|  |
| --- |
| **Vmstools Reference Card** |
|  |
| by Niels T. Hintzen, IMARES, part of Wageningen UR, [niels.hintzen@wur.nl](mailto:niels.hintzen@wur.nl)  2011-02-22. See also: http://code.google.com/p/vmstools/ |
|  |
| **Data** |
| **data(eflalo)** load eflalo2 test dataset |
| **data(tacsat)** load the tacsat test dataset |
| **data(harbours)** load the harbour test dataset |
| **data(VMShf)** load the VMS high ping rate test dataset  **data(correspLevel7to5)** load species linking dataset  **data(correspMixedMetier)** load Mixed métier dataset  **data(europa)** load shapefile of Europe  **data(speciesLatinNames)** load Latin name lookup table  **formatEflalo(eflalo)** put eflalo columns in right format  **formatTacsat(tacsat)** put tacsat columns in right format  **readEflalo(file,sep,dec)** read eflalo from file  **readTacsat(file,sep,dec)** read tacsat from file |
| **Metièr definitions** |
| **compareToOrdination(data, step,clusters,tabClusters)**  compare metiers by simple ordination methods  **extractTableMainSpecies(data, names,#params,logevent)**  find métier from small eflalo dataset  **getEflaloMetierLevel7(data,names,path,criteria,#param,logevent,...)** find métier from full eflalo dataset  **predictMetier(data,cluster,newData)** predict métier for new  eflalo data |
| **Tacsat Behavior Analyses** |
| **filterTacsat(tacsat)** filter out records that do not lay within a speed  range and/or change of heading interval |
| **pointInHarbour(tacsat)** flags tacsat points that are positioned in a  harbour  **pointOnLand(tacsat)** flags tacsat points that are positioned on land  **segmentTacsatSpeed(tacsat,vessel,settings)** detect fishing  speed thresholds  **sortTacsat(tacsat)** sort tacsat data by year, date and position |
| **Link eflalo - tacsat** |
| **merge.vms.to.logbook.at.the.ping.scale (eflalo2,tacsatplus,general,vesselid)** merge eflalo2 and  tacsat+ on tacsat ping level |
| **mergeEflalo2Tacsat(eflalo2,tacsat)** merge eflalo2 and tacsat at  trip level  **effort(x,level,unit) c**alculate effort by trip from eflalo or tacsat  **mergeEflalo2Pings(x,level,unit)** coupling and dispatching  eflalo data onto tacsat pings  **splitAmongPings(tacsat,eflalo,variable,level)**  dispatching eflalo data onto tacsat pings |
|  |
| **Interpolate tacsat** |
| **interpolateTacsat(tacsat,interval,margin,res,method,params,headingAdjustment)** interpolate tacsat data between pings *x*  minutes apart using straight line or cubic Hermite spline interpolation  **interpolation2Tacsat(interpolation,points)** convert  interpolation format into tacsat format |
| **calculateCI(longitudes,latitudes,interpolations,indexInterpolation,tacsat,grid,spatialDataFrame,singleInterpolation,indexTacsat,parameters)** calculate a confidence  interval around the interpolation |
| **diffInter(interpolation,tacsatHighRes)** calculate difference  between true high-resolution data and interpolated dataset |
| **distanceInterpolation(interpolation)** calculate length of  interpolation |
| **distanceTacsat(tacsat,index)** calculate distance between gps  coordinates of a complete VMS dataset  **addWidth(interpolation, gearWidth)** add a gearwidth to an  interpolation  **Calculate indicators**  **indicators(#indicator,tacsat,...) c**alculate DCF indicators 5-  7 based on tacsat dataset  **tacsatMCP(tacsat,threshold)** flag pings within a minimum convex  polygon |
| **Plotting** |
| **createGrid(xrange,yrange,resx,resy)** create spatial grid |
| **mapGrid(spatialDataFrame,...)** map grids |
| **vmsGridCreate(data,cellsize,...)** create and map grids  **Grid2KML(output,quantity)** export data to KML file  **landingsMaps2GIFanim(files,species)** create animated GIF  from single plots  **pings2EffortMaps(output,file)** auto-create effort maps from  output file  **pings2LandingsMaps(output,file)** auto-create landings maps from  output file  **Databases**  **pings2Fishframe(output,year,country)** format data from  mergeEflalo2Pings into Fishframe format |
| **Converting** |
| **bearing(lon1,lat1,lon2,lat2)** calculate bearing from tacsat  longitude and latitude data |
| **degree2Km(lon,lat,degree)** convert degrees to kilometers, only in  longitudinal direction |
| **distance(lon1,lat1,lon2,lat2)** calculate distance between two  gps coordinates  **lonLatRatio(lon,lat)** ratio between longitude and latitude |
| **km2Degree(lon,lat,km)** convert kilometers to degrees, only in  longitudinal direction |
| **ICESarea(tacsat)** calculate ICES area from gps location |
| **ICESrectangle(tacsat)** calculate ICES rectangle from gps location  **ICESrectangle2LonLat(rectangle)** calculate gps location from  ICES rectangle from |
| **CSquare(lon,lat, degrees)** calculate CSquare notation from gps  Location  **surface(grid,method)** calculate surface of grid cells |