Package ‘redlistr’

October 14, 2022

Title Tools for the IUCN Red List of Ecosystems and Species

Version 1.0.3

Description A toolbox created by members of the International Union for Conservation of Nature (IUCN) Red List of Ecosytems Committee for Scientific Standards. Primarily, it is a set of tools suitable for calculating the metrics required for making assessments of species and ecosystems against the IUCN Red List of Threatened Species and the IUCN Red List of Ecosystems categories and criteria. See the IUCN website for detailed guidelines, the criteria, publications and other information.

URL https://github.com/red-list-ecosystem/redlistr

BugReports https://github.com/red-list-ecosystem/redlistr/issues

Imports grDevices, methods, plyr, rgeos

Depends R (>= 3.3.0), raster (>= 2.5-8), sp (>= 1.2-4)

Suggests devtools, knitr, rgdal, rmarkdown, testthat

License GPL (>= 3) | file LICENSE

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Calvin Lee [cre, aut] (<https://orcid.org/0000-0001-8277-8614>), Nicholas Murray [aut] (<https://orcid.org/0000-0002-4008-3053>)

Maintainer Calvin Lee <calvinkflee@gmail.com>

Repository CRAN

Date/Publication 2019-07-11 13:42:58 UTC

R topics documented:

createGrid ................................................................. 2
extrapolateEstimate .................................................. 3
futureAreaEstimate ................................................... 4
createGrid

Create empty Area of Occupancy (AOO) Grid.

Description

createGrid produces empty grid which can be used as the basis to help compute AOO.

Usage

createGrid(input.data, grid.size)

Arguments

input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.

grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)

Value

A regular grid raster with extent input.data and grid size grid.size. Each grid square has a unique identification number.

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>
extrapolateEstimate

References


See Also

Other AOO functions: getAOOSilent, getAOO, makeAOOGrid

extrapolateEstimate Extrapolate Estimate

Description

extrapolateEstimate uses rates of decline from getDeclineStats to extrapolate estimates to a given time

Usage

extrapolateEstimate(A.t1, year.t1, nYears, ARD = NA, PRD = NA, ARC = NA)

Arguments

A.t1 Area at time t1
year.t1 Year of time t1
nYears Number of years since t1 for prediction. Use negative values for backcasting
ARD Absolute rate of decline
PRD Proportional rate of decline
ARC Annual rate of change

Value

A dataframe with the forecast year, and a combination of:

- Values as extrapolated with absolute rate of decline (ARD)
- Values as extrapolated with proportional rate of decline (PRD)
- Values as extrapolated with annual rate of change (ARC)

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>
References

See Also
Other change_functions: futureAreaEstimate, sequentialExtrapolate

Examples
```r
a.r1 <- 23.55
a.r2 <- 15.79
decline.stats <- getDeclineStats(a.r1, a.r2, year.t1 = 1990, year.t2 = 2012, methods = 'Var'

a.2040.PRD <- extrapolateEstimate(a.r1, a.r2, year.t1 = 1990, nYears = 50,
PRD = decline.stats$PRD)
```

futureAreaEstimate future Area Estimate

Description
Future Area Estimate is now deprecated, please use extrapolateEstimate instead

Usage
futureAreaEstimate(A.t1, year.t1, nYears, ARD = NA, PRD = NA,
Arc = NA)

Arguments
- `A.t1`: Area at time t1
- `year.t1`: Year of time t1
- `nYears`: Number of years since t1 for area prediction
- `ARD`: Absolute rate of decline
- `PRD`: Proportional rate of decline
- `ARC`: Annual rate of change

Value
A dataframe with the forecast year, and a combination of:
- Future area as estimated with absolute rate of decline (ARD)
- Future area as estimated with proportional rate of decline (PRD)
- Future area as estimated with annual rate of change (ARC)
getAOO

Author(s)
Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

References

See Also
Other change_functions: extrapolateEstimate, sequentialExtrapolate

getAOO  
Compute Area of Occupancy (AOO)

Description
getAOO determines the number of area of occupancy (AOO) grid cells occupied by a species or ecosystem. It includes capability for specifying whether at least one percent of the grid cell needs to be occupied before it is counted in the AOO. This functionality is important for assessing the IUCN Red List of Ecosystems Criteria B.

Usage
getAOO(input.data, grid.size, min.percent.rule = FALSE, percent = 1)

Arguments
input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)
min.percent.rule Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.
percent Numeric. The minimum percent to be applied as a threshold for the min.percent.rule

Value
The number of grid cells occupied by the ecosystem or species

Author(s)
Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>
References

See Also
Other AOO functions: createGrid, getAOSilent, makeAOG

Examples
```r
crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
AOO <- getAOS(r1, 10000, min.percent.rule = TRUE, percent = 1)
```

**getAOSilent**

Alternate function for getting AOO (with custom grid)

**Description**

getaOS is identical to getAOS, but allows the custom input of the grid parameter. Used for gridUncertainty.

**Usage**

getaOS(input.data, grid, min.percent.rule = FALSE, percent = 1)

**Arguments**

- `input.data`: Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
- `grid`: Custom grid to be used to calculate AOO. Usually the output of gridUncertainty.
- `min.percent.rule`: Logical. If TRUE one percent of the grid cell must be occupied before it is counted in the AOO.
- `percent`: Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.

**Value**

Value. The AOO calculated with the input distribution and grid.

**Author(s)**

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

**See Also**

Other AOO functions: createGrid, getA0, makeAOG
getArea

Calculates the Area of a Raster.

Description

gArea reports the area of a RasterLayer object using the pixel counting method, or the area of a SpatialPolygons object using rgeos::gArea

Usage

gArea(x, value.to.count)

Arguments

x
Either a RasterLayer or SpatialPolygons object. For a RasterLayer, no data value should be NA

value.to.count
Optional. Value of the cells in a RasterLayer to be counted

Value

The total area of the cells of interest in km^2

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

See Also

Other Change functions: getAreaLoss, getDeclineStats

Examples

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
a.r1 <- getArea(r1) # area of all non-NA cells in r1
getAreaE0O

Calculates area of the created EOO polygon.

Description
getAreaE0O calculates the area of the EOO polygon generated from makeE0O the provided data.

Usage
getAreaE0O(E0O.polygon)

Arguments
E0O.polygon An object of class SpatialPolygons, usually the output from makeE0O.

Value
The area of the E0O.polygon in km2

Author(s)
Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

See Also
Other EOO functions: makeE0O

Examples
```r
crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
E0O.polygon <- makeE0O(r1)
E0O.area <- getAreaE0O(E0O.polygon)
```

getAreaLoss

Area change between two inputs in km2

Description
getAreaLoss reports the difference in area between two inputs. These can be RasterLayers, SpatialPolygons, or numbers. Any combinations of these inputs are valid. If using number as input, ensure it is measured in km2

Usage
getAreaLoss(x, y)
getDeclineStats

Arguments

- **x**: RasterLayer or SpatialPolygons object of distribution or Numeric representing area in km²
- **y**: RasterLayer or SpatialPolygons object of distribution or Numeric representing area in km²

**Value**

Returns the difference in area of the two inputs in km²

**Author(s)**

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

**See Also**

Other Change functions: *getArea, getDeclineStats*

**Examples**

```r
# Example code

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
r2 <- raster(ifelse((volcano<145), NA, 1), crs = crs.UTM55S)
extent(r2) <- extent(0, 6100, 0, 8700)
a.dif <- getAreaLoss(r1, r2) # distribution rasters
```

---

**Description**

getDeclineStats calculates the Proportional Rate of Decline (PRD), Absolute Rate of Decline (ARD) and Annual Rate of Change (ARC), given two areas at two points in time. Also provides the total area difference. Inputs are usually the results from getArea.

**Usage**

```r
getDeclineStats(A.t1, A.t2, year.t1, year.t2, methods)
```

**Arguments**

- **A.t1**: Area at time t1
- **A.t2**: Area at time t2
- **year.t1**: Year of time t1
- **year.t2**: Year of time t2
- **methods**: Method(s) used to calculate rate of decline. Either 'PRD', 'ARD', and/or 'ARC'. See vignette to see a more detailed explanation for each of them.
gridUncertainty

Value

A dataframe with absolute differences between the two inputs, and a selection of:

- Proportional Rate of Decline (PRD)
- Absolute Rate of Decline (ARD)
- Annual Rate of Change (ARC)

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

References


See Also

Other Change functions: getAreaLoss, getArea

Examples

```r
a.r1 <- 23.55
a.r2 <- 15.79
decline.stats <- getDeclineStats(a.r1, a.r2, year.t1 = 1990, year.t2 = 2012,
                                 methods = c('ARD', 'ARC'))
```

Description

gridUncertainty determines the number of area of occupancy (AOO) grid cells occupied by a species or ecosystem systematically. It will only stop when the AOO calculated does not improve (decrease) after a set number of split scenarios.

Usage

```r
gridUncertainty(input.data, grid.size, n.AOO.improvement,
                 min.percent.rule = FALSE, percent = 1)
```
gridUncertaintyBase

Arguments

input.data: Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.

grid.size: A number specifying the width of the desired grid square (in same units as your coordinate reference system).

n.AOO.improvement: Specifies the minimum number of rounds the calculated AOO is not improved before stopping the function.

min.percent.rule: Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.

percent: Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.

Value

A list containing the following:

- Data frame of results showing the minimum AOO calculated for each shift scenario
- Single SpatialPolygonsDataFrame containing the AOO grid which would produce the minimum AOO calculated

Author(s)

Calvin Lee <calvinkflee@gmail.com>

See Also


Examples

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse(volcano<130, NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
x <- gridUncertainty(r1, grid.size = 10000, n.AOO.improvement = 5, min.percent.rule = FALSE, percent = 1)

Description

gridUncertaintyBase helps determine the minimum number of area of occupancy (AOO) grid cells occupied by a species or ecosystem. It varies the location of the AOO grid by shifting in systematically in both x- and y-axes, adding a small amount of random movement (five percent of the grid.size) at each point. It then returns summary statistics for the range of AOOs calculated, and the RasterLayer(s) containing the grids with the minimum AOO. It is the base function which is used by gridUncertainty, gridUncertaintySimulation, and gridUncertaintyRestricted.
gridUncertaintyBase

Usage

gridUncertaintyBase(input.data, grid.size, splits,
min.percent.rule = FALSE, percent = 1, restriction = FALSE,
min.grids.shift)

Arguments

input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.

grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)

splits Specifies the number of ways to split the grid in ONE axis.

min.percent.rule Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.

percent Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.

restriction Logical. If TRUE, allows user to specify areas to focus where grid search is done. Used in gridUncertaintyRestricted.

min.grids.shift Dataframe object with two columns (x.shift and y.shift) specifying the coordinates to restrict the AOO grid placement.

Value

List containing the following:

- Vector of length split*split of calculated AOO for each shifted grid
- Data frame of summary statistics for the results create the AOO grid(s) which return the smallest AOO
- Data frame of the shift(s) required to create the AOO grid(s) with the smallest AOO

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

See Also

createGrid getAOOSilent

gridUncertaintyRandom

Function to compute AOO with grid uncertainty randomly with stop rule

Description

gridUncertaintyRandom helps determine the minimum number of area of occupancy (AOO) grid cells occupied by a species or ecosystem. It varies the location of the AOO grid by shifting in randomly in both x- and y- axes, returning summary statistics for the range of AOOs calculated, and the RasterLayer(s) containing the grids with the minimum AOO. It automatically stops when the AOO no longer improves after a specified number of rounds.

Usage

gridUncertaintyRandom(input.data, grid.size, n.AOO.improvement, min.percent.rule = FALSE, percent = 1, max.n.rounds = 1000)

Arguments

- **input.data**: Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
- **grid.size**: A number specifying the width of the desired grid square (in same units as your coordinate reference system)
- **n.AOO.improvement**: Specifies the minimum number of rounds the calculated AOO is not improved before stopping the function.
- **min.percent.rule**: Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.
- **percent**: Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.
- **max.n.rounds**: Specifies the maximum number of rounds to calculate AOOs. Generally unused except to limit computation time.

Value

List containing the following:

- Data frame of summary statistics for the results
- Data frame showing the distance shifted in x and y directions used to create the AOO grid(s) and their associated AOOs
- List of RasterLayer(s) containing the AOO grid(s) which return the smallest AOO

Author(s)

Calvin Lee <calvinkflee@gmail.com>. Nicholas Murray <murr.nick@gmail.com>
gridUncertaintyRandomManual

See Also

createGrid getAOOSilent


Examples

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
x <- gridUncertaintyRandom(r1, grid.size = 10000, n.AOO.improvement = 50,
min.percent.rule = TRUE, percent = 1)

gridUncertaintyRandomManual

Manual function to compute AOO with grid uncertainty randomly

Description

gridUncertaintyRandomManual helps determine the minimum number of area of occupancy (AOO) grid cells occupied by a species or ecosystem. It varies the location of the AOO grid by shifting in randomly in both x- and y- axes, returning summary statistics for the range of AOOs calculated, and the RasterLayer(s) containing the grids with the minimum AOO. Requires manual input for the number of simulations to perform.

Usage

gridUncertaintyRandomManual(input.data, grid.size, n.sim = 10,
min.percent.rule = FALSE, percent = 1)

Arguments

input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)
n.sim Specifies the number of random grids to be created and tested.
min.percent.rule Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.
percent Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.
gridUncertaintyRestricted

Value
List containing the following:

- Data frame of summary statistics for the results
- Data frame showing the distance shifted in x and y directions used to create the AOO grid(s) and their associated AOOs
- List of RasterLayer(s) containing the AOO grid(s) which return the smallest AOO

Author(s)
Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

See Also
createGrid getAOOSilent
Other gridUncertainty functions: gridUncertaintyBase, gridUncertaintyRandom, gridUncertaintyRestricted, gridUncertaintySimulation, gridUncertainty

Description
gridUncertaintyRestricted determines the number of area of occupancy (AOO) grid cells occupied by a species or ecosystem systematically. It will only stop when the AOO calculated does not improve (decrease) after a set number of split scenarios. The number of grids within each split is restricted to only include those which are already found nearby to ones already with the minimum AOO.

Usage
gridUncertaintyRestricted(input.data, grid.size, n.AOO.improvement, min.percent.rule = FALSE, percent = 1)

Arguments
input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)
n.AOO.improvement Specifies the minimum number of rounds the calculated AOO is not improved before stopping the function.
gridUncertaintySimulation

min.percent.rule
Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.

percent
Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.

Value
A list containing the following:

• Data frame of results showing the minimum AOO calculated for each shift scenario
• Single SpatialPolygonsDataFrame containing the AOO grid which would produce the minimum AOO calculated

Author(s)
Calvin Lee <calvinkflee@gmail.com>

See Also

Description
gridUncertaintySimulation returns the maximum and minimum number of area of occupancy (AOO) grid cells occupied by a species or ecosystem in incremental splits using gridUncertaintyBase.

Usage
gridUncertaintySimulation(input.data, grid.size, simulations, min.percent.rule = FALSE, percent = 1)

Arguments
input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
grid.size A number specifying the width of the desired grid square (in same units as your coordinate reference system)
simulations Specifies the maximum number of splits to be performed on the generated grid
min.percent.rule Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.
percent Numeric. The minimum percent to be applied as a threshold for the min.percent.rule.
**makeAOOGGrid**

**Value**
Data frame of results showing the minimum and maximum AOO calculated for each grid shift scenario.

**Author(s)**
Calvin Lee <calvinkflee@gmail.com>

**See Also**

**Description**
makeAOOGGrid creates grids for species presence based on the presented raster object. It includes capability for specifying whether a minimum percent of the grid cell needs to be occupied before it is counted in the AOO. This functionality is important for assessing the IUCN Red List of Ecosystems Criteria B.

**Usage**

```r
makeAOOGGrid(input.data, grid.size, min.percent.rule = FALSE, percent = 1)
```

**Arguments**

- **input.data**: Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.
- **grid.size**: A number specifying the width of the desired grid square (in same units as your coordinate reference system)
- **min.percent.rule**: Logical. If TRUE, a minimum area threshold must be passed before a grid is counted as an AOO grid.
- **percent**: Numeric. The minimum percent to be applied as a threshold for the min.percent.rule

**Value**
A shapefile of grid cells occupied by an ecosystem or species

**Author(s)**
Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>
makeEOO

Creates Extent of occurrence (EOO) Polygon

Description

makeEOO creates a minimum convex polygon enclosing all occurrences of the provided data

Usage

makeEOO(input.data)

Arguments

input.data Object of an ecosystem or species distribution. Accepts either raster or spatial points formats. Please use a CRS with units measured in metres.

Value

An object of class SpatialPolygons representing the EOO of input.data. Also inherits its CRS.

Author(s)

Nicholas Murray <murr.nick@gmail.com>, Calvin Lee <calvinkflee@gmail.com>

References


See Also

Other EOO functions: getAreaEOO

Examples

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
AOO_grid <- makeAOOGrid(r1, 10000, min.percent.rule = TRUE, percent = 1)

References

sequentialExtrapolate

**Examples**

crs.UTM55S <- '+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs'
r1 <- raster(ifelse((volcano<130), NA, 1), crs = crs.UTM55S)
extent(r1) <- extent(0, 6100, 0, 8700)
EOO.polygon <- makeEOO(r1)

**sequentialExtrapolate**  Sequential extrapolation estimate

**Description**

sequentialExtrapolate uses rates of decline from getDeclineStats and generates a sequence of estimates at regular time-steps. Useful for generating a sequence for plotting graphs.

**Usage**

sequentialExtrapolate(A.t1, year.t1, nYears, ARD = NA, PRD = NA,
ARC = NA)

**Arguments**

- **A.t1**  Area at time t1
- **year.t1**  Year of time t1
- **nYears**  Number of years since t1 for prediction. Use negative values for backcasting
- **ARD**  Absolute rate of decline
- **PRD**  Proportional rate of decline
- **ARC**  Annual rate of change

**Value**

A dataframe with the forecast year, and a combination of:

- Sequence of values as extrapolated with absolute rate of decline (ARD)
- Sequence of values as extrapolated with proportional rate of decline (PRD)
- Sequence of values as extrapolated with annual rate of change (ARC)

**Author(s)**

Calvin Lee <calvinkflee@gmail.com>

**References**

See Also

Other change_functions: `extrapolateEstimate, futureAreaEstimate`

Examples

```r
a.r1 <- 23.55
a.r2 <- 15.79
decline.stats <- getDeclineStats(a.r1, a.r2, year.t1 = 1990, year.t2 = 2012,
                                methods = "PRD")
a.2040.PRD.seq <- sequentialExtrapolate(a.r1, a.r2, year.t1 = 1990, nYears = 50,
                                PRD = decline.stats$PRD)
```
Index

* AOO functions
  createGrid, 2
  getAOO, 5
  getAOOSilent, 6
  makeAOOGrid, 17

* Change functions
  getArea, 7
  getAreaLoss, 8
  getDeclineStats, 9

* EOO functions
  getAreaEOO, 8
  makeEOO, 18

* change_functions
  extrapolateEstimate, 3
  futureAreaEstimate, 4
  sequentialExtrapolate, 19

* gridUncertainty functions
  gridUncertainty, 10
  gridUncertaintyBase, 11
  gridUncertaintyRandom, 13
  gridUncertaintyRandomManual, 14
  gridUncertaintyRestricted, 15
  gridUncertaintySimulation, 16

createGrid, 2, 6, 12, 14, 15, 18
extrapolateEstimate, 3, 5, 20
futureAreaEstimate, 4, 4, 20
getAOO, 3, 5, 6, 18
getAOOSilent, 3, 6, 6, 12, 14, 15, 18
getArea, 7, 9, 10
getAreaEOO, 8, 18
getAreaLoss, 7, 8, 10
getDeclineStats, 7, 9, 9
gridUncertainty, 10, 12, 14–17
gridUncertaintyBase, 11, 11, 14–17
gridUncertaintyRandom, 11, 12, 13, 15–17
gridUncertaintyRandomManual, 11, 12, 14, 14, 16, 17

gridUncertaintyRestricted, 11, 12, 14, 15, 16, 17
gridUncertaintySimulation, 11, 12, 14–16, 16

makeAOOGrid, 3, 6, 17
makeEOO, 8, 18
sequentialExtrapolate, 4, 5, 19