java.util.ArrayList@Sat, 20 Jan 2018 09:19:59:715..81536329444851863O7
rows 138
cols 6

#	Package	Category	Function Name	Function Description	Function Parameters	Class path
1	jeconkr	AC.finance.cf	AC.finance.cf.bondcashflow		Number: ; Number: ; double: ; double: ; Number: ; double: ; Number: ; Number: ; Map: ; Map: ; Double:	jeconkr.finance.lib.server.functions.HW.CFFunctions
2	jeconkr	AC.finance.cmo	AC.finance.cmo.F	the function presents the CMO valuation algorithm as a function:	CalculatorCMO: CalculatorCMO calc - CMO calculator object; Map: Map<String, String> params - function parameters	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
3	jeconkr	AC.finance.cmo	AC.finance.cmo._get	retrun CMO or CMO calculator object field as array	Object: Object out - (i) CMO object or (ii) CMO calculator object; Map: ap<String, Object> keyMap - key parameters used to retrieve the field; int: int isTransposed - if 1, then transpose the returned array[][]	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
4	jeconkr	AC.finance.cmo	AC.finance.cmo.calc	create a CMO calculator object	CMO: CMO cmo - analyzed CMO object; List: List<Double> discounts - list of discount factors; Double: Double lnLossRateMean - mean of LN(loss rate); Double: Double lnLossRateStdev - stdev of LN(loss rate); Double: Double lnAmortRateMea - mean of LN(amortiz	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
5	jeconkr	AC.finance.cmo	AC.finance.cmo.get	get the field from CMO or CMO calculator object using a given key	Object: Object out - (i) CMO object or (ii) CMO calculator object; Map: Map<String, Object> keyMap - key parameters used to retrieve the field	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
6	jeconkr	AC.finance.cmo	AC.finance.cmo.model	create a cmo object	String: String name - name of the CMO object; Object: Object cpnRate - coupon rate (i) a number if fixed or (ii) array[] if floating; Number: Number periodCount - number of periods (until maturity); Object[][]:	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
7	jeconkr	AC.finance.irb	AC.finance.irb._get		Object: ; String: ; int:	jeconkr.finance.lib.server.functions.FSTP.irb.IRBFunctions
8	jeconkr	AC.finance.irb	AC.finance.irb.calcTS		String: ; List: ; List: ; Map:	jeconkr.finance.lib.server.functions.FSTP.irb.IRBFunctions
9	jeconkr	AC.finance.irb	AC.finance.irb.calcTSN		String: ; Double[][]: ; Double[][]: ; Map:	jeconkr.finance.lib.server.functions.FSTP.irb.IRBFunctions
10	jeconkr	AC.finance.irb	AC.finance.irb.get		Object: ; String:	jeconkr.finance.lib.server.functions.FSTP.irb.IRBFunctions
11	jeconkr	AC.finance.srm	AC.finance.srm.calc		String: ; IShortRateModel: ; Map: ; Map: ; List:	jeconkr.finance.lib.server.functions.HW.SRMFunctions
12	jeconkr	AC.finance.srm	AC.finance.srm.calibrate		String: ; Map: ; Double:	jeconkr.finance.lib.server.functions.HW.SRMFunctions
13	jeconkr	AC.finance.srm	AC.finance.srm.get		ICalculatorSRM: ; String:	jeconkr.finance.lib.server.functions.HW.SRMFunctions
14	jeconkr	AC.finance.srm	AC.finance.srm.tree		String: ; Double: ; Double: ; Double: ; Object: ; Map: ; IFunctionX: ; IFunctionX: ; Object: ; Map:	jeconkr.finance.lib.server.functions.HW.SRMFunctions
15	jeconkr	AC.finance: CMO	AC.finance.cmo.validate	validate the estimated CMO object (CMO structure and CMO cash flows)	CalculatorCMO: CalculatorCMO calc - CMO calculator object	jeconkr.finance.lib.server.functions.FSTP.cmo.CMOFunctions
16	jedt	AC.edit.docx	AC.edit.docx._get		Object: ; String: ; Number:	jedt.lib.server.functions.docx4j.DocxFunctions
17	jedt	AC.edit.docx	AC.edit.docx.fields	retrieve the list of content control field values from the document	WordprocessingMLPackage: pkg - java object of MS Word document; String[]: keys - ids of the loaded content control fields; Number: loadCount - execute the method if loadCount value changes	jedt.lib.server.functions.docx4j.DocxFunctions
18	jedt	AC.edit.docx	AC.edit.docx.get	return the document data using a given key	Object: obj - either (i) WordprocessingMLPackage or (ii) List<String>; String: key - key to obtain and display the output either from document or search results	jedt.lib.server.functions.docx4j.DocxFunctions
19	jedt	AC.edit.docx	AC.edit.docx.highlight	dd / remove highlight to the content control fields	WordprocessingMLPackage: pkg - java object of MS Word document; Number: highlightMode - if 0 then highlight is removed; otherwise highlight is added; String: keyword - id of the controlcontrol fields, which are highligted; Number: highlightControl - if	jedt.lib.server.functions.docx4j.DocxFunctions
20	jedt	AC.edit.docx	AC.edit.docx.load	load a MS Word document	String: folderPath - full path to the document folder; String: fileName - file name of the loaded document; Number: loadCount - execute the method if loadCount value changes	jedt.lib.server.functions.docx4j.DocxFunctions
21	jedt	AC.edit.docx	AC.edit.docx.remove	remove all blank control control fields and return the number of removed fields	WordprocessingMLPackage: pkg - java object of MS Word document; String: key - remove only the fields with key=sdt-blank value; Number: removeControl - if 1, then execute the method	jedt.lib.server.functions.docx4j.DocxFunctions
22	jedt	AC.edit.docx	AC.edit.docx.repair	repair all document blocks (custom fields enclosed with {} symbols) that are converted into the sdt elements	WordprocessingMLPackage: pkg - java object of MS Word document; String[][]: keys - keys of repaired custom fields; Number: repairControl - if 1, then execute the method	jedt.lib.server.functions.docx4j.DocxFunctions
23	jedt	AC.edit.docx	AC.edit.docx.save	save the loaded document as a docx file	WordprocessingMLPackage: pkg - java object of MS Word document; String: folderPath - full path to the document folder; String: fileName - file name of the saved document; Number: saveControl - if 1, then execute the method	jedt.lib.server.functions.docx4j.DocxFunctions

24	jedt	AC.edit.docx	AC.edit.docx.search	search all document elements using given xpath and keyword (or regular expression)	WordprocessingMLPackage: xpath - xpath used to search for document elements; String: keyword - keyword used to search for document elements; String: regexp - regular expression used to search for document elements; String: srchCount - execute the method i	jedt.lib.server.functions.docx4j.DocxFunctions
25	jedt	AC.edit.docx	AC.edit.docx.table	insert data array in the document tables	WordprocessingMLPackage: pkg - java object of MS Word document; Object[][]: data - inserted data; String: keyword - table id to identify the table in the document; Number: rowIndex - first row in the table to insert data; Number: colIndex - first column i	jedt.lib.server.functions.docx4j.DocxFunctions
26	jedt	AC.edit.docx	AC.edit.docx.tag	wrap all custom fields (enclosed with {} symbols) into the sdt elements	WordprocessingMLPackage: pkg - java object of MS Word document; Number: tagControl - if 1, then execute the method	jedt.lib.server.functions.docx4j.DocxFunctions
27	jedt	AC.edit.docx	AC.edit.docx.update	update the values of content control fields for a given list of key	WordprocessingMLPackage: pkg - java object of MS Word document; String[]: keys - ids of the updated content control fields; String[]: values - values of the updated content control fields; Number: updateControl - if 1, then execute the method	jedt.lib.server.functions.docx4j.DocxFunctions
28	jedt	AC.edit.svg	AC.edit.svg.get	retrieve a specific field of the svg document using related key	Map: svgOutput - mapping file name => parse output object; String: fileName - file name, which repective parsed object is searched for the field; String: key - field key	jedt.lib.server.functions.xml.SvgFunctions
29	jedt	AC.edit.svg	AC.edit.svg.load	load and parse an svg file to retrieve the data	String: folderPath - full path to the folder with svg files; Object[][]: params - array of the form param[i] = {file name, minimum value on y-axis,maximum value on y axis}; Number: recalculate - if 1, then the file is reloaded and re-parsed even if it was	jedt.lib.server.functions.xml.SvgFunctions
30	jedt	AC.edit.xml	AC.edit.xml.data	retrieve data for all nodes with a given name and childs from a given name list	INode: node - starting node to traverse the document; String: nodeName - name of the searched node; List: childNames - list of searched child names	jedt.lib.server.functions.xml.XmlFunctions
31	jedt	AC.edit.xml	AC.edit.xml.load	Load an xml file and return the root of the xml document	String: xmlFilePath - full path to the xml document	jedt.lib.server.functions.xml.XmlFunctions
32	jedt	AC.edit.xml	AC.edit.xml.mapping	mapping node A => {child nodes B} returned as an array (node A, data, child nodes B, child node data)	INode: root - starting node to traverse the document; String: nodeName; String: nodeIdName; List: descNames; String: nodeRefName	jedt.lib.server.functions.xml.XmlFunctions
33	jedt	AC.edit.xml	AC.edit.xml.structure	return the structure of the xml document as a list of all distinct node name sequences	INode: node - the node, starting from which all child node sequences are analyzed and added to the list	jedt.lib.server.functions.xml.XmlFunctions
34	jkr	AC.data	AC.data._get	return the data object field by key. If data is a list it is converted to array[], if data is a map it is converted to [keys, values] array or is parsed to array[][] if map depth exceeds 1 (is of the form key-1=>{key-2=>...})	Object: data - data object; String: String key - field key (typically blank); Number: transpose - if 1 and data is a two-dimensional list, then the list is converted to two-dimensional array and is transposed	jkr.parser.lib.server.functions.data.DataFunctions
35	jkr	AC.data	AC.data.copy	copy a list of data	List: x - input list of data	jkr.parser.lib.server.functions.data.DataFunctions
36	jkr	AC.data	AC.data.element	get element of data object (by key for a map and by index for a list)	Object: obj - data object (a map or a list); Object: key - a key object (if data is a map) or int if data is a list)	jkr.parser.lib.server.functions.data.DataFunctions
37	jkr	AC.data	AC.data.filter	filter the rows of two-dimensional array of data by keyword in column with agiven index	List: data - two-dimensional array of data; Number: index - column index; String: keyword - keyword to filter data	jkr.parser.lib.server.functions.data.DataFunctions
38	jkr	AC.data	AC.data.list_lin_x	create a list of n points distributed uniformly on the interval [a, b]	Number: a - minimum value of the range; Number: b - maximum value of the range; Number: n - number of points in the list	jkr.parser.lib.server.functions.data.DataFunctions
39	jkr	AC.data	AC.data.list_x	create a list of objects	Object[]: Object[] data - array of objects; Number: isNumeric - if 1, then blank or non-numeric elements are replaced with Double.NaN values; Number: skipBlank - if 1, then all blank cells data is excluded from the list	jkr.parser.lib.server.functions.data.DataFunctions
40	jkr	AC.data	AC.data.list_xx	create two-dimensional list from two-dimensional array of objecs	Object[][]: data - array of objects; Number: transposed - if 1, then the created two-dimensional list is transposed	jkr.parser.lib.server.functions.data.DataFunctions
41	jkr	AC.data	AC.data.listmap	create a list of map objects: {keys=>values[i]}	Object[]: keys - array of key objects; Object[][]: values - two dimensional array of value objects	jkr.parser.lib.server.functions.data.DataFunctions
42	jkr	AC.data	AC.data.map	Create a map keys=>value object. Keys can model multi-level map keys. In this acse the method parses the keys and creates a multi-level output map object.	Object[]: keys - list of keys; Object[]: values - list of values	jkr.parser.lib.server.functions.data.DataFunctions
43	jkr	AC.data	AC.data.merge	merge two two-dimensional arrays (by rows)	List: X1 - first two-dimensional arrray; List: X1 - second two-dimensional arrray	jkr.parser.lib.server.functions.data.DataFunctions
44	jkr	AC.data	AC.data.size	return data object size, where object is a map or a list	Object: obj - data object; Number: dim - used for lists and equals 1 for number of list rows and 2 for number of list columns	jkr.parser.lib.server.functions.data.DataFunctions
45	jkr	AC.data	AC.data.sort	copy and sort a list data	List: x - input list of data	jkr.parser.lib.server.functions.data.DataFunctions

46	jkr	AC.data.io	AC.data.io.export	export data	Object[][]: data - exported data array; String: type - export type (currently supports only 'csv'; String: folderPath - path to csv file folder; String: fileName - csv file name; Number: transpose - if 1, then exported data array is transposed	jkr.parser.lib.server.functions.io.IOFunctions
47	jkr	AC.data.io	AC.data.io.exportcsv	export array[][] data to a csv file	Object[][]: data - exported data; String: folderPath - csv file folder path; String: fileName - csv file name; Number: transpose - if 1, then exported data array is transposed	jkr.parser.lib.server.functions.io.IOFunctions
48	jkr	AC.data.io	AC.data.io.importcsv	import csv file	String: folderPath - csv file folder path; String: fileName - csv file name; Number: transpose - if 1, then imported data array is transposed	jkr.parser.lib.server.functions.io.IOFunctions
49	jkr	AC.jmc	AC.jmc.getVars	return an array of variables from the code	ICalculator: calculator - jmc calculator that executes the parsed code; String[]: names - array of variable names, which values are retrieved from the calculator	jkr.parser.lib.server.functions.jmc.JmcFunctions
50	jkr	AC.jmc	AC.jmc.parse	parse jmc code	String: id - code id; String: code - code text that is parsed	jkr.parser.lib.server.functions.jmc.JmcFunctions
51	jkr	AC.math.F	AC.math.F.eval	performs function evaluation y = F(x), where F:Rn->R for a collection of function arguments	IFunctionX: F - evaluated function; Object: args - function arguments (i) a set of arguments or (ii) a list of function arguments, where each argument is a List<Double> object	jkr.parser.lib.server.functions.jmc.general.FFunctions
52	jkr	AC.math.F	AC.math.F.eval1	function values y = F(x), where F:Rn->R; after constructing y sequence, F(x0) is evaluated	IFunctionX: F - evaluated functio; List: args - list of x arguments; Object: x0 - last argument at which F is evaluated	jkr.parser.lib.server.functions.jmc.general.FFunctions
53	jkr	AC.server	AC.server.functions	return a list of server function names and functions information		jkr.parser.lib.server.functions.ServerFunctions
54	jkr	AC.test	AC.test.connection	test connection with the server and return session information (host, user, key)		jkr.parser.lib.server.functions.test.TestFunctions
55	jkr	AC.test	AC.test.helloworld	test of a simple 'Hello World!' server function		jkr.parser.lib.server.functions.test.TestFunctions
56	jkr	AC.test	AC.test.type	return the type of the created object	Object: obj - object which type is being tested	jkr.parser.lib.server.functions.test.TestFunctions
57	jmathkr	AC.math.F	AC.math.F.composite	composite function Rn->R: h(x)=g o f(x)	IFunctionX: f - mapping X -> Y; IFunctionX: g - mapping Y -> V	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
58	jmathkr	AC.math.F	AC.math.F.constant	function Rn->R: f(x)=c	Number: c - constant; Number: x - dimension of function argument	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
59	jmathkr	AC.math.F	AC.math.F.linear	function Rn->R: f(x)=a*x+c	Object: a - (i) number or (ii) list that models nx1 vector; Number: c	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
60	jmathkr	AC.math.F	AC.math.F.list	function Rn->Rm: f(x)=(f1(x),...,fm(x))	List: fns - list of functions (f1(x),...,fm(x))	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
61	jmathkr	AC.math.F	AC.math.F.log	function Rn->R: f(x)=a*log(x)	Object: a - (i) number or (ii) list that models nx1 vector	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
62	jmathkr	AC.math.F	AC.math.F.polynom		Number: ; Number:	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
63	jmathkr	AC.math.F	AC.math.F.pow	function Rn->R: f(x)=x^a	Object: a - (i) number or (ii) list that models nx1 vector	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
64	jmathkr	AC.math.F	AC.math.F.quad	function Rn->R: f(x)=xQx+ax+c	List: Q - two-dimesional list that models nxn matrix; List: a - List that models nx1 vector; Number: Number c	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
65	jmathkr	AC.math.F	AC.math.F.solve	root solver for F:R->R; F(x)=f0, where x in [a, b]	IFunctionX: F - mapping R -> R; Number: f0; Number: a - range lower bound; Number: b - range upper bound; Number: eps; Number: n - max number of iterations	jmathkr.lib.server.xlloop.functions.math.calculus.function.FFunctions
66	jmathkr	AC.math.M	AC.math.M.XXT		List: ; double: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
67	jmathkr	AC.math.M	AC.math.M.det		List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
68	jmathkr	AC.math.M	AC.math.M.eig		List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
69	jmathkr	AC.math.M	AC.math.M.inv		List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
70	jmathkr	AC.math.M	AC.math.M.linear		List: ; List: ; List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
71	jmathkr	AC.math.M	AC.math.M.linseq		List: ; List: ; List: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
72	jmathkr	AC.math.M	AC.math.M.linsum	calculate linear sum z = ax+by	Double: a; List: x; Double: b; List: y	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
73	jmathkr	AC.math.M	AC.math.M.pow		List: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
74	jmathkr	AC.math.M	AC.math.M.project		List: ; List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions

75	jmathkr	AC.math.M	AC.math.M.rank		List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
76	jmathkr	AC.math.M	AC.math.M.reverse		List: ; int: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
77	jmathkr	AC.math.M	AC.math.M.series		List: ; Double[]: ; Integer[]:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
78	jmathkr	AC.math.M	AC.math.M.sort		List: ; int: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
79	jmathkr	AC.math.M	AC.math.M.sublist		List: ; int: ; int: ; int: ; int:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
80	jmathkr	AC.math.M	AC.math.M.times		Double: ; List: ; List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
81	jmathkr	AC.math.M	AC.math.M.transpose		List:	jmathkr.lib.server.xlloop.functions.math.algebra.MFunctions
82	jmathkr	AC.math.op	AC.math.op.divide		Object: ; Object:	jmathkr.lib.server.xlloop.functions.math.calculus.function.OpFunctions
83	jmathkr	AC.math.op	AC.math.op.minus		Object: ; Object:	jmathkr.lib.server.xlloop.functions.math.calculus.function.OpFunctions
84	jmathkr	AC.math.op	AC.math.op.plus		Object: ; Object:	jmathkr.lib.server.xlloop.functions.math.calculus.function.OpFunctions
85	jmathkr	AC.math.op	AC.math.op.times		Object: ; Object:	jmathkr.lib.server.xlloop.functions.math.calculus.function.OpFunctions
86	jmathkr	AC.math.optim	AC.math.optim.constrained	funcion constrained optimization bounded by rectangular [a, b]	IFunctionX: F - optimized function; List: G - list of constraint functions; List: x0 - initial point; List: a - left bound; List: b - right bound; Map: params - optimization parameters	jmathkr.lib.server.xlloop.functions.math.optim.MaxFFunctions
87	jmathkr	AC.math.optim	AC.math.optim.get	get results of function optimization solution	ISolver: solver - optimization solver (ISolver object); String: key - the reference key for the field	jmathkr.lib.server.xlloop.functions.math.optim.MaxFFunctions
88	jmathkr	AC.math.optim	AC.math.optim.line	function optimization over a straight line [a, b]	IFunctionX: F - optimized function; List: a - left bound; List: b - right bound; Map: params - optimization parameters	jmathkr.lib.server.xlloop.functions.math.optim.MaxFFunctions
89	jmathkr	AC.math.optim	AC.math.optim.unconstrained	funcion unconstrained optimization bounded by rectangular [a, b]	IFunctionX: F - optimized function; List: x0 - initial point; List: a - left bound; List: b - right bound; Map: params - optimization parameters	jmathkr.lib.server.xlloop.functions.math.optim.MaxFFunctions
90	jmathkr	AC.math.set	AC.math.set.build	build a set object which models a subset of Rn set	Map: params - parameters of the set with keys (x-min, x-max, point-count, set-constraint)	jmathkr.lib.server.xlloop.functions.math.calculus.set.SetFunctions
91	jmathkr	AC.math.set	AC.math.set.get	get set object fields	SetWrapper: set - the set object; String: key - the key to retrieve set field; keys={list-Rn}	jmathkr.lib.server.xlloop.functions.math.calculus.set.SetFunctions
92	jmathkr	AC.stats.R	AC.stats.R.eval	evaluate R code	String: code - evaluated code	jmathkr.lib.server.xlloop.functions.stats.R.RFunctions
93	jmathkr	AC.stats.R	AC.stats.R.get	return the output of R code execution	ROutput: out - ROutput object; String: key - refernce key for the output field	jmathkr.lib.server.xlloop.functions.stats.R.RFunctions
94	jmathkr	AC.stats.diffusion	AC.stats.diffusion.getTree		ITreeDiffusionR1: ; String: ; Map:	jmathkr.lib.server.xlloop.functions.stats.diffusion.DiffusionFunctions
95	jmathkr	AC.stats.diffusion	AC.stats.diffusion.tree		Double: ; double: ; double: ; Object: ; IFunctionX: ; IFunctionX: ; Map:	jmathkr.lib.server.xlloop.functions.stats.diffusion.DiffusionFunctions
96	jmathkr	AC.stats.diffusionctrl	AC.stats.diffusionctrl.getTree		ICalculatorDiffusionCtrlR1: ; String:	jmathkr.lib.server.xlloop.functions.stats.diffusion.DiffusionCtrlFunctions
97	jmathkr	AC.stats.diffusionctrl	AC.stats.diffusionctrl.tree		double: ; double: ; double: ; Number: ; Number: ; Object: ; List: ; Map: ; List: ; IFunctionX: ; IFunctionX: ; Map: ; Object: ; Map:	jmathkr.lib.server.xlloop.functions.stats.diffusion.DiffusionCtrlFunctions
98	jmathkr	AC.stats.distrib	AC.stats.distrib.cdf	alculated cumulative probability distribution for a given distribution object and sample	IDistributionR1: D - distribution object; List: x - list of points x = {xi} for which cdf is estimated	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
99	jmathkr	AC.stats.distrib	AC.stats.distrib.fn	create a distribution object	Object: o1 - (i) list of distribution states or (ii) distribution type (String); Object: o2 - (i) list of distribution values or (ii) sample used to estimate distribution parameters; or (iii) map key=>value with distribution parameters	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
100	jmathkr	AC.stats.distrib	AC.stats.distrib.get	get the object field using a given key. If object is a list, the method is applied to each element of the list	Object: obj - (i) List of objects; or (ii) ITestDistribution object; or (iii) IDistributionR1 object; String: key - key used to retrieve the field	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
101	jmathkr	AC.stats.distrib	AC.stats.distrib.pdf	calculated probability distribution for a given distribution object and sample	IDistributionR1: D - distribution object; List: x - list of points x = {xi} for which pdf is estimated; Number: normalized - if 1 then each pdf(xi) value is multiplied by dxi	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
102	jmathkr	AC.stats.distrib	AC.stats.distrib.qnt	alculated quantiles for a given distribution object and probability list	IDistributionR1: D - distribution object; List: p - list of probabilities p = {pi} for which qnt is estimated	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
103	jmathkr	AC.stats.distrib	AC.stats.distrib.sample	generate a sample of size n from a given distribution	IDistributionR1: D - distribution object; Number: n - sample size	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions

104	jmathkr	AC.stats.distrib	AC.stats.distrib.test	test the distribution type	String: testType - type of the test; Object: distribType - (i) Sring that represents distribution; (ii) distribution object; or (iii) list<String> of distribution types; Object: sample - tested sample	jmathkr.lib.server.xlloop.functions.stats.basic.DistributionFunctions
105	jmathkr	AC.stats.markov	AC.stats.markov.calc	create Markov tree calculator	ITreeMarkovR1: M - Markov tree; Map: params - calculator parameters; List: methods - list of executed methods	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
106	jmathkr	AC.stats.markov	AC.stats.markov.convertOutpu		Object: ; List:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
107	jmathkr	AC.stats.markov	AC.stats.markov.getCalc		ICalculatorMarkovR1: ; String:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
108	jmathkr	AC.stats.markov	AC.stats.markov.getParameter	get Markov calculator parameters	Map: ; String:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
109	jmathkr	AC.stats.markov	AC.stats.markov.getTree		ITreeMarkovR1: ; String: ; Map:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
110	jmathkr	AC.stats.markov	AC.stats.markov.tree	create Markov tree	int: T - number of periods; double: x0 - initial state; Object: states - list/map of states; Object: Qx - transition probabilities	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovFunctions
111	jmathkr	AC.stats.markovctrl	AC.stats.markovctrl.calc		ITreeMarkovCtrlR1: ; Map: ; List:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovCtrlFunctions
112	jmathkr	AC.stats.markovctrl	AC.stats.markovctrl.getCalc		ICalculatorMarkovCtrlR1: ; String: ; Map:	jmathkr.lib.server.xlloop.functions.stats.markov.MarkovCtrlFunctions
113	jmathkr	AC.stats.reg	AC.stats.reg._get		OutputLReg: ; String: ; int:	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
114	jmathkr	AC.stats.reg	AC.stats.reg.get	return the field from the regression output object referenced by a given key	OutputLReg: out - regression output object; String: key - reference key used to retrieve the output object field	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
115	jmathkr	AC.stats.reg	AC.stats.reg.logit	run logit regression y = b X + e	List: y - list that models endogenous nx1 vector; List: X - two-dimensional list that models nxk exogenous variables; Map: params - logit parameters	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
116	jmathkr	AC.stats.reg	AC.stats.reg.ols	run ols regression y = b X + e	List: y - list that models endogenous nx1 vector; List: X - two-dimensional list that models nxk exogenous variables; Map: params - ols parameters	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
117	jmathkr	AC.stats.reg	AC.stats.reg.var	run vector autoregression analysis (var)	List: Y - two-dimensional list that models sample described by an nxm array; Number: p - number of var lags; Number: T - number of future forecasted periods; Number: T0 - number of backtested historical periods; Map: params - var parameters	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
118	jmathkr	AC.stats.reg	AC.stats.reg.varx	run vector autoregression analysis with exogenous variables (varx)	List: Y - two-dimensional list that models sample described by an nxm array; List: X - two-dimensional list that models sample described by an nxk array; List: muX - steady state of the exogenous variables; Number: p - number of varx lags; Map: params - v	jmathkr.lib.server.xlloop.functions.stats.regression.RegressionFunctions
119	jmathkr	AC.stats.rng	AC.stats.rng.normal	generate an nxm two-dimensional list of random numbers	Number: n - number of rows; Number: m - number of columns	jmathkr.lib.server.xlloop.functions.stats.basic.RngFunctions
120	jmathkr	AC.stats.sample	AC.stats.sample.clean	remove non-numeric values from the sample	Object: samples - (i) a list of sample objects or (ii) a map key => list<ISample> of sample objects	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
121	jmathkr	AC.stats.sample	AC.stats.sample.get	convert to list a sample, list of samples, or groupped samples object	Object: samples - (i) ISample object; or (ii) List<ISample> object; or (iii) Map<Object, List<ISample>> object; Number: valuesOnly - if 1, then only values are included in the output list. Otherwise the observation keys are also included	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
122	jmathkr	AC.stats.sample	AC.stats.sample.group	group data by sample key; create a map: sample key => list of samples that correspond to data array columns	Object[]: sampleKeys - array of sample keys (size must match to number of rows in sample data); Object[]: obsKeys - observation keys (typically represented by data row indices); Object[][]: values - array of sample data (each data column represents a samp	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
123	jmathkr	AC.stats.sample	AC.stats.sample.keys	return list of keys after performing group(...) function	Map: samples - samples groupped into the sample key => List<ISample> map object	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
124	jmathkr	AC.stats.sample	AC.stats.sample.mean	return sample mean	List: sample - sample of values	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
125	jmathkr	AC.stats.sample	AC.stats.sample.mode	return sample mode	List: sample - sample of values	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
126	jmathkr	AC.stats.sample	AC.stats.sample.moment	calculate a moment statistic for a sample, list of samples, or groupped samples object	Object: samples - (i) ISample object; or (ii) List<ISample> object; or (iii) Map<Object, List<ISample>> object; String: type - moment type; Double: param - parameter required for calculation of some moments statistics	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
127	jmathkr	AC.stats.sample	AC.stats.sample.sample	create a sample object from an array[] of values	Object[]: values - array of values	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
128	jmathkr	AC.stats.sample	AC.stats.sample.size	return the size of the sample or list of sample objects	Object: samples - (i) ISample object; or (ii) List<ISample> object; or (iii) map: key => List<ISample>	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
129	jmathkr	AC.stats.sample	AC.stats.sample.sort	sort a sample, list of samples, or groupped samples object	Object: samples - (i) ISample object; or (ii) List<ISample> object; or (iii) Map<Object, List<ISample>> object	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions

130	jmathkr	AC.stats.sample	AC.stats.sample.stdev	return sample standard deviation	List: sample - sample of values	jmathkr.lib.server.xlloop.functions.stats.basic.SampleFunctions
131	jmathkr	AC.stats.samplecalc	AC.stats.samplecalc.get	return the sample calculator field using a given key	Object: obj - sample calculator object; String: key - reference key used to retrieve the field	jmathkr.lib.server.xlloop.functions.stats.basic.SampleCalcFunctions
132	jmathkr	AC.stats.samplecalc	AC.stats.samplecalc.hist	estimate sample histogram	List: x - list that models sample described by nx1 array; Map: params - parameters applied to estimate the histogram	jmathkr.lib.server.xlloop.functions.stats.basic.SampleCalcFunctions
133	jmathkr	AC.stats.samplecalc	AC.stats.samplecalc.pca	return the principal component calculator	List: x - two-dimensional list that models nxn matrix	jmathkr.lib.server.xlloop.functions.stats.basic.SampleCalcFunctions
134	jmathkr	AC.stats.simulation	AC.stats.simulation.get		ISimulationModel: ; String:	jmathkr.lib.server.xlloop.functions.stats.simulation.SimulationFunctions
135	jmathkr	AC.stats.simulation	AC.stats.simulation.run		String: ; Object: ; Map: ; int:	jmathkr.lib.server.xlloop.functions.stats.simulation.SimulationFunctions
136	jmathkr	AC.test	AC.test.get	get tested object fields	Object: obj - output object which fields are retrieved; String: key - reference key to retrieve the output object field	jmathkr.lib.server.xlloop.functions.test.TestFunctions
137	jmathkr	AC.test	AC.test.set	build a set object which models a subset of Rn set	Map: params - parameters of the set with keys (x-min, x-max, point-count, set-constraint)	jmathkr.lib.server.xlloop.functions.test.TestFunctions
138			AC.test.equals		aObject; bObject	
139						
140						
141						
142						
143						
144						
145						
146						
147						
148						
149						
150						