Package ‘crp.CSFP’

September 11, 2016

Type Package
Title CreditRisk+ Portfolio Model
Version 2.0.2
Date 2016-09-09
Author Kevin Jakob, Dr. Matthias Fischer & Stefan Kolb
Maintainer Kevin Jakob <kevin.Jakob.Research@gmail.com>
License GPL-2
LazyLoad yes
Imports methods, MASS, utils, graphics
NeedsCompilation no
Repository CRAN
Date/Publication 2016-09-11 18:35:59

R topics documented:

  crp.CSFP-package .......................................................... 4
  a ................................................................. 5
  alpha ............................................................... 5
  alpha.crp ........................................................... 5
  alpha.max .......................................................... 5
  alpha.max<- ....................................................... 5
  alpha<- .......................................................... 6
  B ................................................................. 6
  calc.portfolio.statistics ............................................. 6
  calc.rc ............................................................ 6
  calc.rc<- ......................................................... 7
  CDF ............................................................... 7
  changes.calc.portfolio.statistics .................................. 7
  changes.export ..................................................... 7
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>changes.loss</td>
<td>7</td>
</tr>
<tr>
<td>changes.measure</td>
<td>8</td>
</tr>
<tr>
<td>changes.plausi</td>
<td>8</td>
</tr>
<tr>
<td>changes.plot</td>
<td>8</td>
</tr>
<tr>
<td>changes.rc.sd</td>
<td>8</td>
</tr>
<tr>
<td>changes.rc.vares</td>
<td>8</td>
</tr>
<tr>
<td>changes.read</td>
<td>9</td>
</tr>
<tr>
<td>CP.NR</td>
<td>9</td>
</tr>
<tr>
<td>CP.NR&lt;-</td>
<td>9</td>
</tr>
<tr>
<td>CP.rating</td>
<td>9</td>
</tr>
<tr>
<td>CP.rating&lt;-</td>
<td>9</td>
</tr>
<tr>
<td>crp.CSFP</td>
<td>10</td>
</tr>
<tr>
<td>crp.CSFP-class</td>
<td>11</td>
</tr>
<tr>
<td>crp.round</td>
<td>16</td>
</tr>
<tr>
<td>EC</td>
<td>16</td>
</tr>
<tr>
<td>EL</td>
<td>16</td>
</tr>
<tr>
<td>EL.crp</td>
<td>16</td>
</tr>
<tr>
<td>ES</td>
<td>17</td>
</tr>
<tr>
<td>ES.cont</td>
<td>17</td>
</tr>
<tr>
<td>ES.tau.cont</td>
<td>17</td>
</tr>
<tr>
<td>export</td>
<td>17</td>
</tr>
<tr>
<td>export.to.file</td>
<td>17</td>
</tr>
<tr>
<td>export.to.file&lt;-</td>
<td>18</td>
</tr>
<tr>
<td>file.format</td>
<td>18</td>
</tr>
<tr>
<td>file.format&lt;-</td>
<td>18</td>
</tr>
<tr>
<td>fo</td>
<td>18</td>
</tr>
<tr>
<td>init</td>
<td>19</td>
</tr>
<tr>
<td>integrity.check</td>
<td>21</td>
</tr>
<tr>
<td>LGD</td>
<td>22</td>
</tr>
<tr>
<td>LGD&lt;-</td>
<td>22</td>
</tr>
<tr>
<td>loss</td>
<td>22</td>
</tr>
<tr>
<td>loss.dist</td>
<td>22</td>
</tr>
<tr>
<td>loss.k</td>
<td>23</td>
</tr>
<tr>
<td>loss.unit</td>
<td>23</td>
</tr>
<tr>
<td>loss.unit&lt;-</td>
<td>23</td>
</tr>
<tr>
<td>M</td>
<td>23</td>
</tr>
<tr>
<td>measure</td>
<td>23</td>
</tr>
<tr>
<td>mu.k</td>
<td>24</td>
</tr>
<tr>
<td>name</td>
<td>24</td>
</tr>
<tr>
<td>name&lt;-</td>
<td>24</td>
</tr>
<tr>
<td>NC</td>
<td>24</td>
</tr>
<tr>
<td>NEX</td>
<td>24</td>
</tr>
<tr>
<td>NEX&lt;-</td>
<td>25</td>
</tr>
<tr>
<td>Niter.max</td>
<td>25</td>
</tr>
<tr>
<td>Niter.max&lt;-</td>
<td>25</td>
</tr>
<tr>
<td>NS</td>
<td>25</td>
</tr>
<tr>
<td>nu</td>
<td>25</td>
</tr>
<tr>
<td>path.in</td>
<td>26</td>
</tr>
</tbody>
</table>
R topics documented:

path.in<- ................................................................. 26
path.out ................................................................. 26
path.out<- ............................................................... 26
PD ................................................................. 26
PD.crp ............................................................... 27
PDF ................................................................. 27
pd_sector_var .......................................................... 27
PL ................................................................. 27
PL.crp ............................................................... 27
plausi ................................................................. 28
plot ................................................................. 28
PLOT.PDF .............................................................. 28
PLOT.PDF<- .......................................................... 28
PLOT.range.x .......................................................... 29
PLOT.range.x<- ........................................................ 29
PLOT.range.y .......................................................... 29
PLOT.range.y<- ........................................................ 29
PLOT.scale ............................................................ 30
PLOT.scale<- .......................................................... 30
port.name .............................................................. 30
port.name<- ............................................................. 30
portfolio ............................................................... 30
rating ................................................................. 30
rating.PD ............................................................... 31
rating.PD<- ............................................................ 31
rating.scale.name ....................................................... 31
rating.scale.name<- .................................................... 31
rating.SD .............................................................. 32
rating.SD<- .............................................................. 32
ratings<- ................................................................. 32
rating_pd ............................................................... 32
rc.sd ................................................................. 32
rc.vares .............................................................. 33
read ................................................................. 33
save.memory ........................................................... 33
save.memory<- ........................................................ 33
SD ................................................................. 34
SD.cont ............................................................... 34
SD.crp ............................................................... 34
sec.var ............................................................... 34
sec.var.est ............................................................ 34
sec.var.est<- .......................................................... 35
sec.var.name .......................................................... 35
sec.var.name<- ........................................................ 35
sec.var<- ............................................................... 35
set.changes ............................................................ 35
show ................................................................. 35
sigma.sqr.syst .......................................................... 36
Description


Details

The package provides the opportunity to analyze a given credit portfolio on a very simple level. Key numbers, that can be calculated are the expected loss, standard deviation, value at risk and expected shortfall on any confidence level, as well as risk contributions to them on counterparty level. The results (i.e. the loss distribution) are achieved by an analytical approach. Therefore a lot of theoretical assumptions are necessary. So please make yourself familiar with the framework of this model given in "CreditRisk+", First Boston Financial Products, 1997, before using it.

For first use have a look at crp.CSFP-class, crp.CSFP or init

Author(s)

Kevin Jakob & Dr. Matthias Fischer
Maintainer: Kevin Jakob <Kevin.Jakob.Research@gmail.com>

References

First Boston Financial Products, "CreditRisk+", 1997
### See Also

- `crp.CSFPClass crp.CSFPCrp.init`

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td><em>Get the parameter a of the model.</em></td>
</tr>
<tr>
<td>alpha</td>
<td><em>Get the desired VaR level of the model.</em></td>
</tr>
<tr>
<td>alpha.crp</td>
<td><em>Get the maximum cdf levels for VaR of the model.</em></td>
</tr>
<tr>
<td>alpha.max</td>
<td><em>Get the maximum cdf level for loss distribution</em></td>
</tr>
<tr>
<td>alpha.max&lt;-</td>
<td><em>Set the maximal desired cdf level</em></td>
</tr>
</tbody>
</table>

---

**Description**

The method returns the value of a (used for calculation of loss distribution) as numeric.

---

**Description**

The method returns the value of alpha as numeric.

---

**Description**

The method returns the value of alpha.crp as numeric. These are different from the desired CDF levels alpha because the CDF is not continuous.

---

**Description**

The method returns the value of alpha.max as numeric.

---

**Description**

The method changes the models value of alpha.max.
alpha<-

*Set the cdf level(s) for VaR*

**Description**

The method changes the level(s) for VaR.

---

\( B \)

*Get the parameter \( B \) of the model.*

**Description**

The method returns the value of \( B \) (used for calculation of loss distribution) as matrix.

---

`calc.portfolio.statistics`

*Calculating portfolio statistics*

**Description**

This method calculates simple portfolio key numbers such as the expected loss (EL), standard deviation (SD), potential loss (PL) and others from the original data. Later losses are discretized according to the loss unit and probabilities of default are adjusted. Also the standard deviation for the sectors will be calculated from the SD of the rating classes or given sector variances. Counterparties with potential loss equal to zero or PD of zero are removed for further calculations. All information will be printed.

**See Also**

`crp.CSFP-class`

---

`calc.rc`

*Set the state of calc.rc*

**Description**

The method changes the models value of `calc.rc`. 
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>calc.rc&lt;-</code></td>
<td>Get the value of <code>calc.rc</code>. &lt;br&gt;The method returns the value of <code>calc.rc</code> as logical.</td>
</tr>
<tr>
<td><code>cdf</code></td>
<td>Get the CDF of the model. &lt;br&gt;The method returns the value of CDF as numeric.</td>
</tr>
<tr>
<td><code>changes.calc.portfolio.statistics</code></td>
<td>Get the state of <code>changes.calc.portfolio.statistics</code>. &lt;br&gt;The method returns the value of <code>changes.calc.portfolio.statistics</code>.</td>
</tr>
<tr>
<td><code>changes.export</code></td>
<td>Get the state of <code>changes.export</code>. &lt;br&gt;The method returns the value of <code>changes.export</code>.</td>
</tr>
<tr>
<td><code>changes.loss</code></td>
<td>Get the state of <code>changes.loss</code>. &lt;br&gt;The method returns the value of <code>changes.loss</code>.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>changes.measure</td>
<td>The method returns the value of changes.measure.</td>
</tr>
<tr>
<td>changes.plausi</td>
<td>The method returns the value of changes.plausi.</td>
</tr>
<tr>
<td>changes.plot</td>
<td>The method returns the value of changes.plot.</td>
</tr>
<tr>
<td>changes.rc.sd</td>
<td>The method returns the value of changes.rc.sd.</td>
</tr>
<tr>
<td>changes.rc.vares</td>
<td>The method returns the value of changes.rc.vares.</td>
</tr>
</tbody>
</table>
### changes.read

**Get the state of changes.read**

**Description**

The method returns the value of changes.read.

### CP.NR

**Get the ID numbers of the counterparties of the model**

**Description**

The method returns the value of CP.NR as numeric.

### CP.NR<-  

**Set the ID numbers of the counterparties in the model**

**Description**

The method changes the value of CP.NR.

### CP.rating

**Get counterparties ratings**

**Description**

The method returns the value of CP.rating as numeric.

### CP.rating<-

**Set counterparties ratings**

**Description**

The method changes the value of CP.rating.
crp.CSFP  

Main routine for CSFP-model

Description

This is a summary of all methods you need during a complete run of the model (see details).

Usage

```r
crp.CSFP(this, skip.read)
```

Arguments

- **this**: object of class crp.CSFP
- **skip.read**: logical, indicating if the `read` function should be executed. I.e. it defines is input data should be read from input path. Set this value to FALSE if the model is already fully initialized. If portfolio data are manipulated manually, for example by `NEX(MyModel)=...` and recalculation should be performed, `skip.read` should be set to FALSE.

Details

It executes the following methods in the given order.
- `read` (if `skip.read = TRUE`, which is default)
- `plausi`
- `calc.portfolio.statistics`
- `loss.dist`
- `measure`
- `plot`
- `rc.vares`
- `rc.sd`
- `export`

If errors occur at `read` or `plausi` it stops. The state of `plot.PDF` and `calc.rc` are respected. During the computation, `save.memory` will be set to FALSE because of performance considerations. At the end, it will be switched back to the original state and all aftermath come into effect.

Value

Object of class crp.CSFP

See Also

- `read`, `plausi`, `calc.portfolio.statistics`, `loss.dist`, `measure`, `rc.vares`, `rc.sd`, `export`, `crp.CSFP-class`
Modelling credit risks based on the concept of "Credit Risk+".

Objects from the Class
Objects can be created by calls of the form MyModel=init(...).

Slots

portfolio data:

- cpNnr: is a numeric vector with the counterparty ID-numbers.
- cpNrating: is a numeric vector with the counterparty ratings.
- nex: is a numeric vector with the exposure of each counterparty.
- lgd: is a numeric vector with the counterparty specific LGDs
- pl: is a numeric vector with the potential loss for each counterparty before discretization.
- pd: is a numeric vector with the probability of default according to cpNrating for each counterparty before discretization.
- w: is a matrix with the sector weights for each counterparty.
- nu: is a numeric vector with the discrete losses on counterparty level as multiples of lossNunit.
- plNcrp: is a numeric vector with the potential loss for each counterparty adjusted to the discretization.
- pdNcrp: is a numeric vector with the probability of default according to cpNrating for each counterparty adjusted to the discretization.

rating information:

- rating: is a numeric vector with the different rating classes.
- rating.PD: is a numeric vector with the probabilities of default corresponding to the rating classes.
- rating.SD: is a numeric vector with the standard deviations of the probabilities of default corresponding to the rating classes. If you use sec.var.est=5 this slot is unused.

sector information:

- mu.k: is a numeric vector with the average number of defaults per sector mu.k=sum(W[1,k+1]*PD.crp)
- loss.k: is a numeric vector with the average loss per sector. It is defined by loss.K[k]=sum(W[1,k]*PD*PL).
- sigma.k: is a numeric vector with the sector standard deviation, calculated from rating.SD or sec.var according to sec.var.est.

control parameters:
sec.var.est: is an indicator for the mode, the sector standard deviations should be calculated.
1: Sum of (weights * SD)
2: [Sum of (weights * SD)] / MU(k)
3: Sum of (sqrt(weights) * SD)
4: [Sum of (sqrt(weights) * SD)] / MU(k)
5: Read variances from external file
with: MU.(k) = \text{sum}(W[,k+1]*PD)

loss.unit: is the discretization parameter for net exposures.

Niter.max: is the maximum number of exposure bands/probabilities being calculated.

alpha.max: If a number smaller one is insert, the calculation stops at this level of the CDF. For this mode, an upper bound of Niter.max.global is implemented to stop if this number of iterations is reached. If the desired confidence level is not reached till this threshold, a warning comes up. If you set Niter.max > Niter.max.global manually, the threshold will be ignored.

Niter.max.global: is the maximum number of iterations if alpha.max is specified. Be aware, that a high value can have high memory costs during the algorithm, even if the calculation of the CDF stops very much earlier.

alpha: is the vector of confidence levels (between 0 and 1), the Value at risk and expected shortfall should be calculated. It should be no problem if the entries are not in an ascending order. For the risk contributions only the last entry will be considered.

PLOT.PDF: is a flag for plotting the PDF during crp.csfp main routine or not. It will not be recognized if you start plot directly.

PLOT.scale: is a numeric value defining the scale for the horizontal axis (the losses) of the PDF plot.

PLOT.range.x: is a numeric vector with two entries representing the range on the x-axis (the losses) for the plot of the loss distribution. If you insert values smaller one, this will be interpreted as levels of the CDF. The default for PLOT.range.x / ~.y (0,0) means, that R will choose axis range by itself.

PLOT.range.y: is the same as PLOT.range.x above for vertical axis.

calc.rc: is a flag for calculating the risk contributions or not during crp.csfp main routine. It will not be recognized if you start rc.vares or rc.sd directly.

save.memory: is a switch for the save memory mode. If save.memory=TRUE, loss and CDF will not be stored permanently. If needed, the CDF is recalculated from the PDF. The same happens to a and B at the end of loss.dist if calc.rc=FALSE.

model information and risk measures:

NS: is a numeric value with the number of sectors.

NC: is a numeric value with the number of counterparties.

sec.var: is a numeric vector with the sector variances from sec.var.name.

EL: is a numeric value with the expected loss before discretization.

EL.crp: is a numeric value with the expected loss after discretization, calculated from the PDF. Differences to EL are caused by alpha.max or Niter.max being to small.

sigma_sqr_div: is a numeric value with the diversifiable part of SD.crp. The value is already squared.
sigma_sqr_syst: is a numeric value with the systemic part of SD. The value is already squared.

SD: is a numeric value with the portfolio standard deviation.

SD.crp: is a numeric value with the portfolio standard deviation after discretization, calculated from the PDF. Differences to SD are caused by loss.unit being to large or alpha.max being to small.

VaR: is a numeric vector containing the calculated value at risk for the given levels in alpha.

EC: is a numeric vector containing the economic capital for the given levels in alpha. In the CR+ framework this is defined as EC=VaR-EL.crp.

ES: is a numeric vector containing the expected shortfall for the given levels in alpha.

risk contributions:

VaR.cont: is a numeric vector with the risk contributions to the last entry in VaR for each counterparty.

ES.cont: is a numeric vector with the risk contributions to the last entry in ES for each counterparty.

ES.tau.cont: is a numeric vector with the risk contributions to TAU, corresponding to the the last entry in ES for each counterparty.

SD.cont: is a numeric vector with the risk contributions to the portfolio standard deviation.

loss distribution:

loss: is a numeric vector with the different losses / exposure bands, the PDF is calculated for.

PDF: is a numeric vector with the probability density function.

CDF: is a numeric vector with the cumulative distribution function.

providing model input:

input: is a temporary list, used if input data (portfolio, rating.scale, sec.var) are passed directly to init.

path.in: is a character string with the path to the directory, where are the input files. All input files have to be in this directory. The path should end with "\" or "/".

port.name: is a character string with the name of the portfolio file, ending with ".csv".

rating.scale.name: is a character string with the name of the rating file, ending with ".csv".

sec.var.name: is a character string with the name of the file containing the sector variances, ending with ".csv".

file.format: is a character string defining the format of the input files if no data frames are provided. You can choose between 'csv', which means that the separation character is ',' and the decimal character is '.' and 'csv2', which means that the separation character is ';' and the decimal character is ','.

model output:

export.to.file: logical, defining if results should be exported to path.out

path.out: is a character string with the path to the directory, where the output should be written to if export.to.file = TRUE, ending with "\" or "/". Be aware, that actually, by writing any output, the model will create a subdirectory in path.out with its name. So you can use
path.in also as path.out (which is the default case), create different models from the same input data (or different files, lying in the same directory), without worrying about the output path or overwriting other results.

**internal model parameters:**

- \( a \): is a numeric value calculated during `loss.dist`, necessary for the PDF and needed in `rc.vares`.
- \( B \): is a matrix calculated during `loss.dist`, necessary for the PDF and needed in `rc.vares`.
- \( M \): is a numeric value giving the maximal exposure band, which is calculated during `loss.dist()`. 
- `read.ok`: is a flag indicating if reading input files / data frames was successful.
- `plausi.ok`: is a flag indicating if the plausibility check was successful.
- `rc.ok`: is a flag indicating if the calculation of risk contributions to VaR and ES was successful.
- `VaR.pos`: is a numeric vector with the positions of VaR in CDF.
- `alpha.crp`: is a numeric vector containing the CDF-levels of VaR. Because they are from the calculated CDF `alpha.crp` will always be a little bit greater as `alpha`. With the help of `alpha.crpr` it is not necessary to store the CDF for calculating risk contributions to VaR and ES (`save.memory = TRUE` can be used).
- `name`: is a character string with the name of the model. This slot is set at the time you run the first method on your model, not by creation via `init`. If you change the model name (the name of the R object) `name` is updated automatically by the first method running on it.
- `changes.read`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `read()`.
- `changes.plausi`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `plausi()`.
- `changes.calc.portfolio.statistics`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `calc.portfolio.statistics`.
- `changes.loss`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `loss.dist`.
- `changes.measure`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `measure`.
- `changes.plot`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `plot`.
- `changes.rc.vares`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `rc.sd`.
- `changes.rc.sd`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `rc.vares`.
- `changes.export`: is an internal flag indicating changes on input parameters (by call to `<slot>`) affecting `export`.

**Methods**

- `calc.portfolio.statistics` Calculating portfolio statistics
- `loss.dist` Calculating the loss distribution
rc.sd Calculating risk contributions to standard deviation
rc.vares Calculating risk contributions to VaR and ES
crp.CSFP Main routine for CSFP-model
export Export risk contributions
measure Calculating portfolio measures
plausi Checking input data for plausibility
plot Plotting the PDF
read Reading the input files
summary Summarize portfolio key numbers
write.summary Writing summary to file
<slot> You can enter every slot via <slot> e.g. VaR(MyModel) gives the calculated value at risk.
<slot> <- You can change the value of a slot if this is an input parameter via <slot> <- e.g.
alpha(MyModel)<-c(0.999, 0.9995). You can do so with all slots that are available in init
and the slots containing portfolio information (CP.NR, NEX, CP.rating, LGD, W), rating information
(rating, rating.PD, rating.PD) and the sector variances sec.var

Author(s)
Kevin Jakob & Dr. Matthias Fischer

References
First Boston Financial Products, "CreditRisk+", 1997

See Also
init crp.CSFP

Examples
MMyModel=init(path.in = system.file("data",package="crp.CSFP"), loss.unit=1e6,
calc.rc=TRUE)

# or pass portfolio directly and use random sector variances
Path = system.file("data",package="crp.CSFP")
portfolio = read.csv(paste(Path,"/portfolio.csv",sep=""))
rating.scale = read.csv(paste(Path,"/rating_pd.csv",sep=""))
sec.var = data.frame(Var=runif(3,0,2))
MMyModel=init(portfolio=portfolio,rating.scale=rating.scale,sec.var=sec.var,
loss.unit=1e6,calc.rc=TRUE)

# execute portfolio calculation
MMyModel=crp.CSFP(MMyModel)
### crp.round

#### Description
The function rounds numerical values to the next integer. If the first digit after the decimal point is equal to 5 the function rounds up.

#### Usage
```
crp.round(a)
```

#### Arguments
- **a**
  
  A numerical value which should be rounded.

#### Value

numeric

#### Author(s)

Dr. Matthias Fischer

### EC

#### Description
The method returns the value of EC as numeric.

### EL

#### Description
The method returns the value of EL as numeric.

### EL.crp

#### Description
The method returns the value of EL.crp as numeric.
ES  Get the expected shortfall of the model

Description
The method returns the value of ES as numeric.

ES.cont  Get the expected shortfall contributions

Description
The method returns the value of ES.cont as numeric.

ES.tau.cont  Get the corresponding tau for expected shortfall contributions

Description
The method returns the value of ES.tau.cont as numeric.

export  Export risk contributions and loss distribution

Description
This method exports the risk contributions, calculated via rc.vares and rc.sd the loss distribution and a summary of the model to files named "RC.csv", "lossdist.csv" and "summary.csv" in the path.out/name directory. In save.memory mode, the content of the risk contribution attributes will be deleted afterwards.

See Also
rc.vares, rc.sd, crp.CSFP, crp.CSFP-class,

export.to.file  Get the status of export.to.file

Description
The method returns the current value of export.to.file.
**export.to.file** - *Set the state of export to file*

**Description**

The method changes the value of slot `export.to.file`.

---

**file.format** - *Get the file format of the model*

**Description**

The method returns the value of `file.format` as character.

---

**file.format<-** - *Set the file format*

**Description**

The method changes the models value of `file.format`.

---

**fo** - *Function to convert numerical output.*

**Description**

The Function writes numerical values as a multiple of the next power of 1000 with respect to three digits in order to make large values more readable. The trailed abbreviation corresponds to `1e3` (Thd), `1e6` (Mio), `1e9` (Bil), `1e12` (Tril).

**Usage**

`fo(x)`

**Arguments**

- `x` Numerical value that should be formated.

**Value**

A character string.

**Author(s)**

Kevin Jakob
**init**  

*Initializing a new entity of class crp.CSFP*

**Description**

This function helps to create a new entity of a crp.CSFP object. The arguments, given to the functions become the attributes of the new model.

**Usage**

```r
init(path.in = "", path.out = "", port.name = "portfolio.csv",  
     rating.scale.name = "rating_pd.csv", sec.var.name = "pd_sector_var.csv",  
     sec.var.est = 5, loss.unit = 1e+06, Niter.max = 0, alpha.max = 0.9999,  
     Niter.max.global = 1e+05, alpha = c(0.999), PLOT.PDF = TRUE,  
     export.to.file = FALSE, calc.rc = FALSE, PLOT.scale = 1e+06,  
     PLOT.range.x = c(0, 0), PLOT.range.y = c(0, 0), save.memory = FALSE,  
     file.format = "csv", portfolio = data.frame(), rating.scale = data.frame(),  
     sec.var = data.frame())
```

**Arguments**

- **portfolio** is a data frame containing the portfolio information. The structure has to be the same as described in `port.name` for the .csv files. If not provided, the file is read from `path.in`.

- **rating.scale** is a data frame containing the rating master scale. The structure has to be the same as described in `rating.scale.name` for the .csv files. If not provided, the file is read from `path.in`.

- **sec.var** is a data frame containing the sector variances. The structure has to be the same as described in `sec.var.name` for the .csv files. If not provided, the file is read from `path.in`.

- **path.in** is a character string with the path to the directory, where are the input files. All input files have to be in this directory. It must end with "...\" or ".../". In alternative the files can be passed as data frames to `init`, please have a look at `portfolio`, `rating.scale` or `sec.var` below.

- **file.format** is a character string defining the format of the input files. You can choose between 'csv', which means that the separation character is ',' and the decimal character is '.' and 'csv2', which means that the separation character is ';' and the decimal character is '.'.

- **path.out** is a character string with the path to the directory, where the output should be written to if `export.to.file` = TRUE, ending with "...\" or ".../". Be aware, that actually, by writing any output, the model will create a subdirectory in `path.out` with its name. So you can use your `path.in` as `path.out` (which is the default case), create different models from the same input data (or from different files, lying in the same directory), without worrying about the output path or overwriting other results.
port.name is a character string with the name of the portfolio file, ending with ".csv". The file must contain the following columns: CPnumber, CPname, exposure, lgd, maturity, rating, S1, S2, .... Take care of the right spelling of the column titles and capitalization.

rating.scale.name is a character string with the name of the rating file, ending with ".csv". The file must contain the following columns: RATING, PD, SD. The SD column is not necessary if sec.var.est=5. The rating classes have to be integer values. A class '0' is allowed. All counterparties in this class are removed before analyzing the portfolio. Take care of the right spelling of the column titles and capitalization.

sec.var.name is a character string with the name of the file containing the sector variances, ending with ".csv". The file must contain the column Var. Take care of the right spelling and capitalization. The file is not used if sec.var.est != 5.

sec.var.est is an indicator for the mode, the sector standard deviations should be calculated.
1: \text{Sum of (weights} \times \text{SD)}
2: \left \lfloor \text{Sum of (weights} \times \text{SD)} \right \rfloor / \text{MU(k)}
3: \text{Sum of (sqrt(weights)} \times \text{SD)}
4: \left \lfloor \text{Sum of (sqrt(weights)} \times \text{SD)} \right \rfloor / \text{MU(k)}
5: \text{Read variances from external file}
with: \text{MU.(k)=sum(W[k],k+1]xPD)}

loss.unit is the discretization parameter for net exposures.
Niter.max is the maximum number of exposure bands/probabilities being calculated.
alpha.max in alternative to Niter.max, one can also define the maximum CDF level.
Niter.max.global is the maximum number of iterations if alpha.max is provided. Be aware, that a high value can have high memory costs during the algorithm, even if the calculation of the CDF stops very much earlier.

alpha is the vector of confidence levels (between 0 and 1), the Value at risk and expected shortfall should be calculated. It should be no problem if the entries are not in an ascending order. For the risk contributions only the last entry will be considered.

PLOT.PDF is a logical indicator for plotting the PDF or not. It will not be recognized if you start plot directly.
export.to.file is a logical indicator defining if loss distribution, risk contributions and a summary should be exported to path.out.
calc.rc is a flag for calculating the risk contributions or not. It will not be recognized if you start rc.vares or rc.sd directly.
PLOT.scale is a numeric value defining the scale for the horizontal axis (the losses) of the plot of the PDF.
PLOT.range.x is a numeric vector with two entries representing the range on the x-axis (the losses) for the plot of the loss distribution. If you insert values smaller one, this will be interpreted as levels of the CDF. The defaults for PLOT.range.x /~ y (0,0) have the meaning, that R will choose axis range by itself.
PLOT.range.y is the same as PLOT.range.x above for vertical axis.
**integrity.check**

`save.memory` is a switch for the save memory mode. If `save.memory=TRUE`, `loss` and `CDF` will not be stored permanently. If needed, the CDF is recalculated from the PDF. The same happens to `a` and `B` at the end of `loss.dist` if `calc.rc=FALSE`.

**Value**

A new object of class `crp.CSFP`.

**Author(s)**

Kevin Jakob & Dr. Matthias Fischer

**See Also**

`crp.CSFP`

**Examples**

```r
mymodel = init(path.in = system.file("data", package = "crp.CSFP"), loss.unit = 1e6, calc.rc = TRUE)

# or pass portfolio directly and use random sector variances
Path = system.file("data", package = "crp.CSFP")
portfolio = read.csv(paste(Path, "/portfolio.csv", sep = ""))
rating.scale = read.csv(paste(Path, "/rating_pd.csv", sep = ""))
sec.var = data.frame(Var = runif(3, 0, 2))

mymodel = init(portfolio = portfolio, rating.scale = rating.scale, sec.var = sec.var,
loss.unit = 1e6, calc.rc = TRUE)
```

---

**integrity.check**  
*Internal method to ensure model integrity*

**Description**

This method is called each time you access a slot by `<slot>`. It checks the state of internal integrity flags in order to guarantee, that the slot you want to access has a value, consistent with the rest of the model.

For example, if you run `crp.CSFP` (complete calculation of the model), then change `loss.unit` by `loss.unit(MyModel)<...` to another value (which causes a call to `set.changes`) and want to access `EL.crp`, a warning message is printed because you did not recalculate `EL.crp`. Instead you have to run `calc.portfolio.statistics, loss.dist` and measure again to get the right value for `EL.crp`.

**See Also**

`crp.CSFP-class, crp.CSFP`
**LGD**

*Get the loss given defaults of the model*

**Description**

The method returns the value of LGD as numeric.

**LGD<-**

*Set the counterparty specific LGDs*

**Description**

The method changes the LGDs of counterparties.

**loss**

*Get the several losses (exposure bands) of the model*

**Description**

The method returns the value of loss as numeric.

**loss.dist**

*Calculating the loss distribution*

**Description**

This method uses an algorithm from Gundlach/Lehrbass p.74f to compute the loss distribution. A small modification on the structure of the two loops was done to compute the CDF parallel to the PDF and stop if a desired level is reached. Data a and B (corresponding to alpha and b in Gundlach/Lehrbass p.74), necessary for risk contributions, will be erased at the end if calc.rc=FALSE and save.memory=TRUE.

**See Also**

rc.vares, crp.CSFp-class, crp.CSFp,
loss.k

Get the expected loss per sector

Description
The method returns the value of loss.k as numeric.

loss.unit

Get the loss unit of the model

Description
The method returns the value of loss.unit as numeric.

loss.unit<-  Set the loss unit

Description
The method changes the models value of loss.unit

M

Get the number of iterations for loss distribution

Description
The method returns the value of M as numeric.

measure

Calculating portfolio measures

Description
This method calculates key numbers from the discrete loss distribution, such as expected loss EL.crp, standard deviation SD.crp, value at risk VaR and expected shortfall ES to the given levels in alpha, prints them on the screen and uses write.summary to write them - together with the input parameter - to the output directory.

See Also

crp.CSFP, crp.CSFP-class,
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\mu_k)</td>
<td>Get the expected number of defaults per sector</td>
</tr>
<tr>
<td>name</td>
<td>Get the name of the model</td>
</tr>
<tr>
<td>name&lt;-</td>
<td>Set the name of the model</td>
</tr>
<tr>
<td>NC</td>
<td>Get the number of counterparties in the model</td>
</tr>
<tr>
<td>NEX</td>
<td>Get the net exposure per counterparty</td>
</tr>
</tbody>
</table>

**Description**

The method returns the value of \(\mu_k\) as numeric.

The method returns the value of name as character.

The method changes the value of name.

The method returns the value of NC as numeric.

The method returns the value of NEX as numeric.
NEX<-  
Set the net exposure per counterparty

**Description**

The method changes the value of NEX.

---

Niter.max
Get the desired number of iterations or cdf level for loss distribution

**Description**

The method returns the value of Niter.max as numeric.

---

Niter.max<-  
Set the maximal number of iterations or desired cdf level

**Description**

The method changes the models value of Niter.max

---

NS
Get the number of sectors of the model

**Description**

The method returns the value of NS as numeric.

---

nu
Get the discrete losses of the model

**Description**

The method returns the value of nu as numeric.
path.in \hspace{1cm} \textit{Get the input path of the model}

**Description**

The method returns the value of \texttt{path.in} as character.

path.in<- \hspace{1cm} \textit{Set input path}

**Description**

The method changes the models value of \texttt{path.in}

path.out \hspace{1cm} \textit{Get the output path of the model}

**Description**

The method returns the value of \texttt{path.out} as character.

path.out<- \hspace{1cm} \textit{Set output path}

**Description**

The method changes the models value of \texttt{path.out}

\texttt{PD} \hspace{1cm} \textit{Get the counterparty probabilities of default of the model}

**Description**

The method returns the value of \texttt{PD} as numeric.
Description

The method returns the value of \texttt{PD.crp} as numeric.

\begin{description}
\item[PDF] \textit{Get the PDF of the model}
\end{description}

Description

The method returns the value of PDF as numeric.

\begin{description}
\item[pd\_sector\_var] \textit{Sector variances for the Credit Suisse example portfolio}
\end{description}

Description

The file contains the sector variances corresponding to the example portfolio from the Credit Suisse First Boston for CreditRisk+.

References

First Boston Financial Products, "CreditRisk+$", 1997

\begin{description}
\item[PL] \textit{Get the potential losses per counterparty}
\end{description}

Description

The method returns the value of PL as numeric.

\begin{description}
\item[PL.crp] \textit{Get the potential losses per counterparty after discretization}
\end{description}

Description

The method returns the value of PL.crp as numeric.
plausi Checking input data for plausibility

Description

This method checks the input data for plausibility. The following checks are done:
- The PD for ratign classes should be non-decreasing according to the rating.
- The exposure should not be negative.
- The sector weights of each CP should not be negative and sum up to a number less or equal 1.
- The sector variances should not be negative.

See Also

- crp.CSFP, crp.CSFP-class.

plot Plotting the PDF

Description

This method creates a plot in a new window of the loss distribution together with lines indicating the EL, crp, VaR and ES. The X-/Y- ranges are taken from plot.range.x /-y.

See Also

- crp.CSFP, crp.CSFP-class.

PLOT.PDF Get the state of PLOT.PDF

Description

The method returns the value of PLOT.PDF as logical.

PLOT.PDF<- Set the state of PLOT.PDF

Description

The method changes the models value of PLOT.PDF
PLOT.range.x

Get the plot range for losses

Description

The method returns the value of PLOT.range.x as numeric.

PLOT.range.x<- Set the plot range for the losses

Description

The method changes the models value of PLOT.range.x

PLOT.range.y

Get the plot range for probabilities

Description

The method returns the value of PLOT.range.y as numeric.

PLOT.range.y<- Set the plot range for the probabilities

Description

The method changes the models value of PLOT.range.y

PLOT.scale

Get the plot scale for losses

Description

The method returns the value of PLOT.scale as numeric.
### PLOT.scale<-

*Set the plot scale for portfolio losses*

**Description**

The method changes the models value of PLOT.scale

---

### port.name

*Get the name of the portfolio file*

**Description**

The method returns the value of port.name as character.

---

### port.name<-

*Set the name for the portfolio file*

**Description**

The method changes the models value of port.name

---

### portfolio

*Portfolio data for the Credit Suisse example portfolio*

**Description**

The file contains the portfolio corresponding to the example portfolio from the Credit Suisse First Boston for CreditRisk+.

**References**

First Boston Financial Products, "CreditRisk+", 1997

---

### rating

*Get the rating classes of the model*

**Description**

The method returns the value of rating as numeric.
**rating.PD**

*Get the PDs of rating classes*

**Description**

The method returns the value of rating.PD as numeric.

**Methods**

```
signature(this = "crp.CSF"")
```

---

**rating.PD<-**

*Set the PDs for rating classes*

**Description**

The method changes the PDs of the corresponding rating classes.

---

**rating.scale.name**

*Get the name of the file containing the risk matrix of the model*

**Description**

The method returns the value of rating.scale.name as character.

---

**rating.scale.name<-**

*Set the name for the file containing the rating scale*

**Description**

The method changes the models value of rating.scale.name

---

**rating.SD**

*Get the standard deviations corresponding to rating classes*

**Description**

The method returns the value of rating.SD as numeric.
**rating.Nsd**

*Set the standard deviations corresponding to rating classes*

**Description**

The method changes the value of `rating.Nsd`.

**rating<-**

*Set the rating classes of the model*

**Description**

The method changes the value of `rating`.

**rating_pd**

*Risk matrix for the Credit Suisse example portfolio*

**Description**

The file contains the risk matrix corresponding to the example portfolio from the Credit Suisse First Boston for CreditRisk+.

**References**

First Boston Financial Products, "CreditRisk+", 1997

**rc.sd**

*Calculating risk contributions to standard deviation*

**Description**

This method calculates the contributions on counterparty level to the portfolio standard deviation. No data out of loss.dist are required.

**See Also**

`rc.vares, export.crp.CSFP-class, crp.CSFP`
Calculating risk contributions to VaR and ES

Description

This method calculates the risk contributions on counterparty level to the value at risk (VaR), expected shortfall (ES) and the corresponding TAU. The confidence level that is taken is the last entry in the models alpha vector. It is necessary, that loss.dist was executed before, to compute and save required data.

See Also

loss.dist, rc.sd, export, crp.CSFp-class, crp.CSFp.

Reading the input files

Description

This method reads the input files port.name, rating.scale.name and sec.var.name if needed (i.e. if sec.var.est=5) from the path.in directory and matches the data to the corresponding slots. Counterparties with NEX=0, LGD=0, CP.rating=0 or PD=0 are removed from the portfolio. Please make sure, that the input files have the correct form given in init. In the context of the main routine crp.CSFp one can skip this method by setting skip.read=TRUE.

See Also

init, crp.CSFp, crp.CSFp-class.

Get the state of save.memory

Description

The method returns the value of save.memory as logical.

Set the state of save.memory

Description

The method changes the models value of save.memory
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td><em>Get the standard deviation of the model</em></td>
</tr>
<tr>
<td>SD.cont</td>
<td><em>Get the contributions to standard deviation</em></td>
</tr>
<tr>
<td>SD.crp</td>
<td><em>Get the discretized standard deviation of loss distribution</em></td>
</tr>
<tr>
<td>sec.var</td>
<td><em>Get self estimated sector variances</em></td>
</tr>
<tr>
<td>sec.var.est</td>
<td><em>Get the mode for sector variance estimation</em></td>
</tr>
</tbody>
</table>

**Description**

- **SD**: The method returns the value of SD as numeric.
- **SD.cont**: The method returns the value of SD.cont as numeric.
- **SD.crp**: The method returns the value of SD.crp as numeric.
- **sec.var**: The method returns the value of sec.var as numeric.
- **sec.var.est**: The method returns the models value of sec.var.est.
sec.var.est<-  

Set the mode for sector variance estimation

Description
The method changes the models value of sec.var.est

sec.var.name

Set the name of the file with the sector variances

Description
The method changes the value of sec.var.name.

sec.var.name<-  

Set the name of the file with the sector variances

Description
The method changes the value of sec.var.name.

sec.var<-  

Set self estimated sector variances

Description
The method changes the models value of sec.var

set.changes

Internal method for model integrity

Description
The method is called by set.changes every time an input parameter is changed by an <slot> method in order to set flags indicating if the integrity of the model is satisfied or at which point in the algorithms changes come into effect.

See Also
integrity.check crp.CSFP crp.CSFP-class
show

*Show summary of object crp.CSFP*

**Description**

This method extends the show function for objects of class crp.CSFP. A short summary with the most important model parameters is printed.

**See Also**

`show crp.CSFP`, `crp.CSFP-class`.

---

**sigma.sqr.syst**

*Get the systematik risk of the model*

**Description**

The method returns the value of `sigma_sqr_syst` (already squared) as numeric. The variable/method was renamed from `sigma.sqr.syst` to `sigma_sqr_syst` for S3 compatibility.

---

**sigma_k**

*Get the sector standard deviation*

**Description**

The method returns the value of `sigma_k` as numeric. The variable/method was renamed from `sigma.k` to `sigma_k` for S3 compatibility.

---

**sigma_sqr_div**

*Get the diversifiable risk of the model*

**Description**

The method returns the value of `sigma_sqr_div` (already squared) as numeric. The variable/method was renamed from `sigma.sqr.div` to `sigma_sqr_div` for S3 compatibility.
**summary**

### Summarize portfolio key numbers

**Description**

This method summarizes the input parameter, portfolio statistics from `calc.portfolio.statistics` and the results of `measure` in a list.

**See Also**

`write.summary`, `crp.CSFP`, `crp.CSFP-class`.

---

**VaR**

### Get the value at risk of the model

**Description**

The method returns the value of `VaR` as numeric.

---

**VaR.cont**

### Get the value at risk contributions on counterparty level

**Description**

The method returns the value of `VaR.cont` as numeric.

---

**VaR.pos**

### Get the position of value at risk in CDF

**Description**

The method returns the value of `VaR.pos` as numeric.

---

**w**

### Get the sector weights of counterparties

**Description**

The method returns the value of `w` as matrix.
write.summary

\[ w \leftarrow \]

*Set the sector weights of counterparties*

**Description**

The method changes the value of \( w \).

---

write.summary

*Writing summary to file*

**Description**

This method writes the list, created by `crp.summary` into a file, called "summary.csv" into the output directory.

**See Also**

`summary.crp.CSFP`, `crp.CSFP-class`
# Index

- **Topic** rating.PD, 31
- **Topic** classes  
  crp.CSFP-class, 11
- **Topic** datasets  
  pd_sector_var, 27  
  portfolio, 30  
  rating_pd, 32
- **Topic** methods  
  a, 5  
  alpha, 5  
  alpha.crp, 5  
  alpha.max, 5  
  alpha.max<=, 5  
  alpha<=, 6  
  B, 6  
  calc.portfolio.statistics, 6  
  calc.rc, 6  
  calc.rc<=, 7  
  CDF, 7  
  changes.calc.portfolio.statistics, 7  
  changes.export, 7  
  changes.loss, 7  
  changes.measure, 8  
  changes.plausi, 8  
  changes.plot, 8  
  changes.rc.sd, 8  
  changes.rc.vares, 8  
  changes.read, 9  
  CP.NR, 9  
  CP.NR<=, 9  
  CP.rating, 9  
  CP.rating<=, 9  
  crp.CSFP, 10  
  EC, 16  
  EL, 16  
  EL.crp, 16  
  ES, 17  
  ES.cont, 17  
  ES.tau.cont, 17  
  export, 17  
  export.to.file, 17  
  export.to.file<=, 18  
  file.format, 18  
  file.format<=, 18  
  integrity.check, 21  
  LGD, 22  
  loss, 22  
  loss.dist, 22  
  loss.k, 23  
  loss.unit, 23  
  loss.unit<=, 23  
  M, 23  
  measure, 23  
  mu.k, 24  
  name, 24  
  name<=, 24  
  NC, 24  
  NEX, 24  
  NEX<=, 25  
  Niter.max, 25  
  Niter.max<=, 25  
  NS, 25  
  nu, 25  
  path.in, 26  
  path.in<=, 26  
  path.out, 26  
  path.out<=, 26  
  PD, 26  
  PD.crp, 27  
  PDF, 27  
  PL, 27  
  PL.crp, 27  
  plausi, 28  
  plot, 28
plotNpdf, 28
plotNpdf<M, 28
plotNrangeNx, 29
plotNrangeNx<M, 29
plotNrangeNy, 29
plotNrangeNy<M, 29
plotNscale, 29
plotNscale<M, 30
portNname, 30
portNname<M, 30
rating, 30
rating.PD, 31
rating.PD<-, 31
rating.scale.name, 31
rating.scale.name<M, 31
rating.SD, 31
rating.SD<-, 32
rating<-, 32
rc.sd, 32
rc.vares, 33
read, 33
save.memory, 33
save.memory<-, 33
SD, 34
SD.cont, 34
SD.crp, 34
sec.var, 34
sec.var.est, 34
sec.var.est<-, 35
sec.var.name, 35
sec.var.name<-, 35
sec.var<-, 35
set.changes, 35
show, 36
sigma.sqr.syst, 36
sigma.k, 36
sigma.sqr.div, 36
summary, 37
VaR, 37
VaR.cont, 37
VaR.pos, 37
W, 37
W<-, 38
write.summary, 38

a, 5
a, crp.CSFP-method (a), 5
a-methods (a), 5
alpha, 5
alpha, crp.CSFP-method (alpha), 5
alpha-methods (alpha), 5
alpha.crp, 5
alpha.crp, crp.CSFP-method (alpha.crp), 5
alpha.crp-methods (alpha.crp), 5
alpha.max, 5
alpha.max, crp.CSFP-method (alpha.max), 5
alpha.max-methods (alpha.max), 5
alpha.max<-, 5
alpha.max<-, crp.CSFP, numeric-method
(alpha.max<->), 5
alpha.max<--methods (alpha.max<--), 5
alpha<-, 6
alpha<-, crp.CSFP, numeric-method
(alpha<->), 6
alpha<--methods (alpha<--), 6
B, 6
B, crp.CSFP-method (B), 6
B-methods (B), 6
calc.portfolio.statistics, 6, 10
calc.portfolio.statistics, crp.CSFP-method
calc.portfolio.statistics, 6
calc.portfolio.statistics-methods
calc.portfolio.statistics, 6
calc.rc, 6
calc.rc, crp.CSFP-method (calc.rc), 6
calc.rc-methods (calc.rc), 6
calc.rc<-, 7
calc.rc<-, crp.CSFP, logical-method
calc.rc<->), 7
calc.rc<--methods (calc.rc<--), 7
calc.rc<--methods (calc.rc<--), 7
calc.rc<--methods (calc.rc<--), 7
CDF, 7
CDF, crp.CSFP-method (CDF), 7
CDF-methods (CDF), 7
changes.calc.portfolio.statistics, 7
changes.calc.portfolio.statistics, crp.CSFP-method
(changes.calc.portfolio.statistics), 7
changes.calc.portfolio.statistics-methods
(changes.calc.portfolio.statistics), 7
changes.export, 7
changes.export, crp.CSFP-method
(changes.export), 7

changes.export-methods
(changes.export), 7

changes.loss, 7

changes.loss, crp.CSFP-method
(changes.loss), 7

changes.loss-methods (changes.loss), 7

changes.measure (changes.loss), 7

changes.measure, crp.CSFP-method
(changes.measure), 8

changes.measure-methods
(changes.measure), 8

changes.plausi, 8

changes.plausi, crp.CSFP-method
(changes.plausi), 8

changes.plausi-methods
(changes.plausi), 8

changes.plot, 8

changes.plot, crp.CSFP-method
(changes.plot), 8

changes.plot-methods (changes.plot), 8

changes.rc.sd, 8

changes.rc.sd, crp.CSFP-method
(changes.rc.sd), 8

changes.rc.sd-methods (changes.rc.sd), 8

changes.rc.vares, 8

changes.rc.vares, crp.CSFP-method
(changes.rc.vares), 8

changes.rc.vares-methods
(changes.rc.vares), 8

changes.read, 9

changes.read, crp.CSFP-method
(changes.read), 9

changes.read-methods (changes.read), 9

CP.NR, 9

CP.NR, crp.CSFP-method (CP.NR), 9

CP.NR-methods (CP.NR), 9

CP.NR<-, 9

CP.NR<-, crp.CSFP, numeric-method
(CP.NR<-,), 9

CP.NR<--methods (CP.NR<--), 9

CP.rating, 9

CP.rating, crp.CSFP-method (CP.rating), 9

CP.rating-methods (CP.rating), 9

CP.rating<-, 9

CP.rating<-, crp.CSFP, numeric-method
(CP.rating<-,), 9

CP.rating<--methods (CP.rating<--), 9

CP.rating<--methods (CP.rating<--), 9

crp.CSFP, crp.CSFP, logical-method
(crp.CSFP), 10

crp.CSFP, crp.CSFP, missing-method
(crp.CSFP), 10

crp.CSFP, crp.CSFP-method (crp.CSFP), 10

crp.CSFP-class, 11

crp.CSFP-methods (crp.CSFP), 10

crp.CSFP-package, 4

crp.CSFP.loss (loss.dist), 22

crp.CSFP.loss, crp.CSFP-method
(loss.dist), 22

crp.CSFP.rc.sd (rc.sd), 32

crp.CSFP.rc.sd, crp.CSFP-method (rc.sd), 32

crp.CSFP.rc.vares (rc.vares), 33

crp.CSFP.rc.vares, crp.CSFP-method
(rc.vares), 33

crp.export (export), 17

crp.export, crp.CSFP-method (export), 17

crp.init, 5

crp.init (init), 19

crp.measure (measure), 23

crp.measure, crp.CSFP-method (measure), 23

crp.plausi (plausi), 28

crp.plausi, crp.CSFP-method (plausi), 28

crp.plot (plot), 28

crp.plot, crp.CSFP-method (plot), 28

crp.read (read), 33

crp.read, crp.CSFP-method (read), 33

crp.round, 16

crp.set.changes (set.changes), 35

crp.set.changes, character, crp.CSFP-method
(set.changes), 35

crp.summary (summary), 37

crp.summary, crp.CSFP-method (summary), 37

crp.write.summary (write.summary), 38

crp.write.summary, crp.CSFP-method
(write.summary), 38

crp.write.summary-methods
(write.summary), 38
EC, 16
EC, crp, CSFP-method (EC), 16
EC-methods (EC), 16
EL, 16
EL, crp, CSFP-method (EL), 16
EL-methods (EL), 16
EL, crp, crp, CSFP-method (EL, crp), 16
EL, crp-methods (EL, crp), 16
ES, 17
ES, crp, CSFP-method (ES), 17
ES-methods (ES), 17
ES, cont, 17
ES, cont, crp, CSFP-method (ES, cont), 17
ES-methods (ES, cont), 17
ES, tau, cont, 17
ES, tau, cont, crp, CSFP-method (ES, tau, cont), 17
ES, tau, cont-methods (ES, tau, cont), 17
export, 10, 17, 32, 33
export, crp, CSFP-method (export), 17
export-methods (export), 17
export to file, 17
export to file, crp, CSFP-method (export to file), 17
export to file-methods (export to file), 17
export to file<-, 18
export to file<-, crp, CSFP, logical-method (export to file<->), 18
export to file<--methods (export to file<--), 18
file.format, 18
file.format, crp, CSFP-method (file.format), 18
file.format-methods (file.format), 18
file.format<-, 18
file.format<-, crp, CSFP, character-method (file.format<->), 18
file.format<--methods (file.format<--), 18
fo, 18
get.a (a), 5
get.a, crp, CSFP-method (a), 5
get.ALPHA (alpha), 5
get.ALPHA, crp, CSFP-method (alpha), 5
get.ALPHA, crp (alpha, crp), 5
get.ALPHA, crp, crp, CSFP-method (alpha, crp), 5
get.ALPHA, MAX (alpha, max), 5
get.ALPHA, MAX, crp, CSFP-method (alpha, max), 5
get.B (B), 6
get.B, crp, CSFP-method (B), 6
get.CALC.RISK.CONT (calc.rc), 6
get.CALC.RISK.CONT, crp, CSFP-method (calc.rc), 6
get.CDF (CDF), 7
get.CDF, crp, CSFP-method (CDF), 7
get.changes, crp, calc.portfolio.statistics
(changes, calc.portfolio.statistics), 7
get.changes, crp, calc.portfolio.statistics, crp, CSFP-method
(changes, calc.portfolio.statistics), 7
get.changes, crp, CSFP, loss
(changes, loss), 7
get.changes, crp, CSFP, loss, crp, CSFP-method
(changes, loss), 7
get.changes, crp, CSFP, rc.sd
(changes, rc.sd), 8
get.changes, crp, CRP, rc.sd, crp, CSFP-method
(changes, rc.sd), 8
get.changes, crp, CSFP, rc.varess
(changes, rc.varess), 8
get.changes, crp, CSFP, rc.varess, crp, CSFP-method
(changes, rc.varess), 8
get.changes, crp, export
(changes, export), 7
get.changes, crp, export, crp, CSFP-method
(changes, export), 7
get.changes, crp, measure
(changes, measure), 8
get.changes, crp, measure, crp, CSFP-method
(changes, measure), 8
get.changes, crp, plausi
(changes, plausi), 8
get.changes, crp, plausi, crp, CSFP-method
(changes, plausi), 8
get.changes, crp, plausi, crp, CSFP-method
(changes, plausi), 8
get.changes, crp, plot
(changes, plot), 8
get.changes, crp, plot, crp, CSFP-method
(changes, plot), 8
get.changes, crp, read
(changes, read), 9
INDEX

get.changes.crp.read, crp.CSFP-method (changes.read), 9
get.CP.NR (CP.NR), 9
get.CP.NR, crp.CSFP-method (CP.NR), 9
get.CP.RATING (CP.rating), 9
get.CP.RATING, crp.CSFP-method (CP.rating), 9
get.ES (ES), 17
get.ES, crp.CSFP-method (ES), 17
get.ES.CONT (ES.cont), 17
get.ES.CONT, crp.CSFP-method (ES.cont), 17
get.ES.TAU (ES.tau), 17
get.ES.TAU, crp.CSFP-method (ES.tau.cont), 17
get.file.format (file.format), 18
get.file.format, crp.CSFP-method (file.format), 18
get.LGD (LGD), 22
get.LGD, crp.CSFP-method (LGD), 22
get.LOS (loss), 22
get.LOS, crp.CSFP-method (loss), 22
get.LOS.OUT (loss.unit), 23
get.LOS.OUT, crp.CSFP-method (loss.unit), 23
get.M (M), 23
get.M, crp.CSFP-method (M), 23
get.MU.K (mu.k), 24
get.MU.K, crp.CSFP-method (mu.k), 24
get.NAME (name), 24
get.NAME, crp.CSFP-method (name), 24
get.NC (NC), 24
get.NC, crp.CSFP-method (NC), 24
get.NEX (NEX), 24
get.NEX, crp.CSFP-method (NEX), 24
get.NITER.MAX (Niter.max), 25
get.NITER.MAX, crp.CSFP-method (Niter.max), 25
get.NS (NS), 25
get.NS, crp.CSFP-method (NS), 25
get.NU (nu), 25
get.NU, crp.CSFP-method (nu), 25
get.PATH.IN (path.in), 26
generate.PATH.IN, crp.CSFP-method (path.in), 26
generate.PATH.OUT (path.out), 26
generate.PATH.OUT, crp.CSFP-method (path.out), 26
generate.PD (PD.crp), 27
generate.PD, crp.CSFP-method (PD.crp), 27
generate.PD0 (PD), 26
generate.PD0, crp.CSFP-method (PD), 26
generate.PDF (PDF), 27
generate.PDF, crp.CSFP-method (PDF), 27
generate.PDVAR.NAME (sec.var.name), 35
generate.PDVAR.NAME, crp.CSFP-method (sec.var.name), 35
generate.PL (PL.crp), 27
generate.PL, crp.CSFP-method (PL.crp), 27
generate.PL0 (PL), 27
generate.PL0, crp.CSFP-method (PL), 27
generate.PLOT.PDF (PLOT.PDF), 28
generate.PLOT.PDF, crp.CSFP-method (PLOT.PDF), 28
generate.PLOT.RANGE.X (PLOT.range.x), 29
generate.PLOT.RANGE.X, crp.CSFP-method (PLOT.range.x), 29
generate.PLOT.RANGE.Y (PLOT.range.y), 29
generate.PLOT.RANGE.Y, crp.CSFP-method (PLOT.range.y), 29
generate.PLOT.SCALE (PLOT.scale), 29
generate.PLOT.SCALE, crp.CSFP-method (PLOT.scale), 29
generate.PORT.NAME (port.name), 30
generate.PORT.NAME, crp.CSFP-method (port.name), 30
generate.rating (rating), 30
generate.rating, crp.CSFP-method (rating), 30
generate.rating.PD (rating.PD), 31
generate.rating.PD, crp.CSFP-method (rating.PD), 31
generate.rating.SD (rating.SD), 31
generate.rating.SD, crp.CSFP-method (rating.SD), 31
generate.RISK.NAME (rating.scale.name), 31
generate.RISK.NAME, crp.CSFP-method (rating.scale.name), 31
generate.save.memory (save.memory), 33
generate.save.memory, crp.CSFP-method (save.memory), 33
getSD(SD), 34
getSD, crp.CSFP-method(SD), 34
getSD, CONT(SD.cont), 34
getSD, CONT, crp.CSFP-method(SD.cont), 34
getSD, crp(SD.crp), 34
getSD, crp, crp.CSFP-method(SD.crp), 34
getSEC, VAR(sec.var), 34
getSEC, VAR, crp.CSFP-method(sec.var), 34
getSEC, VAR.EST(sec.var.est), 34
getSELV(VaR.pos), 37
getSELV, crp.CSFP-method(VaR.pos), 37
getSIGMA.K(sigma_k), 36
getSIGMA.K, crp.CSFP-method(sigma_k), 36
getSIGMA2_DIV(sigma_sqr_div), 36
getSIGMA2_DIV, crp.CSFP-method(sigma_sqr_div), 36
getSIGMA2_SYST(sigma.sqr.syst), 36
getSIGMA2_SYST, crp.CSFP-method(sigma.sqr.syst), 36
getV.K(loss.k), 23
getV.K, crp.CSFP-method(loss.k), 23
getVaR(VaR), 37
getVaR, crp.CSFP-method(VaR), 37
getVaR, CONT(VaR.cont), 37
getVaR, CONT, crp.CSFP-method(VaR.cont), 37
getW(W), 37
getW, crp.CSFP-method(W), 37
init, 4, 15, 19, 33
integrity.check, 21, 35
integrity.check, character, crp.CSFP-method(integrity.check), 21
integrity.check-methods(integrity.check), 21
LGDO, 22
LGDO, crp.CSFP-method(LGDO), 22
LGDO-methods(LGDO), 22
LGDO<-, 22
LGDO<-, crp.CSFP, numeric-method(LGDO<-, 22
LGDO<--methods(LGDO<-), 22
loss, 22
loss, crp.CSFP-method(loss), 22
loss-methods(loss), 22
loss.dist, 10, 22, 33
loss.dist, crp.CSFP-method(loss.dist), 22
loss.dist-methods(loss.dist), 22
loss.k, 23
loss.k, crp.CSFP-method(loss.k), 23
loss.k-methods(loss.k), 23
loss.unit, 23
loss.unit, crp.CSFP-method(loss.unit), 23
loss.unit-methods(loss.unit), 23
loss.unit<-, 23
loss.unit<-, crp.CSFP, numeric-method(loss.unit<-), 23
loss.unit<--methods(loss.unit<->), 23
M, 23
M, crp.CSFP-method(M), 23
M-methods(M), 23
measure, 10, 23
measure, crp.CSFP-method(measure), 23
measure-methods(measure), 23
mu.k, 24
mu.k, crp.CSFP-method(mu.k), 24
mu.k-methods(mu.k), 24
name, 24
name, crp.CSFP-method(name), 24
name-methods(name), 24
name<-, 24
name<-, crp.CSFP, character-method(name<-), 24
name<--methods(name<-), 24
NC, 24
NC, crp.CSFP-method(NC), 24
NC-methods(NC), 24
NEX, 24
NEX, crp.CSFP-method(NEX), 24
NEX-methods(NEX), 24
NEX<-, 25
NEX<-, crp.CSFP, numeric-method(NEX<-), 25
NEX<--methods(NEX<--), 25
Niter.max, 25
Niter.max, crp.CSFP-method(Niter.max), 25
Niter.max-methods(Niter.max), 25
Niter.max<-, 25
Niter.max<-, crp.CSFP, numeric-method
   (Niter.max<-, 25)
Niter.max<--methods (Niter.max<-, 25)
   NS, 25
   NS, crp.CSFP-method (NS), 25
   NS-methods (NS), 25
nu, 25
   nu, crp.CSFP-method (nu), 25
   nu-methods (nu), 25
path.in, 26
   path.in, crp.CSFP-method (path.in), 26
   path.in-methods (path.in), 26
   path.in<-, 26
   path.in<--, crp.CSFP, character-method
   (path.in<--), 26
   path.in<--methods (path.in<--), 26
path.out, 26
   path.out, crp.CSFP-method (path.out), 26
   path.out-methods (path.out), 26
   path.out<-, 26
   path.out<--, crp.CSFP, character-method
   (path.out<--), 26
   path.out<--methods (path.out<--), 26
PD, 26
   PD, crp.CSFP-method (PD), 26
   PD-methods (PD), 26
PD.crp, 27
   PD.crp, crp.CSFP-method (PD.crp), 27
PD.crp-methods (PD.crp), 27
pd_sector_var, 27
PDF, 27
   PDF, crp.CSFP-method (PDF), 27
   PDF-methods (PDF), 27
PL, 27
   PL, crp.CSFP-method (PL), 27
   PL-methods (PL), 27
PL.crp, 27
   PL.crp, crp.CSFP-method (PL.crp), 27
   PL.crp-methods (PL.crp), 27
plausi, 10, 28
   plausi, crp.CSFP-method (plausi), 28
   plausi-methods (plausi), 28
plot, 28
   plot, crp.CSFP-method (plot), 28
   plot-methods (plot), 28
PLOT.PDF, 28
   PLOT.PDF, crp.CSFP-method (PLOT.PDF), 28
   PLOT.PDF-methods (PLOT.PDF), 28
PLOT.PDF<-, 28
   PLOT.PDF<-, crp.CSFP, logical-method
   (PLOT.PDF<--), 28
   PLOT.PDF<--methods (PLOT.PDF<--), 28
PLOT.range.x, 29
   PLOT.range.x, crp.CSFP-method
   (PLOT.range.x), 29
   PLOT.range.x-methods (PLOT.range.x), 29
   PLOT.range.x<-, 29
   PLOT.range.x<--methods
   (PLOT.range.x<--), 29
PLOT.range.y, 29
   PLOT.range.y, crp.CSFP-method
   (PLOT.range.y), 29
   PLOT.range.y-methods (PLOT.range.y), 29
   PLOT.range.y<-, 29
   PLOT.range.y<--methods
   (PLOT.range.y<--), 29
PLOT.scale, 29
   PLOT.scale, crp.CSFP-method
   (PLOT.scale), 29
   PLOT.scale-methods (PLOT.scale), 29
   PLOT.scale<-, 30
   PLOT.scale<-, crp.CSFP, numeric-method
   (PLOT.scale<--), 30
   PLOT.scale<--methods (PLOT.scale<--), 30
port.name, 30
   port.name, crp.CSFP-method (port.name), 30
   port.name-methods (port.name), 30
   port.name<-, 30
   port.name<--, crp.CSFP, character-method
   (port.name<--), 30
   port.name<--methods (port.name<--), 30
portfolio, 30
rating, 30
   rating, crp.CSFP-method (rating), 30
   rating-methods (rating), 30
rating.PD, 31
   rating.PD, crp.CSFP-method (rating.PD), 31
   rating.PD-methods (rating.PD), 31
rating.PD<-, 31
INDEX

rating.PD<->, crp.CSFP, numeric-method (rating.PD<->), 31
rating.PD<--methods (rating.PD<->), 31
rating.scale.name, 31
rating.scale.name, crp.CSFP-method (rating.scale.name), 31
rating.scale.name-methods (rating.scale.name), 31
rating.scale.name<->, crp.CSFP, character-method (rating.scale.name<->), 31
rating.scale.name<->-methods (rating.scale.name<->), 31
rating.SD, 31
rating.SD, crp.CSFP-method (rating.SD), 31
rating.SD-methods (rating.SD), 31
rating.SD<-, 32
rating.SD<-, crp.CSFP, numeric-method (rating.SD<->), 32
rating.SD<--methods (rating.SD<--), 32
ratings<-, 32
ratings<-, crp.CSFP, numeric-method (ratings<->), 32
ratings<--methods (ratings<--), 32
rating_pd, 32
rc.sd, 10, 17, 32, 33
rc.sd, crp.CSFP-method (rc.sd), 32
rc.sd-methods (rc.sd), 32
rc.vares, 10, 17, 22, 32, 33
rc.vares, crp.CSFP-method (rc.vares), 33
rc.vares-methods (rc.vares), 33
read, 10, 33
read, crp.CSFP-method (read), 33
read-methods (read), 33
save.memory, 33
save.memory, crp.CSFP-method (save.memory), 33
save.memory-methods (save.memory), 33
save.memory<-, 33
save.memory<-, crp.CSFP, logical-method (save.memory<->), 33
save.memory<--methods (save.memory<--), 33
SD, 34
SD, crp.CSFP-method (SD), 34
SD-methods (SD), 34
SD.cont, 34
SD.cont, crp.CSFP-method (SD.cont), 34
SD.cont-methods (SD.cont), 34
SD.crt, 34
SD.crt, crp.CSFP-method (SD.crt), 34
SD.crt-methods (SD.crt), 34
sec.var, 34
sec.var, crp.CSFP-method (sec.var), 34
sec.var-methods (sec.var), 34
sec.var.est, 34
sec.var.est, crp.CSFP-method (sec.var.est), 34
sec.var.est-methods (sec.var.est), 34
sec.var.est<-, 35
sec.var.est<-, crp.CSFP, numeric-method (sec.var.est<->), 35
sec.var.est<--methods (sec.var.est<--), 35
sec.var.name, 35
sec.var.name, crp.CSFP-method (sec.var.name), 35
sec.var.name-methods (sec.var.name), 35
sec.var.name<-, 35
sec.var.name<-, crp.CSFP, character-method (sec.var.name<->), 35
sec.var.name<--methods (sec.var.name<--), 35
sec.var.name<--methods (sec.var.name<--), 35
sec.var<-, 35
sec.var<-, crp.CSFP, numeric-method (sec.var<->), 35
sec.var<--methods (sec.var<--), 35
set.ALBRA<-(alpha<->), 6
set.ALBRA<-, crp.CSFP, numeric-method (alpha<->), 6
set.CALC.RISK.CONT<-(calc.rc<->), 7
set.CALC.RISK.CONT<-, crp.CSFP, logical-method (calc.rc<->), 7
set.changes, 35
set.changes, character, crp.CSFP-method (set.changes), 35
set.changes-methods, character, crp.CSFP-method (set.changes), 35
set.file.format<-(file.format<->), 18
set.file.format<-, crp.CSFP, character-method (file.format<->), 18
set.LOSS.UNIT<-(loss.unit<->), 23
set.LOSS.UNIT<-, crp.CSFP, numeric-method (loss.unit<->), 23
set.NAME<-(name<->), 24
INDEX

set.NAME<-.crp.CSFP, character-method (name<->), 24
set.NITER.MAX<-(Niter.max<->), 25
set.NITER.MAX<-.crp.CSFP, numeric-method (Niter.max<->), 25
set.PATH.IN<-(path.in<->), 26
set.PATH.IN<-.crp.CSFP, character-method (path.in<->), 26
set.PATH.OUT<-(path.out<->), 26
set.PATH.OUT<-.crp.CSFP, character-method (path.out<->), 26
set.PDVAR.NAME<-(sec.var.name<->), 35
set.PDVAR.NAME<-.crp.CSFP, character-method (sec.var.name<->), 35
set.PLOT.PDF<-(PLOT.pdf<->), 28
set.PLOT.PDF<-.crp.CSFP, logical-method (PLOT.pdf<->), 28
set.PLOT.RANGE.X<-(PLOT.range.x<->), 29
set.PLOT.RANGE.X<-.crp.CSFP, numeric-method (PLOT.range.x<->), 29
set.PLOT.RANGE.Y<-(PLOT.range.y<->), 29
set.PLOT.RANGE.Y<-.crp.CSFP, numeric-method (PLOT.range.y<->), 29
set.PLOT.SCALE<-(PLOT.scale<->), 30
set.PLOT.SCALE<-.crp.CSFP, numeric-method (PLOT.scale<->), 30
set.PORT.NAME<-(port.name<->), 30
set.PORT.NAME<-.crp.CSFP, character-method (port.name<->), 30
set.RISK.NAME<-(rating.scale.name<->), 31
set.RISK.NAME<-.crp.CSFP, character-method (rating.scale.name<->), 31
set.save.memory<-(save.memory<->), 33
set.save.memory<-.crp.CSFP, logical-method (save.memory<->), 33
set.SEC.VAR.EST<-(sec.var.est<->), 35
set.SEC.VAR.EST<-.crp.CSFP, numeric-method (sec.var.est<->), 35
show, 36, 36
show<-.crp.CSFP-method (show), 36
show-methods (show), 36
sigma.k<-(sigma.k), 36
sigma.k<-.crp.CSFP-method (sigma.k), 36
sigma.k-methods (sigma.k), 36
sigma.sqr.div<-(sigma.sqr.div), 36
sigma.sqr.div<-.crp.CSFP-method (sigma.sqr.div), 36
sigma.sqr.div-methods (sigma.sqr.div), 36
sigma.sqr.syst, 36
sigma.sqr.syst<-.crp.CSFP-method (sigma.sqr.syst), 36
sigma.sqr.syst-methods (sigma.sqr.syst), 36
summary, 37, 38
summary<-.crp.CSFP-method (summary), 37
summary-methods (summary), 37
VaR, 37
VaR<-.crp.CSFP-method (VaR), 37
VaR-methods (VaR), 37
VaR.cont, 37
VaR.cont<-.crp.CSFP-method (VaR.cont), 37
VaR.cont-methods (VaR.cont), 37
VaR.pos, 37
VaR.pos<-.crp.CSFP-method (VaR.pos), 37
VaR.pos-methods (VaR.pos), 37
W, 37
W<-.crp.CSFP-method (W), 37
W-methods (W), 37
W<-, 38
W<-.crp.CSFP, numeric-method (W<-), 38
W<-<methods (W<-), 38
write.summary, 37, 38
write.summary<-.crp.CSFP-method (write.summary), 38
write.summary-methods (write.summary), 38