclose	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series.	Remove all fints objects as data input to wclose.
willad	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series	Remove all fints objects as data input to willad.
willpctr	Errors when using fints object as input	Use a matrix, timetable, or table instead for data input for financial time series.	Remove all fints objects as data input to willpctr.



# R2022b

---

**Version: 6.4**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## SDE Framework: Perform Brownian Bridge and Principal Component constructions

You can perform Brownian bridge construction using the name-value argument for `BrownianMotionMethod` with the following stochastic differential equation (SDE) objects and their associated methods:

Class Names	Methods Supporting Brownian Bridge Construction
bates	simByEuler, simByTransition, simByQuadExp, simulate
bm	simByEuler, simulate
cev	simByEuler, simulate
cir	simByEuler, simByTransition, simByQuadExp, simulate
gbm	simByEuler, simBySolution, simulate
heston	simByTransition, simByQuadExp, simulate
hvw	simBySolution, simulate
merton	simBySolution, simByEuler, simulate
sde	simByEuler, simulate
sdeddo	simByEuler, simulate
sdeId	simByEuler, simulate
sdemrd	simByEuler, simulate

## Backtesting Framework: Management and performance fees

The `backtestStrategy` object supports name-value arguments for `ManagementFee`, `ManagementFeeSchedule`, `PerformanceFee`, `PerformanceHurdle`, and `PerformanceFeeSchedule`. In addition, the `backtestEngine` object supports a read-only property for `Fees` that contains the paid fees for the management and performance fees that you can specify with the `backtestStrategy` object.

## Portfolio Management: Custom objective functions

Estimate the optimal portfolio with a user-defined objective function using `estimateCustomObjectivePortfolio`. For an examples using `estimateCustomObjectivePortfolio`, see the following:

- Diversify Portfolios Using Custom Objective
- Portfolio Optimization Using Social Performance Measure
- Portfolio Optimization Against a Benchmark
- Solve Problem for Minimum Variance Portfolio with Tracking Error Penalty
- Solve Problem for Minimum Tracking Error with Net Return Constraint
- **Solve Robust Portfolio Maximum Return Problem with Ellipsoidal Uncertainty**

- 
- Risk Parity or Budgeting with Constraints

## **Portfolio Management: Brinson model for performance attribution**

The `brinsonAttribution` object implements the Brinson model for portfolio performance attribution. After creating a `brinsonAttribution` object, you can use the following functions:

- `categoryAttribution`
- `categoryReturns`
- `categoryWeights`
- `totalAttribution`
- `summary`

For an example of the workflow using a `brinsonAttribution` object with the associated functions, see [Analyze Performance Attribution Using Brinson Model](#).

## **Portfolio Management Example: Diversification of ESG portfolios**

The Diversify ESG Portfolios example shows how to include qualitative factors for environmental, social, and corporate governance (ESG) in the portfolio selection process.

## **Technical Indicators and Charting: Support for negative prices**

The following technical indicators and charting functions accept negative price data:

- `adline`
- `adosc`
- `bollinger`
- `candle`
- `chaikosc`
- `chaikvolat`
- `highlow`
- `hhigh`
- `kagi`
- `linebreak`
- `llo`
- `macd`
- `medprice`
- `negvolidx`
- `onbalvol`
- `pointfig`
- `posvolidx`
- `prcroc`
- `priceandvol`

- pvtrend
- renko
- ret2tick
- rsindex
- stochosc
- tick2ret
- tsaccel
- tsmom
- typprice
- volarea
- volroc
- wclose
- willad
- willpctr

## ftstool and ftsgui removal

ftstool and ftsgui have been removed. There is no replacement for these functions.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ftstool	Errors	There is no replacement function.	Remove all instances of ftstool.
ftsgui	Errors	There is no replacement function.	Remove all instances of ftsgui.

# R2022a

---

**Version: 6.3**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## SDE Framework: Perform Quasi-Monte Carlo simulation

You can perform Quasi-Monte Carlo simulation using the name-value arguments for `MonteCarloMethod` and `QuasiSequence` for the following stochastic differential equation (SDE) objects and their associated methods:

Class Names	Methods Supporting Quasi-Monte Carlo Simulation
<code>sde</code>	<code>simByEuler</code>
<code>bm</code>	<code>simByEuler</code>
<code>gbm</code>	<code>simByEuler</code> , <code>simBySolution</code>
<code>merton</code>	<code>simBySolution</code>
<code>bates</code>	<code>simByTransition</code> , <code>simByQuadExp</code>
<code>hvw</code>	<code>simBySolution</code>
<code>heston</code>	<code>simByTransition</code> , <code>simByQuadExp</code>
<code>cev</code>	<code>simByEuler</code>
<code>cir</code>	<code>simByEuler</code> , <code>simByTransition</code> , <code>simByQuadExp</code>
<code>sdeddo</code>	<code>simByEuler</code>
<code>sdeId</code>	<code>simByEuler</code>
<code>sdemrd</code>	<code>simByEuler</code>

## Backtesting Framework: Backtesting enhancements

The backtesting framework has the following enhancements:

- Specify time varying cash rates of return.

The `backtestEngine` name-value arguments for `RiskFreeRate` and `CashBorrowRate` support a `timetable` data type.

- Control how the backtesting framework handles missing rebalance dates.

The `backtestEngine` name-value argument for `'DateAdjustment'` enables you to control the date handling behavior for rebalance dates that are missing from the asset prices timetable. If a rebalance date falls on a holiday, you can specify the `"Next"` or `"None"` option for the name-value argument for `'DateAdjustment'`.

- Include NaN values in their asset price data.

The backtest framework supports NaNs in the `assetPrices` timetable and NaNs and `<missing>` in the `signalData` timetable.



---

## Backtesting Framework Example: Backtest with Deep Learning strategies

The Backtest Strategies Using Deep Learning example shows how to construct trading strategies using a Deep Learning Toolbox model and then backtest the strategies using the backtesting framework.

## Portfolio Management: Manage a risk parity portfolio

There are two functions for managing a risk parity portfolio:

- The `portfolioRiskContribution` function enables you to compute individual asset risk contribution to the overall portfolio volatility.
- The `riskBudgetingPortfolio` function enables you to compute risk budgeting portfolios.

The following examples demonstrate using `portfolioRiskContribution` and `riskBudgetingPortfolio` to manage a risk parity portfolio:

- The Risk Budgeting Portfolio example demonstrates how to use `riskBudgetingPortfolio` to create a risk budgeting portfolio and `portfolioRiskContribution` to compute the risk contribution of the assets in the portfolio.
- The Backtest Using Risk-Based Equity Indexation example shows how to use backtesting for a risk parity or equal risk contribution strategy.
- The Create Hierarchical Risk Parity Portfolio example shows how to use a hierarchical risk parity (HRP) technique for a risk parity allocation strategy.

## Portfolio Management: Portfolio enhancements

There are the following enhancements:

- The new name-value argument `'InitialDelta'` is added to `setSolverMINLP` and `setSolver` for the solver option `'TrustRegionCP'`.
- The new name-value argument `'TolX'` is added to `estimateMaxSharpeRatio` to expose `fminbnd` stop tolerance for the `'iterative'` method.

## Data Transformation: Negative prices example

The Financial Toolbox functions `tick2ret` and `ret2tick` support negative prices. This capability has been available since the release of these functions. The Returns with Negative Prices example shows how the functions mathematically treat negative price inputs. Also, the example shows how to interpret results from computations involving negative prices.

## Date and Time: Move of date and time functions to MATLAB

The following date and time functions are removed from Financial Toolbox and moved to MATLAB:

- `day`
- `eomdate`
- `hour`

- lweekdate
- m2xdate
- minute
- month
- months
- nweekdate
- quarter
- second
- today
- weeknum
- x2mdate
- year

# R2021b

---

**Version: 6.2**

**New Features**

**Bug Fixes**

## **Portfolio Management Example: Portfolio optimization using social criteria constraints**

The Portfolio Optimization Using a Social Performance Measure example uses a `Portfolio` object for portfolio optimization with group constraints and a social performance measure for the percentage of women on a company's board.

## **Portfolio Management Example: Portfolio diversification**

The Diversification of Portfolios example uses a `Portfolio` object for three common diversification techniques: equally weighted (EW) portfolio, equal risk contribution (ERC), and most diversified portfolio (MDP).

## **Portfolio Management: Robustness improvements for MINLP solvers**

The mixed integer nonlinear programming (MINLP) solver (`setSolverMINLP`) has changes that improve robustness for portfolio optimization. The changes include:

- Addition of feasibility cuts for the outer approximation algorithm
- Scaling of nonlinear subproblems to improve the numerical stability of the problem
- Tracking of the best feasible solution observed throughout the iterations of the algorithm

## **Reinforcement Learning Example: Price and hedge instruments using Reinforcement Learning Toolbox**

The Hedging an Option Using Reinforcement Learning Toolbox example demonstrates an approach to learn an optimal option hedging policy that outperforms the traditional Black-Scholes-Merton (BSM) model approach.

# R2021a

---

**Version: 6.1**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Backtesting Framework: Plot equity curves for each investment strategy

After running `runBacktest`, you can use `equityCurve` to plot the equity curves of each strategy defined with `backtestStrategy`.

## Backtesting Framework: Specify different frequencies for backtesting from datetime, duration, or calendarDuration arrays

The following frequencies are supported:

- For `runBacktest`, in addition to integers, the 'Start' and 'End' name-value pair arguments support a `datetime` object.
- For `backtestStrategy`, in addition to an integer, the 'RebalanceFrequency' name-value pair argument supports a scalar `duration` object, `calendarDuration` object, or a vector of `datetime` objects.
- For `backtestStrategy`, in addition to an integer, the 'LookbackWindow' name-value pair argument supports a `duration` object or `calendarDuration` object.

## Backtesting Framework: Define cash borrowing rate and risk free rate as an annualized rate

`backtestEngine` supports name-value pairs for 'RatesConvention' and 'Basis' to specify annualized rates for 'CashBorrowRate' or 'RiskFreeRate' name-value pair arguments.

## Compatibility Considerations

Function	Compatibility Considerations
<code>backtestEngine</code> using the name-value argument for 'RatesConvention' with a value of "Annualized"	The backtest engine interpretation of the two interest rates ( <code>RiskFreeRate</code> and <code>CashBorrowRate</code> ) is changing in R2021a. In R2020b, these arguments were "per row" or "per time step" rates, meaning the specified rates were accrued, in full, at each time step of the backtest in the cash account. For R2021a, the default behavior for <code>RiskFreeRate</code> and <code>CashBorrowRate</code> is to interpret the rate as annualized when the name-value pair argument 'RatesConvention' is set to "Annualized". This setting means the backtest engine computes the year fraction between each time step in the backtest and computes an incremental interest rate for that period based on an annualized rate. Therefore, you no longer need to divide your annualized rates into "per time step" rates based on the frequency of your pricing data.

To revert back to the R2020b workflow, on the `backtestEngine` object, set the 'RatesConvention' name-value argument to "PerStep".

## Portfolio Management Example: Hedging strategies using CVaR portfolio optimization

The Hedging Using CVaR Portfolio Optimization example uses simulated asset scenarios to model two hedging strategies using CVaR portfolio optimization with a `PortfolioCVaR` object. This example

---

then compares the asset selection between a mean-variance portfolio (using a `Portfolio` object) and the CVaR portfolio (using a `PortfolioCVaR` object).

## Financial Time Series: Aggregate timetable data for different periodicities ranging from daily to annually

You can aggregate timetable data for the following periodicities:

- `convert2daily`
- `convert2weekly`
- `convert2monthly`
- `convert2quarterly`
- `convert2semiannual`
- `convert2annual`

### Compatibility Considerations

Function	What Happens When You Use This Function	Compatibility Considerations
<code>todayly</code>	Warns	Use <code>convert2daily</code> instead.
<code>towekly</code>	Warns	Use <code>convert2weekly</code> instead.
<code>tomonthly</code>	Warns	Use <code>convert2monthly</code> instead.
<code>toquarterly</code>	Warns	Use <code>convert2quarterly</code> instead.
<code>tosemi</code>	Warns	Use <code>convert2semiannual</code> instead.
<code>toannual</code>	Warns	Use <code>convert2annual</code> instead.





# R2020b

---

**Version: 6.0**

**New Features**

**Bug Fixes**

## **Backtesting Workflow: Define investment strategies, run backtests, and summarize results**

You can define backtesting strategies using a `backtestStrategy` object and you can create a backtesting engine using a `backtestEngine` object. You can run and analyze the backtesting results with the `runBacktest` and `summary` functions. For more information on the backtesting workflow, see [Backtest Investment Strategies](#) and [Backtest Investment Strategies with Trading Signals](#).

## **Stochastic Differential Equation Models: Transition density simulation for Heston and Bates models**

SDE supports transition density simulation for Heston (`simByTransition`) and Bates (`simByTransition`) models.

## **Financial Time Series: Calculate period-over-period rolling returns with business calendars**

You can create period-over-period returns with `rollingreturns`. You can add business calendar support for `rollingreturns` using the `addBusinessCalendar` function.

# R2020a

---

**Version: 5.15**

**New Features**

**Bug Fixes**

## **Stochastic Differential Equation Models: Perform Monte Carlo simulation for the Bates and Merton jump diffusion models**

SDE support for new objects for Bates and Merton models.

- `bates`
- `merton`
- `simByEuler` for Bates
- `simByEuler` for Merton
- `simBySolution` for Merton

## **Stochastic Differential Equation Models: Simulate Bates, Heston, CIR sample paths by Quadratic-Exponential discretization scheme**

SDE supports a `simByQuadExp` function for Heston, Bates and CIR models.

## **Credit Scorecards: Replace a missing value in credit scorecard predictors with mean, median, mode or a custom value**

Fill missing data (NaN or `<missing>`) for a `creditscorecard` object with `fillmissing`. You can specify one or more predictors with missing values and replace the missing value with a fill value. The fill value can be mean, median, mode or a custom value. For additional information on alternative approaches for "treating" missing data, see [Credit Scorecard Modeling with Missing Values](#).

## **Machine Learning Examples: Series of examples on machine learning for statistical arbitrage**

Financial Toolbox has a series of examples that provide a general workflow for an algorithmic trading strategy that applies machine learning techniques to "big data". The examples analyze the intraday evolution of the limit order book (LOB) of a NASDAQ security with the goal of identifying real-time arbitrage opportunities.

- Machine Learning for Statistical Arbitrage: Introduction
- Machine Learning for Statistical Arbitrage I: Data Management and Visualization
- Machine Learning for Statistical Arbitrage II: Feature Engineering and Model Development
- Machine Learning for Statistical Arbitrage III: Training, Tuning, and Prediction

# R2019b

---

**Version: 5.14**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## **Portfolio Management: Perform CVaR and MAD portfolio optimization using nonlinear solvers (TrustRegionCP and ExtendedCP)**

TrustRegionCP and ExtendedCP are two new nonlinear programming solvers. As internal solvers, they provide faster performance when you estimate the efficient frontier of PortfolioCVaR and PortfolioMAD objects. Both solvers are based on the cutting plane method with effective heuristics. For details, see `setSolver`.

The TrustRegionCP solver has better performance than the `fmincon` solver and TrustRegionCP is now the default `solverType` for PortfolioCVaR and PortfolioMAD objects. You can configure the `solverType` and tune the solver options with `setSolver`.

## **Compatibility Considerations**

The default solver for PortfolioCVaR and PortfolioMAD objects has changed from 'fmincon' to 'TrustRegionCP'. To obtain the previous behavior, use `setSolver` to configure the `solverType` to 'fmincon'.

## **Portfolio Management Example: Optimize the portfolio using a factor model**

A new example of portfolio optimization using a factor model demonstrates two approaches: a problem-based approach using Optimization Toolbox™ and a portfolio optimization approach using a Portfolio object (see Portfolio Optimization Using Factor Models).

## **Transition Probability Estimates: Exclude 'NR' rating from transition matrix**

Use the new name-value pair argument 'excludeLabels' to exclude 'NR' or 'Not Rated' labels from the transition probability computation when you use `transprob`.

## **Functionality being removed or changed**

If you use `displaypoints` with a `creditscorecard` object, the new default behavior is to display a <missing> row for all predictors.

---

## Compatibility Considerations

Function	What Happens When You Use This Function	Compatibility Considerations
displaypoints	Displays a <missing> row for all predictors	<p>displaypoints always displays a &lt;missing&gt; bin for each predictor. The value of the &lt;missing&gt; bin comes from the initial creditscorecard object, and the &lt;missing&gt; bin is set to NaN when the scorecard model has no information on how to assign points to missing data.</p> <p>To configure the points for the &lt;missing&gt; bin, you must use the initial creditscorecard object. For predictors that have missing values in the training set, the points for the &lt;missing&gt; bin are estimated from the data if the 'BinMissingData' name-value pair argument is set to true using creditscorecard. When the 'BinMissingData' parameter is set to false, or when the data contains no missing values in the training set, use the 'Missing' name-value pair argument in formatpoints to indicate how to assign points to the missing data.</p>





# R2019a

---

**Version: 5.13**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Credit Scorecards: Specify equality, inequality, or bound constraints using `fitConstrainedModel`

Use `fitConstrainedModel` to specify equality, inequality, or bound constraints to fit a logistic regression model to the WOE data from a `creditscorecard` object.

## Portfolio Management: Perform CVaR and MAD portfolio optimization with integrality constraints such as minimum and maximum number of assets

The `PortfolioCVaR` and `PortfolioMAD` objects support setting two types of bound constraints, 'Simple' and 'Conditional', for each asset using `setBounds`. The `PortfolioCVaR` and `PortfolioMAD` objects also support setting constraints on the minimum and maximum number of allocated assets using `setMinMaxNumAssets`.

The `PortfolioCVaR` and `PortfolioMAD` objects have three new properties:

- `BoundType`
- `MinNumAssets`
- `MaxNumAssets`

In addition, the `PortfolioCVaR` and `PortfolioMAD` objects have two new functions:

- `setMinMaxNumAssets` specifies the minimum and maximum number of assets allocated, also known as cardinality constraints.
- `setSolverMINLP` configures the preferred MINLP solver and its options to solve the mixed integer nonlinear programming problems.

The updated `setBounds` function now supports 'Simple' and 'Conditional' bounds with a name-value pair for 'BoundType' for `PortfolioCVaR` and `PortfolioMAD` objects. The 'Conditional' type is used for semicontinuous constraints.

By default, the following `estimate` functions account for the `BoundType`, `MinNumAssets`, and `MaxNumAssets` constraints and generate solutions accordingly when you use `PortfolioCVaR` or `PortfolioMAD` objects:

- `estimateFrontier`
- `estimateFrontierByReturn`
- `estimateFrontierByRisk`
- `estimateFrontierLimits`
- `plotFrontier`

For an example of how to use these new constraints, see [Portfolio Optimization with Semicontinuous and Cardinality Constraints](#).

## Portfolio Management: Configure solver options for the solver `linprog` for portfolio optimization

You can configure the 'linprog' solver options for a `Portfolio`, `PortfolioCVaR`, or `PortfolioMAD` object.

Use the `solverType` input argument for `setSolver` with the `'linprog'` option along with the associated name-value pair arguments for `linprog`. You can use the `'linprog'` solver as a helper solver in the portfolio optimization workflow for a `Portfolio`, `PortfolioCVaR`, or `PortfolioMAD` object.

## Compatibility Considerations

The default setting for the helper solver `linprog` has changed from `'interior-point'` to `'dual-simplex'`. To obtain the previous behavior, use `setSolver` to configure the `'Algorithm'` of solver options to `'interior-point'`.

## Portfolio Management Example: Optimize the portfolio using the Black-Litterman model

Example of Black-Litterman portfolio optimization using a `Portfolio` object (see Black-Litterman Portfolio Optimization).

## Functionality being removed or changed

If you use timetables with `ret2tick` or `tick2ret` with non-default `DimensionNames` metadata, you can no longer reference the default `Time` property.

## Compatibility Considerations

Function	What Happens When You Use This Function with <code>DimensionNames</code> Metadata	Use This Reference Instead	Compatibility Considerations
<code>tick2ret</code> or <code>ret2tick</code>	Can no longer reference the default <code>Time</code> property	Reference the corresponding name used for the input timetable	When you use a timetable with the <code>ret2tick</code> or <code>tick2ret</code> functions, if you explicitly change the names of metadata values from the default values, you can no longer reference the default values.  For example, if you change the first dimension name of an input timetable from the default <code>'Time'</code> to <code>'Date'</code> , then the first dimension name of the output timetable is also <code>'Date'</code> , and not <code>'Time'</code> . Any subsequent reference to <code>'Time'</code> generates an error.



# R2018b

---

**Version: 5.12**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Credit Scorecards: Bin missing data in a separate bin

When binning a `creditscorecard`, you can specify a separate bin for missing data for numeric or categorical predictors by specifying a name-value pair argument for `'BinMissingData'`. If `BinMissingData` is `true`, a separate bin labeled `<missing>` displays the missing data for each predictor. The following functions for the credit scorecard workflow now support a `<missing>` bin for numeric or categorical predictors:

- `autobinning`
- `bininfo`
- `modifybins`
- `bindata`
- `plotbins`
- `fitmodel`
- `displaypoints`
- `formatpoints`
- `score`
- `probdefault`
- `validatemodel`

For an example of how to work with missing data, see [Credit Scorecard Modeling with Missing Values](#).

## Portfolio Management: Perform mean-variance portfolio optimization with integrality constraints such as minimum and maximum number of assets

The `Portfolio` object supports setting two types of bound constraints, `'Simple'` and `'Conditional'`, for each asset using `setBounds`. The `Portfolio` object also supports setting constraints on the minimum and maximum number of allocated assets using `setMinMaxNumAssets`.

The `Portfolio` object has three new properties:

- `BoundType`
- `MinNumAssets`
- `MaxNumAssets`

In addition, the `Portfolio` object has two new functions:

- `setMinMaxNumAssets` specifies the minimum and maximum number of assets allocated, also known as cardinality constraints.
- `setSolverMINLP` configures the preferred MINLP solver and its options to solve the mixed integer nonlinear programming problems.

The updated `setBounds` function now supports `'Simple'` and `'Conditional'` bounds with a name-value pair for `'BoundType'`. The `'Conditional'` type is used for semicontinuous constraints.

By default, the following `estimate` functions account for the `BoundType`, `MinNumAssets`, and `MaxNumAssets` constraints and generate solutions accordingly:

- estimateFrontier
- estimateFrontierByReturn
- estimateFrontierByRisk
- estimateFrontierLimits
- estimateMaxSharpeRatio
- plotFrontier

For an example of how to use these new constraints with a Portfolio object, see Portfolio Optimization with Semicontinuous and Cardinality Constraints.

## Implied Volatility: Improve performance of the blsimpv and blkimpv functions when using the Jäckel 2016 method

Improve performance of blsimpv and blkimpv by using a new name-value pair argument for 'Method' with the value of 'search' or 'jackel2016'. For computing implied volatility, the default value is 'jackel2016'.

## Stochastic Differential Equation Models: Perform Monte Carlo simulation for the Cox-Ingersoll-Ross model

The cir object, which uses the Cox-Ingersoll-Ross model, supports Monte Carlo simulation with the functions simByTransition and simByEuler.

## Functionality being removed or changed

If you use modifybins with a creditscorecard object, modifybins does not support the name-value argument 'BinLabels' with the value <missing>.

## Compatibility Considerations

Bin Label Name	What Happens When You Use This Bin Label	Use This Bin Label Instead	Compatibility Considerations
<missing>	Errors	Use any text other than <missing> for a bin label.	When using modifybins, replace all instances of 'BinLabels' specified as <missing> with different label text.





# R2018a

---

**Version: 5.11**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## **Credit Scorecards: Bin creditscorecard data using supervised binning algorithms, including merge and split**

When binning a `creditscorecard`, you can specify two new algorithms for 'Split' and 'Merge' when using `autobinning`. Also `bininfo` supports 'Statistics' options for 'Gini' and 'Chi2'.

## **Portfolio Management: Improve performance for estimations of efficient frontiers and portfolios**

The following portfolio functions are updated to provide improved performance:

- `estimateMaxSharpeRatio`
- `estimateFrontierByRisk` for `Portfolio` or `PortfolioMAD` objects.

## **Sharpe Ratio of Portfolio: Estimate the Sharpe ratio of portfolio weights given custom portfolio weights**

The `estimatePortSharpeRatio` estimates the Sharpe ratio of the given portfolio weights.

## **Portfolio Management: Input table and timetable objects for `estimateAssetMoments`, `setScenarios`, and `simulateNormalScenariosByData`**

When using `estimateAssetMoments` with a `Portfolio` object, or `setScenarios` and `simulateNormalScenariosByData` with a `PortfolioCVaR` or `PortfolioMAD` objects, the `AssetReturns` argument accepts a MATLAB `table` or `timetable` function for a financial time series.

## **Financial Time Series: Input table and timetable objects for technical indicators and financial charts**

Financial technical indicators support a MATLAB `timetable` or `table` function as input for financial data. In addition, the following technical indicators support name-value pair arguments or optional arguments in their syntax.

- `adosc`
- `chaikosc`
- `macd`
- `stochosc`
- `tsaccel`
- `tsmom`
- `chaikvolat`
- `willpctr`
- `negvolidx`
- `posvolidx`

- 
- `rsindex`
  - `adline`
  - `bollinger`
  - `hhigh`
  - `llow`
  - `medprice`
  - `movavg`
  - `onbalvol`
  - `prcroc`
  - `pvtrend`
  - `typprice`
  - `volroc`
  - `wclose`
  - `willad`
  - `ret2tick`
  - `tick2ret`

Financial charts now support a MATLAB `timetable` or `table` function as input for financial data. In addition, the following financial charts now support name-value arguments and target axes in their syntax.

- `candle`
- `highlow`
- `kagi`
- `linebreak`
- `pointfig`
- `priceandvol`
- `renko`
- `volarea`

### **Financial Time Series: `fints` object removed**

The `fints` object will be removed in a future release. For more information, see [Convert Financial Time Series Objects `fints` to Timetables](#).

## Compatibility Considerations

Object Name	What Happens When You Use This Object	Use This MATLAB Function Instead	Compatibility Considerations
fints	Warns	timetable	Replace all instances of fints object with a timetable.  Use fts2timetable to convert a fints object to a timetable.

## Financial Time Series: fints-related functions removed

The following fints-related functions will be removed in a future release. For more information, see Convert Financial Time Series Objects fints to Timetables.

## Compatibility Considerations

fints-related Function Name	What Happens When You Use Function with fints Object	Use This Function Instead	Compatibility Considerations
ascii2fts	Warns	timetable	Remove all instances of ascii2fts. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
fts2ascii	Warns	timetable	Remove all instances of fts2ascii. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
fts2mat	Warns	timetable	Remove all instances of fts2mat. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
diff	Warns	timetable	Remove all instances of diff. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
fillts	Warns	timetable	Remove all instances of fillts. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
filter	Warns	timetable	Remove all instances of filter. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
lagts	Warns	timetable	Remove all instances of lagts. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
leadts	Warns	timetable	Remove all instances of leadts. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
peravg	Warns	timetable	Remove all instances of peravg. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
resamplets	Warns	timetable	Remove all instances of resamplets. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
convertto	Warns	timetable	Remove all instances of convertto. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
toannual	Warns	timetable	Remove all instances of toannual. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
todayly	Warns	timetable	Remove all instances of todayly. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
tomonthly	Warns	timetable	Remove all instances of tomonthly. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
toquarterly	Warns	timetable	Remove all instances of toquarterly. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
tosemi	Warns	timetable	Remove all instances of tosemi. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
toweekly	Warns	timetable	Remove all instances of toweekly. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
merge	Warns	timetable	Remove all instances of merge. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
cov	Warns	timetable	Remove all instances of cov. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nancov	Warns	timetable	Remove all instances of nancov. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
nanmax	Warns	timetable	Remove all instances of nanmax. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nanmean	Warns	timetable	Remove all instances of nanmean. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nanmedian	Warns	timetable	Remove all instances of nanmedian. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nanmin	Warns	timetable	Remove all instances of nanmin. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nanstd	Warns	timetable	Remove all instances of nanstd. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nansum	Warns	timetable	Remove all instances of nansum. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
nanvar	Warns	timetable	Remove all instances of nanvar. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
var	Warns	timetable	Remove all instances of var. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
cumsum	Warns	timetable	Remove all instances of cumsum. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
exp	Warns	timetable	Remove all instances of exp. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
hist	Warns	timetable	Remove all instances of hist. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
log	Warns	timetable	Remove all instances of log. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
log10	Warns	timetable	Remove all instances of log10. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
log2	Warns	timetable	Remove all instances of log2. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
max	Warns	timetable	Remove all instances of max. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
mean	Warns	timetable	Remove all instances of mean. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.



<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
min	Warns	timetable	Remove all instances of min. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
std	Warns	timetable	Remove all instances of std. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
freqnum	Warns	timetable	Remove all instances of freqnum. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
freqstr	Warns	timetable	Remove all instances of freqstr. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
ftsbound	Warns	timetable	Remove all instances of ftsbound. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
ftsuniq	Warns	timetable	Remove all instances of ftsuniq. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
iscompatible	Warns	timetable	Remove all instances of iscompatible. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
issorted	Warns	timetable	Remove all instances of issorted. Convert fints object to a timetable using fts2timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
sortfts	Warns	sort	Replace all instances of sortfts with the MATLAB sort function. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
times	Warns	times	Replace all instances of times with the MATLAB times function. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
corrcoef	Warns	corrcoef	Replace all instances of corrcoef with the MATLAB corrcoef function. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
candle (fts)	Warns	candle	Convert fints object to a timetable using fts2timetable. Replace all instances of candle (fts) with candle.
highlow (fts)	Warns	highlow	Convert fints object to a timetable using fts2timetable. Replace all instances of highlow (fts) with highlow.
fpctkd	Warns	stochosc	Convert fints object to a timetable using fts2timetable. Replace all instances of fpctkd with stochosc.
spctkd	Warns	stochosc	Convert fints object to a timetable using fts2timetable. Replace all instances of spctkd with stochosc.
getnameidx	Warns	contains	Convert fints object to a timetable using fts2timetable. Replace all instances of getnameidx with the MATLAB contains function.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
bar, barh	Warns	bar or barh	Replace all instances of bar, barh with the MATLAB bar or barh functions. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
bar3, bar3h	Warns	bar3 or bar3h	Replace all instances of bar3, bar3h with the MATLAB bar3 or bar3h functions. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
bolling	Warns	bollinger	Replace all instances of bolling with bollinger. Convert fints object to a timetable using fts2timetable.
plot	Warns	plot	Replace all instances of plot with the MATLAB plot function. Convert fints object to a timetable using fts2timetable, then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
smoothts	Warns	smoothdata	Replace all instances of smoothts with smoothdata. Convert fints object to a timetable using fts2timetable.
tsmovavg	Warns	movavg	Replace all instances of tsmovavg with movavg. Convert fints object to a timetable using fts2timetable.
ret2tick (fts)	Warns	ret2tick	Replace all instances of ret2tick (fts) with ret2tick. Convert fints object to a timetable using fts2timetable.
tick2ret (fts)	Warns	tick2ret	Replace all instances of tick2ret (fts) with tick2ret. Convert fints object to a timetable using fts2timetable.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
boxcox	Warns	boxcox	Replace all instances of a fints object for input with an array by using <code>fts2timetable</code> to convert a fints object to a timetable and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
horzcat	Still runs	horzcat	Replace all instances of <code>horzcat</code> with the MATLAB <code>horzcat</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable.
vertcat	Still runs	vertcat	Replace all instances of <code>vertcat</code> with the MATLAB <code>vertcat</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable.
isempty	Still runs	isempty	Replace all instances of <code>isempty</code> with the MATLAB <code>isempty</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable.
end	Still runs	end	Replace all instances of <code>end</code> with the MATLAB <code>end</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable.
length	Still runs	height	Replace all instances of <code>length</code> with <code>height</code> . Use <code>fts2timetable</code> to convert a fints object to a timetable.
minus	Still runs	minus	Replace all instances of <code>minus</code> with the MATLAB <code>minus</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable. For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
mrdivide	Still runs	mrdivide	Replace all instances of <code>mrdivide</code> with the MATLAB <code>mrdivide</code> function. Use <code>fts2timetable</code> to convert a fints object to a timetable. For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
<code>mtimes</code>	Still runs	<code>mtimes</code>	Replace all instances of <code>mtimes</code> with the MATLAB <code>mtimes</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
<code>plus</code>	Still runs	<code>plus</code>	Replace all instances of <code>plus</code> with the MATLAB <code>plus</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
<code>power</code>	Still runs	<code>power</code>	Replace all instances of <code>power</code> with the MATLAB <code>power</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
<code>rdivide</code>	Still runs	<code>rdivide</code>	Replace all instances of <code>rdivide</code> with the MATLAB <code>rdivide</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
<code>size</code>	Still runs	<code>size</code>	Replace all instances of <code>size</code> with the MATLAB <code>size</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>subsasgn</code>	Still runs	<code>subsasgn</code>	Replace all instances of <code>subsasgn</code> with the MATLAB <code>subsasgn</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> , and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
<code>subsref</code>	Still runs	<code>subsref</code>	Replace all instances of <code>subsref</code> with the MATLAB <code>subsref</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> , and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
extfield	Still runs	timetable	Replace all instances of extfield. Use fts2timetable to convert a fints object to a timetable. For more information, see Convert Financial Time Series Objects fints to Timetables.
uminus	Still runs	uminus	Replace all instances of uminus with the MATLAB uminus function. Use fts2timetable to convert a fints object to a timetable, and then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
uplus	Still runs	uplus	Replace all instances of uplus with the MATLAB uplus function. Use fts2timetable to convert a fints object to a timetable, and then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.
chfield	Still runs	timetable	Replace all instances of chfield. Use fts2timetable to convert a fints object to a timetable.
eq (fts)	Still runs	eq	Replace all instances of eq (fts) with the MATLAB eq function. Use fts2timetable to convert a fints object to a timetable.
fetch	Still runs	timetable	Replace all instances of fetch. Use fts2timetable to convert a fints object to a timetable.
fieldnames	Still runs	fieldnames	Replace all instances of fieldnames with the MATLAB fieldnames function. Use fts2timetable to convert a fints object to a timetable.
ftsinfo	Still runs	timetable	Replace all instances of ftsinfo. Use fts2timetable to convert a fints object to a timetable.
getfield	Still runs	getfield	Replace all instances of getfield with the MATLAB getfield function. Use fts2timetable to convert a fints object to a timetable, and then use timetable2table and table2array. For more information, see Convert Financial Time Series Objects fints to Timetables.

<b>fints-related Function Name</b>	<b>What Happens When You Use Function with fints Object</b>	<b>Use This Function Instead</b>	<b>Compatibility Considerations</b>
isequal	Still runs	isequal	Replace all instances of <code>isequal</code> with the MATLAB <code>isequal</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
isfield	Still runs	isfield	Replace all instances of <code>isfield</code> with the MATLAB <code>isfield</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> , and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
rmfield	Still runs	rmfield	Replace all instances of <code>rmfield</code> with the MATLAB <code>rmfield</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> , and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .
setfield	Still runs	setfield	Replace all instances of <code>setfield</code> with the MATLAB <code>setfield</code> function. Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> , and then use <code>timetable2table</code> and <code>table2array</code> . For more information, see <a href="#">Convert Financial Time Series Objects fints to Timetables</a> .

**fints object support removed from Portfolio, PortfolioCVaR, and PortfolioMAD objects**

The `fints` object (`tsobj`) support is removed from `Portfolio`, `PortfolioCVaR`, and `PortfolioMAD` objects.

## Compatibility Considerations

Function Name	What Happens When You Use This Function with a <code>fints</code> Object	Use This MATLAB Function Instead	Compatibility Considerations
<code>estimateAssetMoments</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>setScenarios</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>simulateNormalScenariosByData</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .

### Technical indicators support for `fints` removed

The technical indicators support for a `fints` object as an input argument is removed and replaced by a `timetable`. For more information, see [Convert Financial Time Series Objects `fints` to Timetables](#).

## Compatibility Considerations

Technical Indicator Function	What Happens When You Use This Function with a <code>fints</code> Object	Use This MATLAB Function Instead	Compatibility Considerations
<code>adosc</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>chaikosc</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .



<b>Technical Indicator Function</b>	<b>What Happens When You Use This Function with a fints Object</b>	<b>Use This MATLAB Function Instead</b>	<b>Compatibility Considerations</b>
macd	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
stochosc	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
tsaccel	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
tsmom	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
chaikvolat	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
willpctr	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
negvalidx	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.
posvalidx	Warns	timetable	Replace all instances of fints objects with a timetable.  Use fts2timetable to convert a fints object to a timetable.

Technical Indicator Function	What Happens When You Use This Function with a <code>fints</code> Object	Use This MATLAB Function Instead	Compatibility Considerations
rsindex	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
adline	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
bollinger	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
hhigh	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
lflow	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
medprice	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
onbalvol	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
prcroc	Warns	timetable	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .

Technical Indicator Function	What Happens When You Use This Function with a <code>fints</code> Object	Use This MATLAB Function Instead	Compatibility Considerations
<code>pvtrend</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>typprice</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>volroc</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>wclose</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
<code>willad</code>	Warns	<code>timetable</code>	Replace all instances of <code>fints</code> objects with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .

### User interface tools for data extraction supporting `fints` removed

User interface tools for data extraction supporting `fints` objects are removed. For more information, see [Convert Financial Time Series Objects `fints` to Timetables](#).

### Compatibility Considerations

User Interface Tool Name	What Happens When You Use This App	Use This MATLAB Function Instead	Compatibility Considerations
Financial Time Series app	Warns	<code>timetable</code>	Replace all instances of Financial Time Series app ( <code>ftstool</code> ) with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .

User Interface Tool Name	What Happens When You Use This App	Use This MATLAB Function Instead	Compatibility Considerations
Financial Time Series GUI	Warns	<code>timetable</code>	Replace all instances of Financial Time Series GUI ( <code>ftsgui</code> ) with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .
Interactive Chart	Warns	<code>timetable</code>	Replace all instances of Interactive Chart ( <code>chartfts</code> ) with a <code>timetable</code> .  Use <code>fts2timetable</code> to convert a <code>fints</code> object to a <code>timetable</code> .

### Charting functions support removed for optional inputs for 'Dates' and 'Dateform'

The following charting functions no longer accept optional inputs for 'Dates' and 'Dateform'. For more information, see Convert Financial Time Series Objects `fints` to Timetables.

### Compatibility Considerations

Charting Function	What Happens When You Use This Function with Optional Inputs for 'Dates' and 'Dateform'	Use This Syntax Instead	Compatibility Considerations
<code>candle</code>	Error	Use <code>timetable</code> , <code>table</code> , or a matrix for the data input argument.	Replace all instances of <code>fints</code> input with a <code>timetable</code> , <code>table</code> , or matrix.
<code>highlow</code>	Error	Use <code>timetable</code> , <code>table</code> , or a matrix for the data input argument.	Replace all instances of <code>fints</code> input with a <code>timetable</code> , <code>table</code> , or matrix.

# R2017b

---

**Version: 5.10**

**New Features**

**Bug Fixes**

## Credit Scorecards: Support weights in credit scorecards

You can specify weights by using an optional name-value pair argument `WeightsVar` when creating a `creditscorecard` object. Also, weights are explicitly supported for the following credit scorecard functions:

- `creditscorecard`
- `bininfo`
- `fitmodel`
- `validatemodel`

# R2017a

---

**Version: 5.9**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Default Probability Modeling: Bootstrap default probabilities from bonds using Jarrow-Turnbull model

Bootstrap default probabilities from bonds using Jarrow-Turnbull model with `bondDefaultBootstrap`.

## Credit Scorecards: Support strings in credit scorecard

Support for strings in the data input argument for credit scorecard functions: `creditscorecard`, `bindata`, `score`, `probdefault`, and `validatemodel`.

## Financial Time Series app removal for Database Toolbox support

When working with the Financial Time Series app, the **Database Toolbox** selection is removed from **File > Load**.

## Compatibility Considerations

Menu Item Name	What Happens When You Use This Menu Item	Use This Menu Item Instead	Compatibility Considerations
<b>File &gt; Load</b>	<b>Database Toolbox</b> selection is removed.	Financial Time Series app no longer supports an interface to Database Toolbox™.	Use Database Toolbox directly and export data to a file or the MATLAB workspace to then use with the Financial Time Series app.



# R2016b

---

**Version: 5.8**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Credit Scorecards: Last binning operation in creditscorecard

Credit scorecard supports information on the last binning operation when using `predictorinfo`. The `T` output argument for `predictorinfo` displays information on 'LastestBinning'.

## Functions moved to Financial Toolbox from Financial Instruments Toolbox

The following functions are moved to Financial Toolbox from Financial Instruments Toolbox™:

- `cdsbootstrap` calculates barrier option prices using finite difference method.
- `cdsprice` calculates barrier option prices and sensitivities using finite difference method.
- `cdspread` calculates price for a European barrier options using Black-Scholes option pricing model.
- `cdrpv01` calculates price and sensitivities for a European barrier options using Black-Scholes option pricing model.
- `creditlexposures` computes credit exposures from contract values.
- `exposureprofiles` computes exposure profiles from credit exposures.

## help findemos removal

The `help findemos` command is removed in this release. Use the `demo` command instead.

## Compatibility Considerations

Command Name	What Happens When You Use This Command	Use This Command Instead	Compatibility Considerations
<code>help findemos</code>	Errors	<code>demo 'toolbox' 'financial'</code>	Replace all instances of <code>help findemos</code> with <code>demo 'toolbox' 'financial'</code> .

# R2016a

---

**Version: 5.7**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Plots: Fan chart enhancements

`fanplot` accepts name value pair arguments to control chart colors and line sizes for the historical and forecast lines.

## Date and Time: `datetime` support for calendar functions

Support for `datetime` for the following calendar functions according to these guidelines:

- Functions that take date inputs and output dates. If any of the date inputs are `datetime` arrays, then the date outputs are returned as a `datetime`. Otherwise, the dates are returned as `datenums`.
  - Functions that take date inputs, but do not output dates. In this case, the function should return the same output whether the date inputs are `datenums` or `datetime`.
  - Functions that do not take in date inputs, but output dates. In this case, an extra optional input argument `outputType` is included that allows you to specify the output as a `'datenum'` or a `'datetime'`. The default behavior is `'datenum'`.
- `accrfrac`
  - `acrubond`
  - `acrudisc`
  - `beytbill`
  - `bndconvp`
  - `bndconvy`
  - `bnddurp`
  - `bnddury`
  - `bndkrdur`
  - `bndprice`
  - `bndspread`
  - `bndtotalreturn`
  - `bndyield`
  - `busdate`
  - `busdays`
  - `candle`
  - `cdai`
  - `cdprice`
  - `cdyield`
  - `cfamounts`
  - `cfdates`
  - `cfdatesq`
  - `cfplot`
  - `cfport`
  - `cfprice`

- 
- cfsread
  - cfyield
  - cftimes
  - cpncount
  - cpndaten
  - cpndatenq
  - cpndatepq
  - cpndatep
  - cpndaysn
  - cpndaysp
  - cpnpersz
  - dateaxis
  - date2time
  - datefind
  - datemnth
  - datewrkdy
  - days252bus
  - days360
  - days360e
  - days360isda
  - days360psa
  - days365
  - daysact
  - daysadd
  - daysdif
  - disc2zero
  - discrate
  - eomdate
  - fanplot
  - fbusdate
  - floatdiscmargin
  - floatmargin
  - fvdisc
  - fvvar
  - fwd2zero
  - highlow
  - holidays
  - isbusday
  - kagi

- lbusdate
- linebreak
- lweekdate
- m2xdate
- nweekdate
- nyseclosures
- periodicreturns
- prbyzero
- prdisc
- priceandvol
- prmat
- prtbill
- pvvar
- pyld2zero
- renko
- ret2tick
- tbilldisc2yield
- tbillprice
- tbillrepo
- tbillval01
- tbillyield
- tbillyield2disc
- tbl2bond
- thirdwednesday
- tick2ret
- time2date
- tmfactor
- today
- totalreturnprice
- tr2bonds
- uicalendar
- volarea
- weeknum
- wrkdydif
- x2mdate
- xirr
- yearfrac
- ylddisc
- yldmat

- `yldtbill`
- `zbtprice`
- `zbtyield`
- `zero2disc`
- `zero2fwd`
- `zero2pyld`

## Date and Time: function to return the quarter of a given date

Support for `quarter`. The purpose of this function is to return the quarter of a given date.

## Functionality Removed

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
<code>proddf</code>	Removed	<code>bndprice</code>	Replace all instances of <code>proddf</code> with <code>bndprice</code> .
<code>proddf1</code>	Removed	<code>bndprice</code>	Replace all instances of <code>proddf1</code> with <code>bndprice</code> .
<code>proddl</code>	Removed	<code>bndprice</code>	Replace all instances of <code>proddl</code> with <code>bndprice</code> .
<code>yldodd1</code>	Removed	<code>bndyield</code>	Replace all instances of <code>yldodd1</code> with <code>bndyield</code> .
<code>yldodd2</code>	Removed	<code>bndyield</code>	Replace all instances of <code>yldodd2</code> with <code>bndyield</code> .
<code>yldodd3</code>	Removed	<code>bndyield</code>	Replace all instances of <code>yldodd3</code> with <code>bndyield</code> .
<code>prbond</code>	Removed	<code>bndprice</code>	Replace all instances of <code>prbond</code> with <code>bndprice</code> .
<code>yldbond</code>	Removed	<code>bndyield</code>	Replace all instances of <code>yldbond</code> with <code>bndyield</code> .
<code>checksiz</code>	Removed	N/A	Remove all instances from your code.
<code>checktyp</code>	Removed	N/A	Remove all instances from your code.
<code>checkrng</code>	Removed	N/A	Remove all instances from your code.

## ugarch removal

`ugarch` is removed in this release. Use the `garch` object from the Econometrics Toolbox™ instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarch	Errors	estimate	Replace all instances of <code>ugarch</code> with the <code>garch</code> object to create conditional variance models and use the <code>estimate</code> function to fit conditional variance models to data.

For more information on migrating `ugarch` code to `garch`, see Likelihood Ratio Test for Conditional Variance Models.

### ugarchllf removal

`ugarchllf` is removed in this release. Use the `garch` object from the Econometrics Toolbox instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchllf	Errors	garch	Replace all instances of <code>ugarchllf</code> with <code>garch</code> .

For more information on migrating `ugarchllf` code to `garch`, see Specify GARCH Models Using `garch`.

### ugarchpred removal

`ugarchpred` is removed in this release. Use the `garch` object from the Econometrics Toolbox instead.



---

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchpred	Errors	forecast	Replace all instances of <code>ugarchpred</code> with the <code>garch</code> object to create conditional variance models and use the <code>forecast</code> function to generate minimum mean square error forecasts.

For more information on migrating `ugarchpred` code to `garch`, see Forecast a Conditional Variance Model.

### **ugarchsim removal**

`ugarchsim` is removed in this release. Use the `garch` object from the Econometrics Toolbox instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchsim	Errors	simulate	Replace all instances of <code>ugarchsim</code> with the <code>garch</code> object to create conditional variance models and use the <code>simulate</code> function to generate Monte Carlo simulations from conditional variance models.

For more information on migrating `ugarchsim` code to `garch`, see Simulate Conditional Variance Model.

### **frontcon removal**

`frontcon` has been removed. Use `Portfolio` instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
frontcon	Errors	Portfolio	Replace all instances of frontcon with Portfolio.

For more information on migrating frontcon code to Portfolio, see frontcon Migration to Portfolio Object.

## portopt partial removal

portopt has been partially removed and no longer accepts ConSet or varargin input arguments. In this release, a modified portopt only solves a portfolio problem for long-only fully invested portfolios. Use Portfolio instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
portopt	Error if ConSet or varargin input arguments are used.	Portfolio	If you want to solve a portfolio problem that is more than a long-only fully invested portfolio, replace all instances of portopt with Portfolio.

For more information on migrating portopt code to Portfolio, see portopt Migration to Portfolio Object.

# R2015b

---

**Version: 5.6**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Portfolio Optimization: Calculate mean-variance portfolios with tracking error constraint

Support for two new functions to set up tracking error constraints for a `Portfolio` object.

- `setTrackingPort` sets up tracking or benchmark portfolio for a tracking error constraint.
- `setTrackingError` sets up a maximum portfolio tracking error constraint.

## Credit Scorecards: Set predictor types to numeric or categorical and view summary information

Credit scorecard supports two new functions for reviewing and converting predictor types:

- `predictorinfo` provides a summary of credit scorecard predictors and their properties.
- `modifypredictor` enables you to set properties for credit scorecard predictors to change a predictor type from numeric to categorical or vice versa.

In addition, the `creditscorecard` object has two new properties, `NumericPredictors` and `CategoricalPredictors` which have public `GetAccess` and private `SetAccess`, that is, they cannot be set at the command line using the dot notation.

## Transition Probability Estimates: Enter data using table input

Support for MATLAB table input for `transprob` and `transprobbprep`.

## Simple Interest Convention: Calculate zero, forward, and discount curves using simple interest

Support for simple interest for the following functions:

- `zero2disc` — Support added for `Compounding = 0` for simple interest where there is no compounding.
- `disc2zero` — Support added for `Compounding = 0` for simple interest where there is no compounding.
- `zero2fwd` — Support added for `InputCompounding = 0` for simple interest where there is no compounding, and also `OutputCompounding = 0` for simple interest. See “Functionality Being Changed for `fwd2zero`, `zero2fwd`, `pyld2zero`, and `zero2pyld`” on page 16-3.
- `fwd2zero` — Support added for `InputCompounding = 0` for simple interest where there is no compounding, and also `OutputCompounding = 0` for simple interest. See “Functionality Being Changed for `fwd2zero`, `zero2fwd`, `pyld2zero`, and `zero2pyld`” on page 16-3.
- `date2time` — Support added for `Compounding = 0` for simple interest where there is no compounding.
- `zero2pyld` — Support added for `InputCompounding = 0` for simple interest where there is no compounding. See “Functionality Being Changed for `fwd2zero`, `zero2fwd`, `pyld2zero`, and `zero2pyld`” on page 16-3.
- `pyld2zero` — Support added for `OutputCompounding = 0` for simple interest where there is no compounding. See “Functionality Being Changed for `fwd2zero`, `zero2fwd`, `pyld2zero`, and `zero2pyld`” on page 16-3.

- `zbtprice` — Support added for `OutputCompounding = 0` for simple interest where there is no compounding.
- `zbtyield` — Support added for `OutputCompounding = 0` for simple interest where there is no compounding.

## Functionality Being Changed for `fwd2zero`, `zero2fwd`, `pyld2zero`, and `zero2pyld`

These functions now accept additional optional input arguments that are specified as name-value pairs: `InputCompounding`, `OutputCompounding`, `InputBasis`, and `OutputBasis`.

In addition, for `pyld2zero` and `zero2pyld`, the settings for the default behavior for optional name-value pairs inputs have changed. For more information, see the reference pages for `pyld2zero` and `zero2pyld`.

## `ugarch` removal

`ugarch` will be removed in a future release. Use the `garch` object from the Econometrics Toolbox instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
<code>ugarch</code>	Warns	<code>estimate</code>	Replace all instances of <code>ugarch</code> with the <code>garch</code> object to create conditional variance models and use the <code>estimate</code> function to fit conditional variance models to data.

For more information on migrating `ugarch` code to `garch`, see Likelihood Ratio Test for Conditional Variance Models.

## `ugarchl1f` removal

`ugarchl1f` will be removed in a future release. Use the `garch` object from the Econometrics Toolbox instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchllf	Warns	garch	Replace all instances of ugarchllf with garch.

For more information on migrating ugarchllf code to garch, see Specify GARCH Models Using garch.

### ugarchpred removal

ugarchpred will be removed in a future release. Use the garch object from the Econometrics Toolbox instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchpred	Warns	forecast	Replace all instances of ugarchpred with the garch object to create conditional variance models and use the forecast function to generate minimum mean square error forecasts.

For more information on migrating ugarchpred code to garch, see Forecast a Conditional Variance Model.

### ugarchsim removal

ugarchsim will be removed in a future release. Use the garch object from the Econometrics Toolbox instead.

---

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
ugarchsim	Warns	simulate	Replace all instances of <code>ugarchsim</code> with the <code>garch</code> object to create conditional variance models and use the <code>simulate</code> function to generate Monte Carlo simulations from conditional variance models.

For more information on migrating `ugarchsim` code to `garch`, see [Simulate Conditional Variance Model](#).

### frontcon removal

`frontcon` has been removed. Use `Portfolio` instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
frontcon	Removed	Portfolio	Replace all instances of <code>frontcon</code> with <code>Portfolio</code> .

For more information on migrating `frontcon` code to `Portfolio`, see [frontcon Migration to Portfolio Object](#).

### portopt partial removal

`portopt` has been partially removed and no longer accepts `ConSet` or `varargin` input arguments. In this release, a modified `portopt` only solves a portfolio problem for long-only fully invested portfolios. Use `Portfolio` instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
portopt	Error if ConSet or varargin input arguments are used	Portfolio	If you want to solve a portfolio problem that is more than a long-only fully invested portfolio, replace all instances of portopt with Portfolio.

For more information on migrating portopt code to Portfolio, see portopt Migration to Portfolio Object.



# R2015a

---

**Version: 5.5**

**New Features**

**Bug Fixes**

**Compatibility Considerations**

## Credit scorecard enhancements for model validation, a binning algorithm, and probability of default computation

- Enhancements to `autobinning` for the `Algorithm` name-value pair argument, where a new option `'Monotone'` is supported. `Monotone` is an optimal binning algorithm that ensures monotonicity in the weight of evidence (WOE) of the resulting bins.
- Credit scorecards support model validation using `validateModel` that provides the following three techniques:
  - Receiver Operating Characteristic (ROC)
  - Cumulative Accuracy Profile (CAP)
  - Kolmogorov-Smirnov (KS)
- Credit scorecards support probability of default using `probdefault`.

## autobinning support for 'Monotone' has compatibility impact

The `autobinning` function for credit scorecards has an incompatibility with the previous release. The latest version of `autobinning` supports, by default, new binning behavior where the default for the `'Algorithm'` argument is now a new name-value pair argument for `'Monotone'`. In addition, the algorithms `'EqualFrequency'` and `'EqualWidth'` now support `'SortCategories'` option for categorical data. By default, categorical data is sorted by `'odds'` before binning.

## Compatibility Considerations

To recover the previous behavior, use `autobinning` with the following name-value pairs:

- For the syntax `sc = autobinning(sc)` in R2014b, starting in R2015a, the syntax is equivalent to using:

```
sc = autobinning(sc, 'Algorithm', 'EqualFrequency', 'AlgorithmOptions', {'SortCategories', 'None'})
```
- For the syntax `sc = autobinning(sc, 'Algorithm', 'EqualWidth')` in R2014b, starting in R2015a, the syntax is equivalent to using:

```
sc = autobinning(sc, 'Algorithm', 'EqualWidth', 'AlgorithmOptions', {'SortCategories', 'None'})
```
- For the syntax `sc = autobinning(sc, 'Algorithm', 'EqualFrequency')` in R2014b, starting in R2015a, the syntax is equivalent to using:

```
sc = autobinning(sc, 'Algorithm', 'EqualFrequency', 'AlgorithmOptions', {'SortCategories', 'None'})
```

## Life table calibration and simulation for insurance

Life tables compute the probabilities, hazards, and survivor counts associated with people who are alive at a specified age and have the likelihood of death within a given period in the future. Four main parametric mortality models are supported for life studies: Gompertz, Gompertz-Makeham, Siler, and Heligman-Pollard.

- `lifetableconv` — Convert life table data from either raw form or generated form into different formats and series.
- `lifetablefit` — Calibrate parametric life table models based on life table data.

- `lifetablegen` — Generate life table data from parametric models.

## SDE suite parallel computing example

New example showing how to use Parallel Computing Toolbox™ with SDE functions to improve performance. For details, see [Improving Performance of Monte Carlo Simulation with Parallel Computing](#).

## frontcon removal

`frontcon` will be removed in a future release. Use `Portfolio` instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
<code>frontcon</code>	Warns	<code>Portfolio</code>	Replace all instances of <code>frontcon</code> with <code>Portfolio</code> .

To turn off the `frontcon` warning, see [Turning off the Warning Messages for frontcon](#).

For more information on migrating `frontcon` code to `Portfolio`, see [frontcon Migration to Portfolio Object](#).

## portopt partial removal

`portopt` will be partially removed in a future release and will no longer accept `ConSet` or `varargin` arguments. In a future release, `portopt` will solve the portfolio problem for long-only fully invested portfolios. Use `Portfolio` instead.

## Compatibility Considerations

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
<code>portopt</code>	Warns	<code>Portfolio</code>	If you want to solve a portfolio problem that is more than a long-only fully invested portfolio, replace all instances of <code>portopt</code> with <code>Portfolio</code> .

To turn off the `portopt` warning, see [Turning off the Warning Messages for portopt](#).

For more information on migrating `portopt` code to `Portfolio`, see [portopt Migration to Portfolio Object](#).



# R2014b

---

**Version: 5.4**

**New Features**

**Bug Fixes**

## Credit scorecard functionality

Modeling support for credit scorecard development that includes the following new functions:

- `creditscorecard` creates the `creditscorecard` object.
- `autobinning` applies automatic binning for single or multiple predictors.
- `bininfo` returns bin information for a given predictor.
- `modifybins` lets you modify bins for a given predictor.
- `bindata` bins a dataset using the existing binning rules and performs Weight of Evidence (WOE) transformation.
- `plotbins` plots histogram counts for predictor variables.
- `fitmodel` fits a logistic regression model using Weight of Evidence (WOE) data.
- `setmodel` sets the predictors and coefficients of a linear logistic regression model fitted outside the `creditscorecard` object and returns an updated `creditscorecard` object.
- `displaypoints` returns scorecard points information, such as points per bin or points per predictor.
- `formatpoints` lets you modify point information, such as scaling or rounding.
- `score` determines the score for each row of a dataset.

For more information, see [Using creditscorecard Objects](#), [Credit Scorecard Modeling Workflow](#), and [Case Study for a Credit Scorecard Analysis](#).

## Performance improvements to CVaR portfolio optimization when using the `fmincon` function

Support for `fmincon` gradients when using `setSolver` for CVaR portfolio optimization provides increased performance for CVaR optimizations.

## Performance improvements to SDE Monte Carlo simulation for models with constant parameter or deterministic function of time

Certain SDE models that use a constant parameter or a deterministic function of time have a performance improvement.

## Fan chart visualization function

Support for financial fan charts using `fanplot`. Use `fanplot` to plot the combination of historical and forecast data to visualize possible outcomes.

## SDE functions accept parameters that can be specified as a single input argument

The following SDE functions accept parameters you can specify as a single input argument that is identified as a deterministic function of time if the function accepts a scalar time `t` as its only input argument.

- 
- `bm`
  - `cev`
  - `cir`
  - `diffusion`
  - `drift`
  - `gbm`
  - `heston`
  - `hwv`
  - `sdeId`
  - `sdemrd`

In addition, `ts2func` accepts a new parameter value argument for `Deterministic` to support deterministic functions of time.

### **Default option for the cuttingplane solver for PortfolioCVaR optimization changed**

The default option for the cuttingplane solver for a `PortfolioCVaR` object has changed. The cuttingplane default option for `MasterSolverOptions` has changed from

```
optimoptions('linprog','Algorithm','Simplex','Display','off')
```

to

```
optimoptions('linprog','Algorithm','Dual-Simplex','Display','off')
```

For more information, see [Dual-simplex algorithm in linprog linear programming solver](#) in the Release Notes for Optimization Toolbox.





# R2014a

---

**Version: 5.3**

**New Features**

**Bug Fixes**

## **SDE functions moved to Financial Toolbox from Econometrics Toolbox**

The following Stochastic Differential Equation (SDE) functions have moved from Econometrics Toolbox to Financial Toolbox:

- `bm`
- `cev`
- `cir`
- `diffusion`
- `drift`
- `gbm`
- `heston`
- `hwv`
- `interpolate`
- `sde`
- `sdeddo`
- `sdemrd`
- `simByEuler`
- `simBySolution`
- `simulate`
- `ts2func`

The following sample data sets and examples from the `matlab/toolbox/econ/econdemos` directory have moved to `matlab/toolbox/finance/findemos`:

- `Demo_AmericanBasket`
- `Example_BarrierOption`
- `Example_BlackScholes`
- `Example_CEVModel`
- `Example_CIRModel`
- `Example_CopulaRNG`
- `Example_LongstaffSchwartz`
- `Example_StratifiedRNG`
- `Data_GlobalIdx2.mat`

## **Performance enhancements to SDE Monte Carlo simulation functions**

Monte Carlo simulation performance enhancements to the approximate solution function (`simBySolution`) of GBM and HWV models with constant parameters.

# R2013b

---

**Version: 5.2**

**New Features**

**Compatibility Considerations**

## Mean-absolute deviation (MAD) portfolio optimization

New portfolio object `PortfolioMAD` for mean-absolute deviation (MAD) portfolio optimization.

### optimoptions support

`optimoptions` support when using solver options for `Portfolio`, `PortfolioCVaR`, and `PortfolioMAD` objects for portfolio optimization.

### Compatibility Considerations

There are two possible incompatibility impacts:

- When using `Portfolio` or `PortfolioCVaR` objects and the associated `Portfolio.setSolver` or `PortfolioCVaR.setSolver` methods, the default solver options now reference an `optimoptions` object, instead of an `optimset` structure. If you now use default solver options and operating on them assuming this is an `optimset` structure, some of those operations may no longer work.
- The `Portfolio` or `PortfolioCVaR` objects and the associated `Portfolio.setSolver` or `PortfolioCVaR.setSolver` methods let you pass name-value pair arguments of solver options. In the past, setting solver options that were unused by the solver of choice would simply have no effect, because `optimset` would accept the options, and the solver would simply ignore them. In contrast, `optimoptions` objects generate an error if you attempt to set an invalid option.

`optimoptions` is the default and recommended method to set solver options, however, `optimset` is also supported.

### Functions moved from Financial Instruments Toolbox to Financial Toolbox

The following functions are moved from Financial Instruments Toolbox to Financial Toolbox:

- `cdai`
- `cdprice`
- `cdyield`
- `tbilldisc2yield`
- `tbillprice`
- `tbillrepo`
- `tbillval01`
- `tbillyield`
- `tbillyield2disc`

# R2013a

---

**Version: 5.1**

**New Features**

**Compatibility Considerations**

## Cash flow plot function

Graphical representation for cash flows using `cfplot`.

## Financial Time Series Tool (ftstool) import of Excel XLSX files on Linux and Mac OS X

Support for `ftstool` import of Excel® XLSX files on Linux® and Mac OS X.

## Cutting-plane solver added to PortfolioCVaR object

New solver option ('`cuttingplane`') for `PortfolioCVaR` object for conditional value-at-risk (CVaR) portfolio optimization. For more information, see `setSolver`.

## transprobytotals errors when using the algorithm input argument

The '`totals`' input argument to `transprobytotals` is typically generated by `transprob`. Because `transprob` includes an '`algorithm`' field in this structure since R2011b, you no longer need to specify the '`algorithm`' argument using a name-value pair when calling `transprobytotals`. If you specify an '`algorithm`' argument as a name-value pair when calling `transprobytotals`, you now receive an error.

## Compatibility Considerations

Specifying the '`algorithm`' as a name-value pair argument to `transprobytotals` now causes an error. If you started using this functionality in R2011b or later, most likely you don't have to take any action. If you have used this functionality before R2011b, make sure you remove the '`algorithm`' name-value pair from calls to `transprobytotals`, and that the '`totals`' input argument to `transprobytotals` contains an '`algorithm`' field indicating the desired algorithm. In most cases, the latter can be achieved by recreating the '`totals`' structure with a call to `transprob` which automatically adds the '`algorithm`' field since R2011b.

## Using `datenum`, `datestr`, `datevec` with dates in Financial products might produce inconsistent results

Any time you enter a cell array of date strings that are in different date formats using the MATLAB functions `datenum`, `datestr`, and `datevec`, these functions previously returned an error. In R2013a, this behavior has changed. In Financial products this change can cause an unexpected date format to generate an incorrect value. For example, the following use of `datevec` returned an error before R2013a because of the inconsistent date formats, but in R2013a this code does not return an error:

```
datevec({'10-Oct-2012', '10-1-2012'}),
```

## Compatibility Considerations

As a best practice, you should convert date strings to date numbers before using any functions in Financial Toolbox that use dates as inputs. For more information, see No strict-match requirements for month formats when converting date strings in the MATLAB release notes.

# R2012b

---

**Version: 5.0**

**New Features**

## **Conditional value at risk (CVaR) portfolio optimization**

New portfolio object `PortfolioCVaR` for conditional value at risk (CVaR) portfolio optimization.

## **Margin and spread calculations for floating-rate bonds**

Support for calculating spread measures for floating-rate bonds using `floatdiscmargin` and `floatmargin`.

## **Total (horizon) return calculation for fixed-coupon bonds**

Support for calculating bond horizon return using `bndtotalreturn`.

## **Performance improvements for cfamounts**

Performance improvement for calculating cash flows using `cfamounts`.



# R2012a

---

**Version: 4.2**

**New Features**

## xirr Update

Support is added to `xirr` for a global search heuristic to enhance the robustness of `xirr`.

## Additional Support for Cash Flow Functions

Function	Purpose
<code>cfspread</code>	Calculate the spread over a zero curve for a given cash flow.
<code>cfprice</code>	Calculate the price for a given cash flow given yield to maturity.
<code>cfyield</code>	Calculate the yield to maturity for a given cash flow and price.

## New Demo for Portfolio Optimization Tools

A new demo shows how to set up mean-variance optimization problems using the portfolio object. Run the demo at the MATLAB command line by entering:

```
showdemo portfolioexamples
```

# R2011b

---

**Version: 4.1**

**New Features**

**Compatibility Considerations**

## One-Way Turnover Constraints Added to the Portfolio Object

The portfolio object supports one-way turnover constraints using the new methods `setOneWayTurnover` and `getOneWayTurnover`.

## Portfolio Optimization with Sharpe Ratio Maximization Using a Portfolio Object

The portfolio object supports estimating an efficient portfolio that maximizes the Sharpe ratio using the new method `estimateMaxSharpeRatio`.

## Cash Flow and Time Mapping for Bond Portfolios with Variable Coupon Rates and Variable Face Values

Updated `cfamounts` now supports time-varying `CouponRate` and `Face` scheduling, including support for sinking fund bonds.

## Transition Probability Functions for Credit Quality Thresholds, Nonsquare Matrices, and User-Defined Ratings

Support is added for credit quality thresholds with `transprobtothresholds` and `transprobfromthresholds`. Support is added for data preprocessing for `transprob` using `transprobprep`. Support is added for user-defined ratings and nonsquare transition matrices with `transprobrouptotals` and `transprobbytotals`. For more information, see [Credit Risk Analysis](#).

## New Demo for Forecasting Corporate Default Rates

A new demo shows how to forecast corporate default rates. This includes backtesting and stress testing examples. Run the demo at the MATLAB command line by entering:

```
showdemo Demo_DefaultRatesForecasts
```

## Functionality Being Removed

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
<code>proddf</code>	Warns	<code>bndprice</code>	Replace all instances of <code>proddf</code> with <code>bndprice</code> .
<code>proddf1</code>	Warns	<code>bndprice</code>	Replace all instances of <code>proddf1</code> with <code>bndprice</code> .
<code>proddl</code>	Warns	<code>bndprice</code>	Replace all instances of <code>proddl</code> with <code>bndprice</code> .
<code>yldoddl</code>	Warns	<code>bndyield</code>	Replace all instances of <code>yldoddl</code> with <code>bndyield</code> .

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
yldoddf	Warns	bndyield	Replace all instances of yldoddf with bndyield.
yldoddf1	Warns	bndyield	Replace all instances of yldoddf1 with bndyield.
prbond	Warns	bndprice	Replace all instances of prbond with bndprice.
yldbond	Warns	bndyield	Replace all instances of yldbond with bndyield.
checksiz	Warns	N/A	Remove all instances from your code.
checktyp	Warns	N/A	Remove all instances from your code.
checkrng	Warns	N/A	Remove all instances from your code.

## Warning and Error ID Changes

Many warning and error IDs have changed from their previous versions. These warnings or errors typically appear during a function call.

## Compatibility Considerations

If you use warning or error IDs, you might need to change the strings you use. For example, if you turned off a warning for a certain ID, the warning might now appear under a different ID. If you use a try/catch statement in your code, replace the old identifier with the new identifier. There is no definitive list of the differences, or of the IDs that changed.

## transprobytotals Warns When Using the algorithm Input Argument

The totals input to transprobytotals is typically generated by transprob. Because transprob now includes an algorithm field in this structure, you no longer need to specify the algorithm argument when calling transprobytotals.

## Compatibility Considerations

In a future release, specifying the algorithm argument to transprobytotals will error. Currently, it is still permissible to specify the algorithm argument, although it usually has no effect.



# R2011a

---

**Version: 4.0**

**New Features**

## **Portfolio Turnover and Transaction Costs**

New portfolio object and methods support mean-variance portfolio optimization with general linear constraints, transaction costs, and turnover constraints. For more information, see Portfolio Optimization Tools and Portfolio Optimization Objects.

## **Updated showdemo Command for Credit Rating Demo**

The command to run the demo showing how to use Statistics Toolbox™ functions to support credit ratings is updated. Run the demo at the MATLAB command line by entering:

```
showdemo creditratingdemo
```



# R2010b

---

**Version: 3.8**

**New Features**

## **Estimation of Transition Probabilities for Credit Risk**

Support for estimation of transition matrices based on credit-migration history using both cohort and duration methods. For more information, see `transprob`, `transprobbytotals`, and Estimation of Transition Probabilities.

## **Improved Performance in Portfolio Optimization Functions**

`portopt` is enhanced for improved speed. Specifically, a broader class of problems now uses the faster linear complementarity programming (LCP) algorithm to obtain portfolios on the efficient frontier.

## **New Demo for Credit Rating**

A new demo shows how to use Statistics Toolbox functions to support credit ratings. Run the demo at the MATLAB command line by entering:

```
echodemo demo_creditrating
```

## **New Input and Output Options for Swap Functionality**

`cfamounts` is enhanced to support new parameter/value pairs for swap functionality.

# R2010a

---

**Version: 3.7.1**

**No New Features or Changes**



# R2009b

---

**Version: 3.7**

**New Features**

## Support for the BUS/252 Day-Count Convention

Support for the `Basis` day-count convention for BUS/252. BUS/252 is the number of business days between the previous coupon payment and the settlement date divided by 252. BUS/252 business days are non-weekend, non-holiday days. The `holidays.m` file defines holidays.

## Extended Support for New York Stock Exchange Closures

The current `holidays` function covers holidays and non-trading days from 1950 to 2050. Using `nyseclosures`, you can determine all known and anticipated closures from January 1, 1885 to December 31, 2050.

## Enhancements for Bond Pricing

Support for the following enhancements to bond pricing functions:

- Provide the ability to specify the compounding frequency separately from the coupon frequency.
- Enable specification of a discounting basis. A discounting basis has two purposes in Price/YTM calculations:
  - Computing the accrued interest
  - Computing the discount factors
- Support the specification of a formula for computing the interest in the last coupon period.

The enhanced bond pricing functions are:

Function	Purpose
<code>accfrac</code>	Calculate fraction of coupon period before settlement.
<code>bndprice</code>	Price fixed-income security from yield to maturity.
<code>bndyield</code>	Calculate yield to maturity for fixed-income security.
<code>bndspread</code>	Calculate static spread over spot curve.
<code>bnddurp</code>	Calculate bond duration given price.
<code>bnddury</code>	Calculate bond duration given yield to maturity.
<code>bndconvp</code>	Calculate bond convexity given price.
<code>bndconvy</code>	Calculate bond convexity given yield.
<code>cfamounts</code>	Calculate cash flow and time mapping for a bond portfolio.
<code>cftimes</code>	Calculate time factors corresponding to bond cash flow dates.

# R2009a

---

**Version: 3.6**

**New Features**

## **Support for Key Rate Duration**

Added support for `bndkrdur` to calculate key rate duration for bonds to determine the sensitivities of a bond to nonparallel changes in the yield curve. For more information, see [Calculating Key Rate Durations for Bonds](#).



# R2008b

---

**Version: 3.5**

**No New Features or Changes**



# R2008a

---

**Version: 3.4**

**New Features**

## Enhanced Mean-Variance Portfolio Optimization Based on Linear Complementarity Programming for Portfolio Optimization

Added support for `varargin` argument for `portopt` and `frontcon`.

### Support for Actual/365 (ISDA)

The following functions now support day count conventions for the `basis` argument based on ISDA (International Swap Dealers Association) `actual/365`:

- `accrfrac`
- `acrubond`
- `acrudisc`
- `bndconvp`
- `bndconvy`
- `bnddurp`
- `bnddury`
- `bndprice`
- `bndspread`
- `bndyield`
- `cfamounts`
- `cfdates`
- `cftimes`
- `cpncount`
- `cpndaten`
- `cpndatenq`
- `cpndatep`
- `cpndatepq`
- `cpndaysn`
- `cpnpersz`
- `datemnth`
- `daysadd`
- `daysdif`
- `disc2zero`
- `discrate`
- `fvdisc`
- `fwd2zero`
- `prbyzero`
- `prdisc`
- `prmat`
- `pyld2zero`

- 
- `time2date`
  - `yeardays`
  - `yearfrac`
  - `ylddisc`
  - `ylmdat`
  - `zbtprice`
  - `zbtyield`
  - `zero2disc`
  - `zero2fwd`
  - `zero2pyld`

### **Support for `ret2tick` and `tick2ret` Functions for Time Series Objects**

`ret2tick` and `tick2ret` support financial time series objects.

### **Support for Additional Descriptive Statistics Functions Financial Time Series Objects**

The following covariance methods now support a financial time series object:

- `corrcoef`
- `cov`
- `isempty`
- `nancov`
- `nanmax`
- `nanmedian`
- `nanmin`
- `nanstd`
- `nansum`
- `nanvar`
- `var`

### **Added New Chart Types**

Added support for the following chart types for financial reporting:

- `kagi`
- `renko`
- `linebreak`
- `priceandvol`
- `volarea`



# R2007b

---

**Version: 3.3**

**New Features**

## **ISMA Support for 30/360 Basis as a Variant of 30/360E with Annual Compounding**

The following functions now support day count conventions for the `basis` argument to support 30/360 International Securities Market Association (ISMA) convention as a variant of 30/360E with annual compounding:

- `accrfrac`
- `acrubond`
- `acrudisc`
- `bndconvp`
- `bndconvy`
- `bnddurp`
- `bnddury`
- `bndprice`
- `bndspread`
- `bndyield`
- `cfamounts`
- `cfdates`
- `cftimes`
- `cpncount`
- `cpndaten`
- `cpndatenq`
- `cpndatep`
- `cpndatepq`
- `cpndaysn`
- `cpnpersz`
- `datemnth`
- `daysadd`
- `daysdif`
- `disc2zero`
- `discrate`
- `fvdisc`
- `fwd2zero`
- `prbyzero`
- `prdisc`
- `prmat`
- `pyld2zero`
- `time2date`
- `yeardays`
- `yearfrac`



- 
- `ylddisc`
  - `yldmat`
  - `zbtprice`
  - `zbtyield`
  - `zero2disc`
  - `zero2fwd`
  - `zero2pyld`

### **createholidays Function Added for Different Trading Calendars**

The `createholidays` function now supports <https://www.FinancialCalendar.com> trading calendars. This function can be used from the command line or from the Trading Calendars graphical user interface. Using `createholidays`, you can create `holiday.m` files, in conjunction with `FinancialCalendar.com` data, that can be used instead of the standard `holidays.m` that ships with Financial Toolbox software.

### **Diagonal Covariance Matrix Support Added for Multivariate Normal Regression**

The new diagonal covariance matrix estimation feature makes it possible to estimate large-scale factor models by treating the residual errors as being jointly independent. The following functions support `CovarFormat`, a new input argument:

- `ecmlsrmle`
- `ecmmvnrml`
- `ecmmvnrfish`
- `ecmmvnrobj`
- `ecmmvnrstd`
- `mvnrfish`
- `mvnrml`
- `mvnrroj`
- `mvnrstd`

### **arith2geom and geom2arith Functions Added for Portfolio Analysis**

Two new functions, `arith2geom` and `geom2arith`, support portfolio analysis.



# R2007a

---

**Version: 3.2**

**New Features**

**Bug Fixes**

## ISMA Support Added

The following functions now support the International Securities Market Association (ISMA) convention for the `basis` argument:

- `accfrac`
- `acrubond`
- `acrudisc`
- `bndconvp`
- `bndconvy`
- `bnddurp`
- `bnddury`
- `bndprice`
- `bndspread`
- `bndyield`
- `cfamounts`
- `cfdates`
- `cftimes`
- `cpncount`
- `cpndaten`
- `cpndatenq`
- `cpndatep`
- `cpndatepq`
- `cpndaysn`
- `cpnpersz`
- `datemnth`
- `daysadd`
- `daysdif`
- `disc2zero`
- `discrate`
- `fvdisc`
- `fwd2zero`
- `prbyzero`
- `prdisc`
- `prmat`
- `pyld2zero`
- `time2date`
- `yeardays`
- `yearfrac`
- `ylldisc`

- 
- yldmat
  - zbtprice
  - zbtyield
  - zero2disc
  - zero2fwd
  - zero2pyld



# R2006b

---

**Version: 3.1**

**New Features**

## Investment Performance Metrics

The following new functions are added to compute common investment performance and risk-adjusted metrics:

- `sharpe`, computes the sharpe ratio.
- `inforatio`, computes information ratio and tracking error.
- `portalpha`, computes risk-adjusted alpha and return.
- `lpm`, computes sample lower partial moments.
- `elpm`, computes expected lower partial moments.
- `maxdrawdown`, computes the drop from maximum to minimum return over a period of time.
- `emaxdrawdown`, computes the returns that are transformed into a linear Brownian motion with drift.

## Financial Time Series Tool

Financial Time Series Tool (`ftstool`) is a new graphical user interface to support working with financial time series FINTS objects. `ftstool` interoperates with the Financial Time Series Graphical User Interface (`ftsgui`) and Interactive Charts (`chartfts`).



# R2006a

---

**Version: 3.0**

**New Features**

## Financial Time Series Toolbox Incorporated

As of this release the functionality previously available in Financial Time Series Toolbox has been incorporated into Financial Toolbox software. Financial Toolbox documentation has been modified to include the documentation previously available in the Financial Time Series User's Guide.

Because use of Financial Time Series Toolbox required the purchase and installation of Financial Toolbox software, all customers previously licensed for Financial Time Series Toolbox will continue to have access to it.

## Financial Time Series Frequency Conversion Functions Modified

The suite of time series frequency conversion functions (`todayly`, `toweekly`, `tomonthly`, `tosemi`, and `toannual`) has been extensively modified. Consult the function references in the Financial Toolbox User's Guide for specifics.

## Continuous Compounding Option Removed from `pyld2zero`

Continuous compounding is no longer available for `pyld2zero`. Compounding for this function is now consistent with compounding for the function `zero2pyld`. An error message is generated if you attempt to use continuous compounding with these functions.

## New Statistical Functions

The new functions in Version 3.0 of Financial Toolbox software fall into these four categories:

- “Multivariate Normal Regression Without Missing Data” on page 35-2
- “Multivariate Normal Regression With Missing Data (Expectation Conditional Maximization)” on page 35-2
- “Least Squares Regression With Missing Data (Expectation Conditional Maximization)” on page 35-3
- “Financial Model Transformation Function” on page 35-3

### Multivariate Normal Regression Without Missing Data

<code>mvnrfish</code>	Fisher information matrix for multivariate normal or least-squares regression
<code>mvnrml</code>	Multivariate normal regression (ignore missing data)
<code>mvnrobj</code>	Log-likelihood function for multivariate normal regression without missing data
<code>mvnrstd</code>	Evaluate standard errors for multivariate normal regression model

### Multivariate Normal Regression With Missing Data (Expectation Conditional Maximization)

<code>ecmmvnrfish</code>	Fisher information matrix for multivariate normal regression model
<code>ecmmvnrml</code>	Multivariate normal regression with missing data
<code>ecmmvnrobj</code>	Log-likelihood function for multivariate normal regression with missing data

---

ecmmvnrstd	Evaluate standard errors for multivariate normal regression model
------------	---

### **Least Squares Regression With Missing Data (Expectation Conditional Maximization)**

ecmlsrml	Least-squares regression with missing data
ecmlsrobj	Log-likelihood function for least-squares regression with missing data

### **Financial Model Transformation Function**

convert2sur	Convert a multivariate normal regression model into a seemingly unrelated regression model
-------------	--



# R14SP3

---

**Version: 2.5**

**New Features**

## New Statistical Functions

Version 2.5 introduces a set of financial statistical computation routines that compute values, such as mean and covariance, when there are missing data elements within a larger data set. These routines implement the Expectation Conditional Maximization (ECM) algorithm with various options that depend on the percentage of missing at random (MAR) data within the data set. The table below lists the functions that implement the ECM algorithm in Financial Toolbox software.

The following ECM functions have been added at this release.

### Expectation Conditional Maximization

<code>ecmfish</code>	Fisher information matrix
<code>ecmhess</code>	Hessian of negative log-likelihood function
<code>ecmnit</code>	Initial mean and covariance
<code>ecnmle</code>	Mean and covariance of incomplete multivariate normal data
<code>ecmobj</code>	Negative log-likelihood function
<code>ecmstd</code>	Standard errors for mean and covariance of incomplete data