Package 'smartR'

December 1, 2018

Type Package

Title Spatial Management and Assessment of Demersal Resources for Trawl Fisheries

Version 0.62.0

Date 2018-11-20

Author Lorenzo D'Andrea, Tommaso Russo, Antonio Parisi, Stefano Cataudella

Maintainer Lorenzo D'Andrea <dandrea.lorenz@gmail.com>

Description A tool for assessing bio-economic feedback in different management scenarios. 'smartR' (Spatial Management and Assessment of demersal Resources for Trawl fisheries) combines information from different tasks gathered within the European Data Collection Framework for the fishery sector. The 'smartR' package implements the SMART model (Russo et al., 2014 <doi:10.1371/journal.pone.0086222>), through the object-oriented programming paradigm, and within this package it is possible to achieve the complete set of analyses required by the SMART approach: from the editing and formatting of the raw data; the construction and maintenance of coherent datasets; the numerical and visual inspection of the generated metadata; to the final simulation of management scenarios and the forecast of their effects. The interaction between the user and the application could take place through invocation of methods via the command line or could be entirely committed to the graphical user interfaces (GUI).

License GPL (>= 2)

Depends R6, rjags, maptools, mapdata

Imports PBSmapping, gWidgets2, gWidgets2RGtk2, gstat, lattice, plyr, sp, maps, clues, cluster, spdep, marmap, shape, jpeg, grid, ggplot2, sqldf, gsubfn, chron, reshape2, foreign, ROCR, caret, nnls, mtsdi, gridExtra, rgeos, scales, grDevices, ggthemes, vegan, rpart, ggrepel, rgdal, RColorBrewer, jsonlite, igraph, mapproj

LazyData true

RoxygenNote 6.1.0

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-30 23:40:03 UTC

R topics documented:

`FisheryBySpecie` *FisheryBySpecie*

Description

The `FisheryBySpecie` class implements the class of SMART to handle species samplings.

Format

`R6Class` object.

Value

Object of `R6Class` with attributes and methods for the fishery data.

Fields

`specie` Name of the specie.

`year` Years of the time serie.

`rawLFD` data.frame, raw length frequency distribution.

`abuAvg` data.frame, average abundances by depth' stratum.

`meditsIndex` data.frame, medits index by depth' stratum.

`lengClas` numeric, length classes.

`nCoho` numeric, number of cohorts.

`spreDist` list of DF, lfd by sex.

`sprePlot` plots of LFD statistics.

`spreSpat` list of DF, spatial distribution by sex.

`sampMcmc` list, mcmc output chains.

`groMixout` list of DF, aged individuals by sex.

`groPars` list of DF, growth parameters by sex.

`LWpar` list of DF, length/weight parameters by sex.

#' @section Methods:

`initialize(sing_spe)` Automatic initialization made by the SmartProject class

`setRawData(raw_data)` This method is used load the initial raw dataset

`setYears()` This method is used to store the years in the provided time-serie

`setSpecie()` This method is used to store the name of the specie of the initial raw data

`setLClass()` This method is used to store the unique length values of the sampled specie

`setDepth(bathyMatrix)` This method is used to assign the depth value corresponding to each sampling location

`setStratum(vecStrata)` This method is used to set the depth strata of each sampling location

`setIndSpe()` This method is used to aggregate the abundance data into the medits index

`setAbuAvg()` This method is used to standardize the spatial abundances by depth strata

`setNCoho(num_coh)` This method is used to setup the number of cohorts for the ageing module

`setLWpar(alphaVal, betaVal, sex)` This method is used to store the alpha and beta values for the length/weight relationship

`setWeight(sexVal = "Female")` This method is used to compute the fish weight given their length and the LWrelationship

`setSpreDistSing()` This method is used to spread the aggregated LFD abundances into single individuals

`setSprePlot(sampSex)` This method is used to setup the plots of the LFD statistics

`setSpatDistSing()` This method is used to setup the spatial distribution of the single specimens

`setSpatPlot(sampSex)` This method is used to store the spatial plots of the population

`getMCsamps(numSamp, numAdap, numIter, sexDrop, curveSel)` This method is used to get a sample of the population to feed the mcmc module

`getGrowPar(sexDrop)` This method is used to extract the growth parameters from the mcmc results

`getMCage(sexDrop)` This method is used to assign an age to each fish

`setMCplot(sexDrop, selCurve)` This method is used to setup the plot of the mcmc results

`calcMixDate(nAdap, nSamp, nIter, sexDrop, curveSel)` This method is used to estimate the growth parameters of a population

`ggplotMcmcOut(selCompo, selSex)` This method is used to output the stored plots of mcmc results

FishFleet *FishFleet*

Description

The FishFleet class implements the class of SMART to manage fleet data.

Format

R6Class object.

Value

Object of R6Class with attributes and methods for the fishery data.

Fields

rawRegister data.frame, raw fleet register data.

vmsRegister data.frame, raw fleet register data for vms vessels only.

rawEffort list of DF, raw effort data.

dayEffoMatr list of DF, daily aggregated effort data.

prodMatr list of DF, production data.

effoProd list of DF, merged effort and production data.

effoProdMont list of DF, monthly aggregated effort and production data.

effoMont list of DF, monthly aggregated effort data.

effoProdAll data.frame, monthly aggregated effort and production data.

effoAll data.frame, monthly aggregated effort data.

regHarbsUni data.frame, Harbours name, longitude, latitude and distance from the environment grid.

regHarbsBox data.frame, Harbours name, longitude, latitude, number of registered vessels and distance from the environment grid within the grid box.

rawProduction list of DF, raw production data.

rawEconomy data.frame, raw economic data.

registerIds character, vessel identification from fleet register.

predProd list of matrix, simulated production.

productionIds list of int, vessel ids with production data available.

prodSpec list of character, specie with production data.

specSett list of DF, logit parameter settings by specie.

specLogit list, logit results by specie.

effortIds list of int, vessel ids with effort data available.

idsEffoProd list of int, merged vessel ids with both effort and production data available.

`effoProdAllLoa` data.frame, monthly aggregated effort, production and loa data.

`effoAllLoa` data.frame, monthly aggregated effort and loa data.

`effortIndex` data.frame, effort index by vessel, year and month with loa data.

`daysAtSea` data.frame, days at sea index by vessel, year, month with loa and Kw data.

`prodIndex` data.frame, production index by vessel, year and month.

`resNNLS` list, lander results by specie.

`betaMeltYear` list of DF, melted yearly productivity by specie.

`prodMeltYear` list of DF, melted yearly production by specie.

`fishPoinPara` data.frame, fishing point parameters.

`ecoPrice` list of DF, price/size by specie.

`inSpatialReg` data.frame, input for spatial index regression.

`inEffortReg` data.frame, input for effort index regression.

`inProductionReg` data.frame, input for production index regression.

`outSpatialReg` list, output for spatial index regression.

`outEffortReg` list, output for effort index regression.

`outProductionReg` list, output for production index regression.

`plotSpatialReg` ggplot, spatial index regression results.

`plotEffortReg` ggplot, effort index regression results.

`plotProductionReg` ggplot, production index regression results.

Methods

`setVmsRegister()` This method is used to exclude the fleet register records of vessels without vms

`setRegHarbs()` This method is used to fetch the harbours coordinates

`setEcoPrice(sel_specie, price_df)` This method is used to set the price/size attribute by species

`saveFleetHarb(harb_path)` This method is used to export the rds with the harbours' coordinates

`loadFleetHarb(harb_path)` This method is used to import the rds with the harbours' coordinates

`loadFleetRegis(register_path)` This method is used to load the raw fleet register

`loadMatEffort(effort_path)` This method is used to import the raw effort matrix

`loadRawEconomy(economic_path)` This method is used to load the raw csv file with economic data

`setInSpatial()` This method is used to setup the input for the spatial regression

`setInEffort()` This method is used to setup the input for the effprt regression

`getRegSpatial()` This method is used to compute the spatial regression

`getRegEffort()` This method is used to compute the effort regression

`getRegProduction()` This method is used to compute the production regression

`getCostOutput()` This method is a wrapper function to get the economic regressions

`setCostPlot()` This method is used to setup the plot of the economic regression

`loadProduction(production_path)` This method is used to load the raw csv of the production data

`setFishPoinPara(speed_range, depth_range)` This method is used to setup the fishign points parameters

`setWeekMonthNum()` This method is used to assign the week and month num to the raw effort data

`setFishPoin()` This method is used to filter the fishing points

`plotFishPoinStat()` This method is used to show the basic statistics for the fishing points

`plotSpeedDepth(which_year, speed_range, depth_range)` This method is used to show the speed/depth profile

`setEffortIds()` This method is used to set the distinct vessel' ids in the effort dataset

`setProdSpec()` This method is used to set the distinct specie in the production dataset

`setBetaMeltYear(specie)` This method is used to set the melted yearly productivity by specie

`setProdMeltYear(specie)` This method is used to set the melted yearly production by specie

`plotTotProd(specie)` This method is used to plot the total production by specie

`plotNNLS(specie, thresR2)` This method is used to show the NNLS results

`setSpecSettItm(specie, thresh, brea, max_xlim)` This method is used to set the logit parameters by specie

`plotLogitROC(selSpecie)` This method is used to show the ROC of the logit results

`setSpecLogitConf(selSpecie, cutoff)` This method is used to set the confusion matrix of the logit results by specie

`setLogitTrain(selSpecie, train, cp_val, cv_val)` This method is used to setup the train dataset for the logit model

`setLogitTest(selSpecie, test)` This method is used to setup the test dataset for the logit model

`setLogitPred(selSpecie, test)` This method is used to compute the prediction for the logit model

`setLogitCut(selSpecie)` This method is used to tune the cutoff of the logit model

`setLogitRoc(selSpecie)` This method is used to set the ROC of the logit model

`setLogitConf(selSpecie, test)` This method is used to set the confusion matrix of the logit results

`setSpecLogit(selSpecie, selModel, cp, cv)` This method is a wrapper function to get the logit model

`getMatSpeLand(specie)` This method is used to get the input data for the logit model

`setEffoProdAll()` This method is used to combine the effort/production data from the yearly list into a single data.frame

`setEffoAll()` This method is used to combine the effort data from the yearly list into a single data.frame

`setEffoProdAllLoa()` This method is used to add the LOA data to the effort/production data

`setEffoAllLoa()` This method is used to add the LOA data to the effort data

`setProdIds()` This method is used to get the vessel ids with production data available

`setIdsEffoProd()` This method is used to get the vessel ids with both effort and production data available

`plotCountIDsEffoProd()` This method is used to set the plot of the basic statistics of the effort/production data

`plotCountIDsEffo()` This method is used to set the plot of the basic statistics of the effort data

`plotCountIDsProd()` This method is used to set the plot of the basic statistics of the production data

`setEffoProdMatr()` This method is used to merge the effort and production data

`setEffoProdMont()` This method is used to aggregate the effort/production data by month

`setEffoMont()` This method is used to aggregate the effort data by month

`setProdMatr()` This method is used to create the production matrix from the raw production data

`setDayEffoMatrGround(maxFG)` This method is used to assign the fishing grounds to the raw effort data

`readRegisterEU(reg_path)` This method is used to load the raw european fleet register

`cleanRegister()` This method is used to clean the raw data in the fleet register

`plotRegSum()` This method is used to plot the basic statistics for the fleet register data

`setRegIds()` This method is used to get the distinct vessels ids in the fleet register

---

`SampleMap` *SampleMap*

Description

The `SampleMap` class implements the class of SMART to control geographical data.

Format

`R6Class` object.

Value

Object of `R6Class` with attributes and methods for the Environmental data.

Fields

`gridPath` Stores the file path to the selected Environment grid shapefile.

`gridName` Stores the file name of the Environment grid shapefile.

`gridShp` Stores the SpatialPoligon object of the Environment grid shapefile.

`gridBbox` Stores the bounding box coordinates of the Environment grid shapefile.

`gridBboxExt` Stores the extended bounding box coordinates of the Environment grid shapefile.

`gridBboxSP` Stores the bounding box of the Environment grid as a SpatialPoligon.

`areaGrid` Stores the total area covered by the Environment grid.

areaStrata Stores the area covered by depth strata.

weightStrata Stores the area covered by depth strata relative to the total area.

harbDbf Stores coordinates and names of the harbours.

bioPath Stores the file path of the Substrate map.

bioName Stores the file name of the Substrate map.

bioShp Stores the SpatialPoligon object of the Substrate map.

bioDF Stores the data.frame representation of the Substrate map.

gridPolySet Stores the PolySet object of the Environment grid.

gridFortify Stores the fortified SpatialPoligon object of the Environment grid.

nCells Stores the number of cells in the Environment grid.

griCent Stores the coordinates of the cells' centroids.

gridBathy Stores the bathymetric matrix.

centDept Stores the depth of the cells' centroids.

clusInpu Stores the input data for the spatial clustering.

clusMat Stores the results of the spatial clustering.

indSil Stores the silhouette index of the spatial clustering result.

indCH Stores the Calinski-Harabasz index of the spatial clustering result.

cutFG Stores the number of cuts for the spatial clustering.

availData Stores the names of the variables for the spatial clustering.

rawInpu Stores the raw input for the spatial clustering.

cutResult Stores the summary data of the spatial clustering result.

cutResEffo Stores the average effort data from the spatial clustering result.

cutResShp Stores the SpatialPoligon object of the spatial clustering result.

cutResShpCent Stores the coordinates of the clusters' centroids.

cutResShpFort Stores the fortified SpatialPoligon object of the spatial clustering result.

fgWeigDist Stores the weighted distance between harbours and fishing grounds.

ggBioDF Stores the plot of the Substrate map.

ggDepth Stores the plot of the Bathymetric map.

ggDepthFGbox Stores the boxplot of the depth values of each fishing ground.

ggEffoFGbox Stores the boxplot of the effort values of each fishing ground.

ggEffoFGmap Stores the plot of the Effort map.

ggBioFGmat Stores the tilemap of the substrate values of each fishing ground.

ggCutFGmap Stores the plot of the Fishing ground configuration.

ggIchFGlin Stores the plot of the Calinski-Harabasz index.

ggSilFGlin Stores the plot of the silhouette index.

ggBetaFGmap Stores the plot of the Productivity map.

ggBetaFGbox Stores the boxplot of the Productivity values of each fishing ground.

ggProdFGmap Stores the plot of the Production map.

ggProdFGbox Stores the boxplot of the Production values of each fishing ground.

ggMapFgFishery Stores the plot of the Fishery data coordinates.

ggMapFgSurvey Stores the plot of the Survey data coordinates.

gooMap Stores the satellite view of the area of study.

gooMapPlot Stores the satellite plot of the area of study.

gooGrid Stores the plot of the Environment Grid.

gooBbox Stores the plot of the Bounding Box of the Environment Grid.

sampColScale Stores the color scale for the species plots.

plotRange Stores the plot ranges for the Environmental Grid.

Methods

setAreaGrid() This method is used to compute the total area covered by the environmental grid.

setAreaStrata(vectorStrata) This method is used to compute the area covered by each depth strata.

setWeightStrata() This method is used to compute the area covered by each depth strata relative to the total area of the grid.

loadHarbDbf(dbf_path) This method is used to load a dbf file of coordinates and harbours names.

set_ggMapFgSurvey(rawSampCoo) This method is used to setup the plot of the spatial distribution of survey data.

set_ggMapFgFishery(rawSampCoo) This method is used to setup the plot of the spatial distribution of fishery data.

createGridBbox() This method is used to setup the bounding box of the environment grid.

getGooMap() This method is used to retrieve the satellite view of the area of study.

setGooPlot() This method is used to setup the base plot of the area of study.

setPlotRange() This method is used to setup the ranges of the base plot.

setGooGrid() This method is used to setup the plot of the environment grid.

plotGooGrid() This method is used to plot the environment grid.

plotGooGridData(grid_data) This method is used to plot the environment grid.

setSampColScale(fac_col) This method is used to setup the color scale for the species' plots.

plotGooSpeSur(poi_data) This method is used to plot the spatial distribution of the survey data

plotGooSpeFis(poi_data) This method is used to plot the spatial distribution of the fishery data

setGooBbox() This method is used to setup the bounding box of the environment grid.

plotGooBbox() This method is used to plot the bounding box of the environment grid.

setGridPath(path2grid) This method is used to store the path to the grid file.

setGridName() This method is used to store the name of the grid file.

loadGridShp() This method is used to load the grid file.

setBioPath() This method is used to store the path of the seabed substrates file.

`setBioName()` This method is used to store the name of the seabed substrates file.

`loadBioShp()` This method is used to load the seabed substrates file.

`addBioShp(bio_path)` This method is used store the path and name of the seabed file and then load the SpatialPoligon object.

`loadBioDF(bio_path)` This method is used to load a Data.Frame of seabed substrates.

`plotBioDF()` This method is used to plot the map of substrates.

`setGgBioDF()` This method is used to setup the plot of substrates.

`ggplotBioDF()` This method is used to plot the map of substrates.

`createPolySet()` This method is used to store the PolySet object of the Environment grid.

`fortifyGridShp()` This method is used to fortify the SpatialPolygon od the Environment grid.

`setNumCell()` This method is used to setup the number of cells in the grid.

`setGridCenter()` This method is used to store the coordinates of cells centroids.

`getGridBath()` This method is used to retrieve the bathymetric matrix.

`saveGridBath(bathy_path)` This method is used to save the bathymetric matrix to file.

`loadGridBath(bathy_path)` This method is used to load the bathymetric matrix from file.

`getCentDept()` This method is used to assign the depth to each cell centroids.

`setGgDepth(isoLine)` This method is used to setup the bathymetric plot.

`ggplotGridBathy()` This method is used to plot the bathymetric map.

`plotSamMap(title, celCol)` This method is used to plot the Environment map.

`plotCoho(abbs)` This method is used to plot the spatial distribution of the species.

`setClusInpu(whiData, howData)` This method is used to setup the input for the spatial clustering

`calcFishGrou(numCuts, minsize, maxsize, modeska, skater_method, nei_queen)` This method is used to run the spatial clustering routine

`plotFishGrou(ind_clu)` This method is used to plot the fishing ground configuration

`setCutResult(ind_clu)` This method is used to choose a fishing ground configuration

`setDepthFGbox()` This method is used to setup the boxplot of depth by fishing ground

`setEffoFGbox()` This method is used to setup the boxplot of effort by fishing ground

`setEffoFGmap()` This method is used to setup the map of effort by fishing ground

`setBioFGmat()` This method is used to setup the tileplot of substrate by fishing ground

`setCutFGmap()` This method is used to plot the fishing ground map

`setIchFGlin(numCut)` This method is used to setup the plot of the Calinski-Harabasz index

`setSilFGlin(numCut)` This method is used to setup the plot of the Silhouette index

SmartProject *SmartProject Class*

Description

The SmartProject class implements the main class of **smartR** package.

Format

R6Class object.

Value

Object of R6Class with attributes and methods to fullfill a complete analisys with the SMART approach.

Fields

rawDataSurvey Stores the raw survey data after being populated by loadSurveyLFD() method.

yearInSurvey Stores the distinct years in the rawDataSurvey time-serie.

specieInSurvey Stores the distinct species in the rawDataSurvey time-serie.

surveyBySpecie Stores a list of SurveyBySpecie objects, one for each species in the time-series.

rawDataFishery Stores the raw fishery data as is in the provided csv file. The attribute is populated by loadFisheryLFD() method.

yearInFishery Stores the distinct years in the rawDataFishery time-serie.

specieInFishery Stores the distinct species in the rawDataFishery time-serie.

fisheryBySpecie Stores a list of FisheryBySpecie objects, one for each species in the time-series.

gooLstCoho Stores a list of plots of species cohorts spatial distribution .

sampMap Stores the environment object.

fleet Stores the FishFleet object.

simProd Stores the simulated pattern of production.

simEffo Stores the simulated pattern of effort.

simBanFG Stores a vector of fishable/banned fishing grounds.

simSpatialCost Stores the simulated pattern of spatial costs.

simEffortCost Stores the simulated pattern of effort costs.

simProdCost Stores the simulated pattern of production costs.

simTotalCost Stores the simulated pattern of total costs.

simRevenue Stores the simulated pattern of revenue by species and fishing ground.

simTotalRevenue Stores the simulated pattern of total revenues.

simCostRevenue Stores the simulated pattern of costs and revenues.

`simResPlot` Stores the plots with the simulation' results.

`outGmat` Stores the evolution of gains during the simulation.

`outOptimEffo` Stores the resulting pattern of effort.

`outWeiProp` Stores the annual proportion of fish by cohort and fishing ground.

`outWeiPropQ` Stores the seasonal proportion of fish by cohort and fishing ground.

Methods

`setCostInput()` This method is used to setup the required data for costs computation

`setInProduction()` This method is used to setup the required data for production costs computation

`setDaysAtSea()` This method is used to compute the number of Days at Sea of each vessel

`setEffortIndex()` This method is used to compute the value of the Effort Index

`setProductionIndex()` This method is used to compute the value of the Production Index

`getHarbFgDist()` This method is used to compute the weighted average distance of every fishing ground to each harbour

`setFgWeigDist()` This method is used as helper function to get the weighted average distance between fishing ground and harbours

`setRegHarbBox()` This method is used to compute the distance of each harbour to every fishing ground centroid

`loadSurveyLFD(csv_path)` This method is used to load the raw survey LFD data from a csv file

`loadFisheryLFD(csv_path)` This method is used to load the raw fishery LFD data from a csv file

`setYearSurvey()` This method is used to store the distinct year in the survey time-series

`setYearFishery()` TThis method is used to store the distinct year in the fishery time-series

`loadMap(map_path)` This method is used to load the Environmental Grid and initialize the Environment object

`createFleet()` This method is used to initialize the Fleet object

`setSpecieSurvey()` This method is used to store the distinct species in the survey dataset

`setSpecieFishery()` This method is used to store the distinct species in the fishery dataset

`splitSpecieSurvey()` This method is used to split the survey dataset by species

`splitSpecieFishery()` This method is used to split the fishery dataset by species

`addSpecieSurvey(sing_spe)` This method is used to initialize a new `surveyBySpecie` object

`addSpecieFishery(sing_spe)` This method is used to initialize a new `fisheryBySpecie` object

`setSpreaFishery()` This method is used to prepare the fishery LFD data for MCMC analysis

`setSpatFishery()` This method is used to setup the plot with the spatial distribution of the fishery dataset

`setSpreaSurvey()` This method is used to prepare the survey LFD data for MCMC analysis

`setSpatSurvey()` This method is used to setup the plot with the spatial distribution of the survey dataset

`setDepthSurvey()` This method is used to assign the depth of each survey tow

`setStratumSurvey()` This method is used to assign a depth stratum to each survey tow

`setAbuAvgAll()` This method is used to compute the spiecies abundances at each survey stratum

`setMeditsIndex()` This method is used to compute the MEDITS index

`setStrataAbu()` This method is used to compute the weighted number of individuals of each size in every stratum

`loadFleeEffoDbs(effort_path, met_nam, onBox = TRUE, perOnBox = 1)` This method is used to extract the vms data from one or more vmsbase DB

`ggplotRawPoints(year)` This method is used to plot the raw vms points

`ggplotFgWeigDists()` This method is used to plot the weighted average distance between harbours and fishing grounds

`setAvailData()` This method is used to gather the required data for the spatial clustering

`predictProduction(specie)` This method is used to compute the estimated production

`simProdAll(selRow = numeric(0))` This method is used to compute the simulated production

`genSimEffo(method = "flat", selRow = numeric(0), areaBan = numeric(0))` This method is used to create a simulated pattern of effort

`getSimSpatialCost()` This method is used to compute the simulated spatial costs

`getSimEffortCost()` This method is used to compute the simulated effort costs

`getSimProdCost()` This method is used to compute the simulated production costs

`getSimTotalCost()` This method is used to collect all the simulated costs

`getSimRevenue(selRow = numeric(0), timeScale = "Year")` This method is used to compute the simulated revenues

`getLWstat()` This method is used to compute the length/weight statistics for each fishing ground

`simulateFishery(thr0 = 100, effoBan = numeric(0), timeStep = "Year")` This method is used to simulate one year of fishing

`setSimResults()` This method is used to store the results of a simulation run

`ggplotFishingPoints(year)` This method is used to plot the fishing points

`setCellPoin()` This method is used assign a cell to each vms point

`setTrackHarb()` This method is used to assign the harbour to each fishing trip

`setFishGround(numCut)` This method is used to setup the fishing ground configuration

`addFg2Fishery()` This method is used to add the fishing ground information to each fishery data point

`addFg2Survey()` This method is used to add the fishing ground information to each survey data point

`setWeekEffoMatrCell()` This method is used to combine the raw effort points in weekly aggregated effort by cell

`setWeekEffoMatrGround()` This method is used to combine the raw effort points in weekly aggregated effort by fishing ground

`ggplotGridEffort(year)` This method is used to plot the gridded fishing effort

`getNnlsModel(specie, minobs, thr_r2)` This method is used to compute the coefficients of the NNLS model

`cohoDisPlot(whoSpe, whoCoh, whiYea, interp)` This method is used to store the spatial distribution of the species by cohort

Examples

```
# Initialize SmartProject
yourSmartRstudy <- SmartProject$new()

# Initialize fleet object
yourSmartRstudy$createFleet()


######################
## Environment Data ##
######################

# Locate the example environment asset' file
envAssetPath <- system.file("extdata/mapAsset.RDS", package = "smartR")

# Load environment asset' data
yourSmartRstudy$importEnv(readRDS(envAssetPath))

# Setup case study' map
yourSmartRstudy$sampMap$getGooMap()
yourSmartRstudy$sampMap$setGooGrid()
yourSmartRstudy$sampMap$setGooBbox()
yourSmartRstudy$sampMap$setGgDepth()
yourSmartRstudy$sampMap$setGgBioDF()
# View case study' grid
print(yourSmartRstudy$sampMap$gooGrid)


################
## Fleet Data ##
################

# Locate the example fleet asset' file
effAssetPath <- system.file("extdata/effAsset.RDS", package = "smartR")

# Load fleet asset' data
yourSmartRstudy$fleet$rawEffort <- readRDS(effAssetPath)

# Setup fishing vessel ids
yourSmartRstudy$fleet$setEffortIds()

# View speed distribution to setup fishing point filter
yourSmartRstudy$fleet$plotSpeedDepth(
which_year = "2012",
speed_range = c(2, 8),
depth_range = c(-20, -600)
)

# Setup fishing points' filter
yourSmartRstudy$fleet$setFishPoinPara(
speed_range = c(2, 8),
depth_range = c(-20, -600)
```

```
)

# Compute fishing points
yourSmartRstudy$fleet$setFishPoin()

# Assign cell id to each fishing point
yourSmartRstudy$setCellPoin()

# Add week and month number to each point
yourSmartRstudy$fleet$setWeekMonthNum()


#####################
## Fishing Grounds ##
#####################

# Setup available data to identify fishing areas
yourSmartRstudy$setAvailData()

# Setup cluster analysis input
yourSmartRstudy$sampMap$setClusInpu()

# Run cluster analysis with the SKATER method
yourSmartRstudy$sampMap$calcFishGrou(numCuts = 3, minsize = 10,
 modeska = "S", skater_method = "manhattan", nei_queen = FALSE)

# Setup cluster plot with 3 clusters
yourSmartRstudy$sampMap$setCutResult(ind_clu = 3)

# Map of the clusters' configuration
print(yourSmartRstudy$sampMap$ggCutFGmap)
```

`smartR` *Spatial Management and Assessment of demersal Resources for Trawl fisheries in R*

Description

smartR (Spatial Management and Assessment of demersal Resources for Trawl fisheries in R), a tool for assessing bio-economic feedback in different management scenarios. smartR combines information from different tasks gathered within the European Data Collection Framework for the fishery sector. The smartR package implements the SMART model in R, through the object-oriented programming paradigm, and within this package it is possible to achieve the complete set of analyses required by the SMART approach: from the editing and formatting of the raw data; the construction and maintenance of coherent datasets; the numerical and visual inspection of the generated metadata; to the final simulation of management scenarios and the forecast of their effects. The interaction between the user and the application could take place through invocation of methods via the command line or could be entirely committed to the graphical user interfaces (GUI).

Start Here

Here are listed the most important elements of a Smart analysis. The first, smartRgui, is the GUI developed to guide and assist the user, while the other five are the classes that make up the smartR package. See SmartProject for an example workflow.

smartRgui GUI to assist the analysis

SmartProject main project class

SurveyBySpecie survey data class

FisheryBySpecie fishery data class

SampleMap Environment data class

FishFleet Fleet data class

`smartRgui` *smartR GUI*

Description

The `smartRgui` function implements the main graphical user interface of smartR. Please, contact the package maintainer for detailed instructions.

Usage

```
smartRgui(smartRstudy = NULL)
```

Arguments

`smartRstudy`	SmartProject class

Value

This function does not return a value.

Examples

```
## Not run:
yourSmartRstudy <- SmartProject$new()
smartRgui(smartRstudy = yourSmartRstudy)

## End(Not run)
```

SurveyBySpecie *SurveyBySpecie*

Description

The SurveyBySpecie class implements the class of SMART to handle species samplings.

Format

R6Class object.

Value

Object of R6Class with attributes and methods for the survey data.

Fields

specie Name of the specie.

year Years in the time-serie.

rawLFD data.frame, raw length frequency distribution.

abuAvg data.frame, average abundances by depth' stratum.

meditsIndex data.frame, medits index by depth' stratum.

lengClas numeric, length classes.

nCoho numeric, number of cohorts.

spreDist list of DF, lfd by sex.

sprePlot plots of LFD statistics.

spreSpat list of DF, spatial distribution by sex.

sampMcmc list, mcmc output chains.

groMixout list of DF, aged individuals by sex.

groPars list of DF, growth parameters by sex.

LWpar list of DF, length/weight parameters by sex.

Methods

initialize(sing_spe) Automatic initialization made by the SmartProject class

setRawData(raw_data) This method is used load the initial raw dataset

setYears() This method is used to store the years in the provided time-serie

setSpecie() This method is used to store the name of the specie of the initial raw data

setLClass() This method is used to store the unique length values of the sampled specie

setDepth(bathyMatrix) This method is used to assign the depth value corresponding to each sampling location

`setStratum(vecStrata)` This method is used to set the depth strata of each sampling location

`setIndSpe()` This method is used to aggregate the abundance data into the medits index

`setAbuAvg()` This method is used to standardize the spatial abundances by depth strata

`setNCoho(num_coh)` This method is used to setup the number of cohorts for the ageing module

`setLWpar(alphaVal, betaVal, sex)` This method is used to store the alpha and beta values for the length/weight relationship

`setWeight(sexVal = "Female")` This method is used to compute the fish weight given their length and the LWrelationship

`setSpreDistSing()` This method is used to spread the aggregated LFD abundances into single individuals

`setSprePlot(sampSex)` This method is used to setup the plots of the LFD statistics

`setSpatDistSing()` This method is used to setup the spatial distribution of the single specimens

`setSpatPlot(sampSex)` This method is used to store the spatial plots of the population

`getMCsamps(numSamp, numAdap, numIter, sexDrop, curveSel)` This method is used to get a sample of the population to feed the mcmc module

`getGrowPar(sexDrop)` This method is used to extract the growth parameters from the mcmc results

`getMCage(sexDrop)` This method is used to assign an age to each fish

`setMCplot(sexDrop, selCurve)` This method is used to setup the plot of the mcmc results

`calcMixDate(nAdap, nSamp, nIter, sexDrop, curveSel)` This method is used to estimate the growth parameters of a population

`ggplotMcmcOut(selCompo, selSex)` This method is used to output the stored plots of mcmc results

Index