

# Package ‘FAMEFMR’

March 26, 2021

**Title** Functions used in the DELWP FAME fire analysis process

**Version** 0.1.4.0

**Description** The package contains all the functions necessary to run the DELWP FAME process of analysis of effects of a given fire history on environmental resilience metrics. These include Tolerable fire Interval (TFI) status, area burned below TFI (BBTFI) and changes in relative abundance (RA) of fauna species. It allows calculation of all inter fire intervals and numbers of times an area has been burned by a given sequence of fires supplied as a shapefile. The FAME analysis can be run using standard R scripts, or alternatively via a shiny web GUI. The package does not contain the required input files (lookup tables, rasters of species HDMs etc) as these are too large to be loaded to github and are available separately from an aws repository. The package is best used with the associated scripts and shiny app that can be cloned or downloaded from <https://github.com/nevilamos/FAMEshiny>.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Imports** doParallel,  
dplyr,  
fasterize,  
foreign,  
foreach,  
iterators,  
magrittr,  
Matrix.utils,  
mgcv,  
raster,  
Rfast,  
rlang,  
sf,  
tabularaster,  
tibble,  
tictoc,  
tidyr,  
tidyselect

**Depends** data.table

## R topics documented:

|                                  |           |
|----------------------------------|-----------|
| add_xystring . . . . .           | 2         |
| calcBBTFI_2 . . . . .            | 3         |
| calcDeltaAbund . . . . .         | 4         |
| calcU_All_Combs . . . . .        | 4         |
| calc_DraftSpList . . . . .       | 5         |
| calc_SpeciesRA . . . . .         | 6         |
| calc_Spp_EFG_LMU . . . . .       | 7         |
| calc_TFI_2 . . . . .             | 8         |
| cellsToHectares . . . . .        | 9         |
| cropNAborder . . . . .           | 9         |
| fhProcess . . . . .              | 10        |
| get_Spp_No . . . . .             | 11        |
| inputRasters . . . . .           | 12        |
| Join_Names . . . . .             | 12        |
| LBY_f . . . . .                  | 13        |
| makeAbundDataLong . . . . .      | 13        |
| makeGS_LU . . . . .              | 14        |
| makeGS_Summary . . . . .         | 14        |
| makeHDMVals . . . . .            | 15        |
| makeHDMValsfromRasters . . . . . | 15        |
| make_Spp_LU_list . . . . .       | 16        |
| notAllIn . . . . .               | 16        |
| removeEmptyDirs . . . . .        | 17        |
| unlistPivot_wider . . . . .      | 17        |
| <b>Index</b>                     | <b>18</b> |

---

|              |   |
|--------------|---|
| add_xystring | <i>Adds concatenated String of X and Y coordinates of centroids of polygons to Simple Features polygon object. This String acts as a key to identify spatially identical polygons for use in tidyverse pivot functions.</i> |
|--------------|---|

---

### Description

Adds concatenated String of X and Y coordinates of centroids of polygons to Simple Features polygon object. This String acts as a key to identify spatially identical polygons for use in tidyverse pivot functions.

### Usage

```
add_xystring(myDF)
```

### Arguments

myDF                      sf polygon object

### Value

character vector of XYStrings

calcBBTFI\_2

*Calculate area BBTFI and BBTFI rasters***Description**

Calculate area BBTFI and BBTFI rasters

**Usage**

```
calcBBTFI_2(
  myFHAnalysis = FHAnalysis,
  myAllCombs = allCombs,
  makeBBTFIrasters = makeBBTFIrasters,
  myResultsDir = NULL
)
```

**Arguments**

**myFHAnalysis**     list of Fhire history analysis components created by function fhProcess()

**myAllCombs**       list made by function calc\_U\_AllCombs

**makeBBTFIrasters**     logical whether or not to export rasters for BBTFI to disk

**myResultsDir**     path of directory where results will be written usually generated by FAME script

**Details**

Calculate summary area burned below TFI BBTFI for each SEASON in analysis (accommodating Hi and Lo fire intensity of first burn to determine TFI) and cumulative area BBTFI. Also optionally outputs rasters mapping areas BBTFI

**Value**

list containing:

- the date sequence matrix for each cell of the raster
- the EFG TFI Lookup for each cell of the raster
- the raster resolution used.
- Optionally outputs rasters of BBTFI to disk if makeBBTFIrasters==TRUE.

---

calcDeltaAbund

*Summary of changes in relative abundance*


---

### Description

Summary of changes in relative abundance

### Usage

```
calcDeltaAbund(
  SpYearSumm = SpYearSummWide,
  myFHAnalysis,
  myBaseline,
  myResultsDir
)
```

### Arguments

|              |  |
|--------------|--|
| SpYearSumm   | data.frame output by function calc_SpeciesRA()   |
| myFHAnalysis | list containing all the fire history spatial attributes created by function fhProcess  |
| myBaseline   | integer single SEASON or sequence of SEASons used to create the baseline relative species abundance for comparison of change |
| myResultsDir | path of directory where results will be written usually generated by FAME script #'  |

### Details

Calculates the change in relative abundance compared to a baseline SEASON or mean of SEASONS

### Value

data frame wide format summary chance in relative abundance of species SEASON

---

calcU\_All\_Combs

*Calculate all combinations of input raster values*


---

### Description

Calculate all combinations of input raster values

### Usage

```
calcU_All_Combs(
  myFHAnalysis = FHAnalysis,
  myCropRasters = cropRasters,
  myRasterRes = RasterRes
)
```

**Arguments**

|               |  |
|---------------|--|
| myFHAnalysis  | list containing all the fire history spatial attributes created by function fhProcess()                                    |
| myCropRasters | list of rasters and indices and cell values created by function cropNAborder()   |
| myRasterRes   | Resolution of Rasters to be used in the analysis set in settings file in R-script version needs to be set in shiny version |

**Value**

list of:

- data.table giving all combinations of cell values from the input rasters for the FAME analysis
- integer index mapping unique combinations (above) to raster cells

---

|                  |   |
|------------------|---|
| calc_DraftSpList | <i>Calculate a draft species list for defined polygon</i> |
|------------------|---|

---

**Description**

Calculate a draft species list for defined polygon

**Usage**

```
calc_DraftSpList(
  REG_NO,
  RasterRes = 225,
  PUBLIC_LAND_ONLY,
  myPoly = clipPoly,
  generalRasterDir = "../InputGeneralRasters",
  splist = "../ReferenceTables/DraftTaxonListStatewidev2.csv",
  myHDMVals = "../HDMs/HDMVals225.qs"
)
```

**Arguments**

|                  |   |
|------------------|---|
| REG_NO           | integer DELWP fire region number 1:6 ,99 for Statewide analysis, or 7 for ad-hoc boundary polygon default =7 (see look up table REG_LUT for values) |
| RasterRes        | integer 225 - raster resolution is always 225 for this function for speed   |
| PUBLIC_LAND_ONLY | logical whether to restrict analysis to public land only or the whole polygon   |
| myPoly           | default clipPoly sf polygon data frame of LF_REGIONs (default) or ad hoc polygon - used in conjunction with REG_NO                                  |
| generalRasterDir | relative path to directory containing rasters of FIRE_REG, and PUBLIC LAND (PLM_GEN)  |
| splist           | path to default species attribute table default is  |
| myHDMVals        | matrix of cell values for Habitat Distribution Model rasters always 225m for speed  |

## Details

Calculate the proportion of cells for the HDM in the region for each species is intended only as a starting point and requires manual quality control to produce a useful species list for the area by editing the resulting .csv file

## Value

data.frame created from splist with columns appended for:

- cellsInState count of the number of cells in the state within the Binary HDM for the species
- cellsInArea count of the number of cells within myPoly and within the Binary HDM for the species
- areaProp proportion of binary HDM for the state within myPoly

---

|                |  |
|----------------|--|
| calc_SpeciesRA | <i>Species' relative abundance calculation and summary</i> |
|----------------|--|

---

## Description

Species' relative abundance calculation and summary

## Usage

```
calc_SpeciesRA(
  myFHAnalysis,
  myAllCombs = allCombs,
  myHDMspp_NO = HDMspp_NO,
  myWriteSpRasters = FALSE,
  myLU_List = LU_List,
  myResultsDir = ResultsDir,
  myHDMVals = HDMVals,
  myTaxonList = TaxonList,
  writeYears = NULL,
  myWriteSp = writeSp,
  myIDX = cropRasters$IDX
)
```

## Arguments

|                  |  |
|------------------|--|
| myFHAnalysis     | list containing all the fire history spatial attributes created by function fhProcess  |
| myAllCombs       | all combinations of raster values object produced by function calc_U_AllCombs  |
| myHDMspp_NO      | vector of TAXON_IDs for species to be included in output   |
| myWriteSpRasters | logical: whether to also write species abundance rasters to disk   |
| myLU_List        | list of species abundance lookup arrays created by function make_Spp_LU_list()   |
| myResultsDir     | path of directory where results will be written usually generated by FAME script   |
| myHDMVals        | list of sparse matrices of cell values for Habitat Distribution Model rasters for (at least) all TAXON_ID in myHDMspp_NO generally provided in settings file and read by FAME script |

|             |   |
|-------------|---|
| myTaxonList | data.frame of species attributes ( read from default or user provided .csv)   |
| writeYears  | vector for SEASONS for which rasters are to be written otherwise if write-SpRasters == TRUE, if writeYears == NULL then all SEASONS are written out |
| myWriteSp   | vector of TAXON_IDs provided if only subset of species rasters are required as output.  |
| myIDX       | index of cells to extract values for from cropRasters object  |

### Details

Calculates the relative abundance of species for each raster cell in analysis and summaries these as summed abundance each season. Optionally it also write relative abundance rasters for species to disk

### Value

list of two data frames:

- SpYearSummWide summary of relative abundance of species by pivoted wide by SEASONS
- SpYearSummLong Long Format summary of relative abundance of species by SEASONS

---

|                  |  |
|------------------|--|
| calc_Spp_EFG_LMU | <i>Calculate the species in each EFG in given area for GSO calculations.</i> |
|------------------|--|

---

### Description

works by using the indices of the standard dimensions raster that are in the supplied shapefile region boundary (via function cropNAborder )

### Usage

```
calc_Spp_EFG_LMU(
  REG_NO,
  RasterRes = 225,
  PUBLIC_LAND_ONLY,
  myPoly = clipPoly,
  generalRasterDir = "./InputGeneralRasters",
  splist = "./ReferenceTables/DraftTaxonListStatewidev2.csv",
  myHDMVals = "./HDMs/HDMVals225.qs",
  myResultsDir = ResultsDir,
  TFI_LUT = TFI_LUT
)
```

### Arguments

|                  |   |
|------------------|---|
| REG_NO           | integer DELWP fire region number 1:6 ,99 for Statewide analysis, or 7 for ad hoc boundary polygon default =7 (see look up table REG_LUT for values) |
| RasterRes        | integer 225 - raster resolution is always 225 for this function for speed   |
| PUBLIC_LAND_ONLY | logical whether to restrict analysis to public land only or the whole polygon   |

|                  |   |
|------------------|---|
| myPoly           | default clipPoly sf polygon data frame of LF_REGIONs (default) or ad hoc polygon - used in conjunction with REG_NO  |
| generalRasterDir | relative path to directory containing rasters of FIRE_REG, and PUBLIC LAND (PLM_GEN)                                |
| splist           | path to default species attribute table default is <code>"/ReferenceTables/DraftTaxonListStatewidev2.csv"</code>    |
| myHDMVals        | sparse matrix of cell values for Habitat Distribution Model rasters at 225m pixel size # saved as a qs file on disk |
| myResultsDir     | path of directory where output will be saved  |
| TFI_LUT          | data.frame lookup table for EFG loaded in setup   |

calc\_TFI\_2

*Main Tolerable fire interval (TFI) status calculation*

## Description

Main Tolerable fire interval (TFI) status calculation

## Usage

```
calc_TFI_2(
  myFHAnalysis = FHAnalysis,
  myAllCombs = allCombs,
  myTFI_LUT = TFI_LUT,
  OutputRasters = makeTFIRasters,
  myResultsDir = ResultsDir
)
```

## Arguments

|               |   |
|---------------|---|
| myFHAnalysis  | list containing all the fire history spatial attributes created by function fhProcess()                   |
| myAllCombs    | list made by function calc_U_AllCombs   |
| myTFI_LUT     | data.frame Lookup table from EFG for "MIN_LO_TFI", "MIN_HI_TFI", "MAX_TFI", "EFG_NAME" read from settings |
| OutputRasters | logical whether to output rasters of TFI status for each year   |
| myResultsDir  | path of directory where results will be written usually generated by FAME script                          |

## Details

Calculates where each cell is currently at below MinTFI or above MAX\_TFI returns the per cell and long table summarised by multiple admin units and evc

## Value

data.frame with area in each TFI status for each combination in myAllCombs



---

|                 |  |
|-----------------|--|
| cellsToHectares | <i>Calculates multiplier to convert from raster cell count to area in hectares</i> |
|-----------------|--|

---

### Description

Calculates multiplier to convert from raster cell count to area in hectares

### Usage

```
cellsToHectares(RasterMetres = RasterRes)
```

### Arguments

RasterMetres      numeric Value cell resolution in Metres (usually from RasterRes in settings file).

### Value

numeric Multiplier to convert cell count to area in hectares

---

|              |  |
|--------------|--|
| cropNAborder | <i>crop border of NA cells from rasters and get cell indices for remaning cells function to get the minimum bounding box of the cells with non NA values in a raster and save them to crop other rasters to same extent. also creates some rasters cropped to correct extent for instance for region and EFG also gets indices of cells in raster of same extent as crop to the shape provided</i> |
|--------------|--|

---

### Description

crop border of NA cells from rasters and get cell indices for remaning cells function to get the minimum bounding box of the cells with non NA values in a raster and save them to crop other rasters to same extent. also creates some rasters cropped to correct extent for instance for region and EFG also gets indices of cells in raster of same extent as crop to the shape provided

### Usage

```
cropNAborder(
  REG_NO = 7,
  myRasterRes = RasterRes,
  PUBLIC_LAND_ONLY,
  myPoly = clipPoly,
  generalRasterDir = "../InputGeneralRasters"
)
```

**Arguments**

|                  |   |
|------------------|---|
| REG_NO           | integer DELWP fire region number 1:6 ,99 for Statewide analysis, or 7 for ad hoc boundary polygon default =7 (see look up table REG_LUT for values) |
| myRasterRes      | numeric raster resolution of the analysis in metres ( usually set in settings file or shiny app)  |
| PUBLIC_LAND_ONLY | Logical TRUE/FALSE  |
| myPoly           | default clipPoly sf polygon data frame of LF_REGIONs (default) or ad hoc polygon - used in conjunction with REG_NO                                  |
| generalRasterDir | relative path to directory containing rasters of DELWP FIRE_REG, DELWP REGION, EFG, PUBLIC LAND (PLM_GEN)   |

**Value**

A list containing:

- Raster raster cropped of all border rows and columns that are all NA,
- Extent extent of the raster
- IDX integer vector cell numbers of cells in the cropped raster
- clipIDX integer vector cell numbers only for cells with the input polygon
- EFG integer vector EFG values for cells within clipped area
- RGN integer vector Fire Region numbers for cells within clipped area
- DELWP integer vector DELWP Region numbers for cells within clipped area
- PLM logical for cells within clipped area

---

fhProcess

---

*Main Fire History Fire Sequence analysis function*


---

**Description**

Main Fire History Fire Sequence analysis function

**Usage**

```
fhProcess(
  rawFH = "path of the rawFH file to use - a shapefile",
  start.SEASON = NULL,
  end.SEASON = NULL,
  OtherAndUnknown,
  validFIRETYPE
)
```

## Arguments

|                 |   |
|-----------------|---|
| rawFH           | path to the input fire history shapefile usually provided in settings   |
| start.SEASON    | integer First SEASON for which output is wanted (four digit year as integer), if NULL then second season in in history is used (cannot use first season because it has no interval, this may still fail if there is no overlap) |
| end.SEASON      | integer Last SEASON required, if NULL then largest value in fire history scenario used  |
| OtherAndUnknown | integer Value to use for cases where fire type is: "OTHER" or "UNKNOWN" = NA, "BURN" = 1, "BUSHFIRE" = 2. NA = Fire excluded from analysis. usually set in settings file  |
| validFIRETYPE   | character vector of valid FIRETYPE values for checking the input file , provided in settings file.  |

## Details

The function takes a shapefile of Fire history contain polygons with two fields FIRETYPE and SEASON Where polygons of different FIRETYPE or SEASON overlap the function constructs unique non-overlapping polygon of their intersections ( and non intersecting areas ) and attributes each polygon with sequential fire SEASON (SEAS01, SEAS02 ...) and corresponding FIRETYPE (TYPE01,TYPE02 ...)

It then calculates all the intervals between sequential fires, and Time Since fire (TSF) and Last Fire Type (LFT) and Last burnt year (LBY) for each SEASON as defined in the input arguments, these values are append to the output sf polygon dataframe.

## Value

A list containing:

- OutDF sf polygons dataframe containing all the fire history attributes
- TimeSpan integer vector sequence of SEASONS to in the analysis output
- YSFNames names of TSF years in output, needed by downstream functions
- LBYNames names of LBY years in output, needed by downstream functions
- LFTNames names of LBY years in output, needed by downstream functions

---

|            |  |
|------------|--|
| get_Spp_No | <i>Extract VBA (Victorian Biodiversity Atlas) species ID numbers from file paths extracts four or five digit species numbers (Victorian Biodiversity Atlas TAXON_IDs) from vector of paths or file names containing files of e.g. species HDMS containing the 5 digit TAXON_ID in their name</i> |
|------------|--|

---

## Description

Extract VBA (Victorian Biodiversity Atlas) species ID numbers from file paths extracts four or five digit species numbers (Victorian Biodiversity Atlas TAXON\_IDs) from vector of paths or file names containing files of e.g. species HDMS containing the 5 digit TAXON\_ID in their name

Usage

get\_Spp\_No(x = "Vector of Sp file Pathnames")

Arguments

x                      Vector of species file Pathnames containing VBA numbers

Value

numeric vector of 4or 5 digits (usually TAXON\_ID)

---

|              |  |
|--------------|--|
| inputRasters | <i>Set correct input general rasters</i> |
|--------------|--|

---

Description

Set correct input general rasters

Usage

inputRasters(RasterRes)

Arguments

RasterRes              numeric raster resolution of the analysis in metres ( usually set in settings file or shiny app)

Value

list of input raster names correct for RasterRes or error if RasterRes is not 75 or 225

---

|            |   |
|------------|---|
| Join_Names | <i>Joins one or more lookup tables to table containing ID values Function joins Lookup tables (LUTS) to dataframe containing ID_NO: Name combinations</i> |
|------------|---|

---

Description

Joins one or more lookup tables to table containing ID values Function joins Lookup tables (LUTS) to dataframe containing ID\_NO: Name combinations

Usage

Join\_Names(myDF, LUTS = c("TFI\_LUT", "FIREFMZ\_LUT", "REG\_LUT", "DELWP\_LUT"))

Arguments

myDF                    dataframe or similar containing indices for the LUTS listed, to which the LUTS will be dplyr::left\_joined

LUTS                    vector of names of LUTS in memory defaults =c("TFI\_LUT","FIREFMZ\_LUT","REG\_LUT","DELWP\_LUT")

### Value

a data.frame with the LUTS joined to it

---

|       |  |
|-------|--|
| LBY_f | <i>Calculate last burned year matrix (LBY)</i> |
|-------|--|

---

### Description

Calculate last burned year matrix (LBY)

### Usage

```
LBY_f(M, y)
```

### Arguments

|   |  |
|---|--|
| M | numeric matrix of fire sequences sequence in rows, values are SEASON |
| y | numeric SEASON   |

### Details

Function to calculate last burnt year (LBY) from matrix of rows of fire season iterating by year (y) used in calc\_TFI\_2

### Value

matrix of last burned year row for each unique fire history column for each SEASON

---

|                   |  |
|-------------------|--|
| makeAbundDataLong | <i>Makes Long format Fauna abundance table</i> |
|-------------------|--|

---

### Description

Makes Long format Fauna abundance table

### Usage

```
makeAbundDataLong(
  AbundDataByGSFile = "../ReferenceTables/OrdinalExpertLong.csv",
  myEFG_TSF_4GS = EFG_TSF_4GS
)
```

### Arguments

|                   |  |
|-------------------|--|
| AbundDataByGSFile | .csv input file containing fields "EFG_NO", "GS4_NO", "FireType" , "Abund", "TAXON_ID" with Abund values for both FireTypes for each growth stage "GS4_NO" |
| myEFG_TSF_4GS     | table of each combination of "EFG_NO", "GS4_NO", and "YSF" generally read in at beginning of FAME in settings file   |

**Details**

Supporting function to Make long format table for fauna abundance scores by "TAXON\_ID", Fire-Type EFG and Growth Stage from input wide format table currently deals only with FireType of "High" and "Low" which are converted to 2 and 1 respectively

**Value**

long format table with one row for each combination of "EFG\_NO", "GS4\_NO", "FireType", "Abund", "TAXON\_ID" and "YSF" sorted by TAXON\_ID

---

|           |  |
|-----------|--|
| makeGS_LU | <i>Make long format Growth Stage Lookup matrix</i> |
|-----------|--|

---

**Description**

Make long format Growth Stage Lookup matrix

**Usage**

```
makeGS_LU(myEFG_TSF_4GS = EFG_TSF_4GS)
```

**Arguments**

myEFG\_TSF\_4GS    data.frame of growth stages for each EFG with start and end years

**Details**

expands a growth stage lookup table (provided in settings file) from four growth stages (1:4) per EFG with their years since fire spans as min(YSF) and max(YSF) to an array with YSF as row, EFG\_NO as column and growth stage (1:4) as value. NOTE: YSF has 1 added to both the Lookup and the input to deal with YSF==0 which cannot be used in the array indexing

**Value**

matrix rows YSF, columns EFG\_NO, values GS number (1:4)

---

|                |  |
|----------------|--|
| makeGS_Summary | <i>Summarise area by growth stage.</i> |
|----------------|--|

---

**Description**

Summarise area by growth stage.

**Usage**

```
makeGS_Summary(myFHAnalysis = FHAnalysis, myAllCombs = allCombs)
```

**Arguments**

myFHAnalysis    list containing all the fire history spatial attributes created by function fhProcess  
 myAllCombs      list made by function calc\_U\_AllCombs

**Details**

Generates wide and long format summary of area for each EFG and season grouped by EFG, EFG\_NAME, PLM ,FIRE\_FMZ\_NAME, FIRE\_REGION\_NAME, DELWP\_REGION.

**Value**

list of two data.frames grouped by EFG, EFG\_NAME, PLM ,FIRE\_FMZ\_NAME, FIRE\_REGION\_NAME, DELWP\_REGION

- GS\_Summary\_wide Wide format table summarises area by Growth Stage and SEASON
- GS\_Summary\_long Long format table summarises area by Growth Stage and SEASON

---

|             |  |
|-------------|--|
| makeHDMVals | <i>Extract HDM values for relevant cells and resolution for use in RA calculations</i> |
|-------------|--|

---

**Description**

Extract HDM values for relevant cells and resolution for use in RA calculations

**Usage**

```
makeHDMVals(
  myHDMspp_NO = HDMspp_NO,
  myCropRasters = cropRasters,
  RasterRes = myFHAnalysis$RasterRes
)
```

**Arguments**

|               |   |
|---------------|---|
| myHDMspp_NO   | vector of TAXON_IDs for species to be included in output  |
| myCropRasters | list of rasters and indices and cell values created by function cropNAborder()                            |
| RasterRes     | numeric raster resolution of the analysis in metres (225 or 75 usually set in settings file or shiny app) |

---

|                        |  |
|------------------------|--|
| makeHDMValsfromRasters | <i>Function makes a matrix of HDM values(1,NA) constrained to those cells that are indexed in the cropped area</i> |
|------------------------|--|

---

**Description**

Function makes a matrix of HDM values(1,NA) constrained to those cells that are indexed in the cropped area

**Usage**

```
makeHDMValsfromRasters(myHDMspp_NO = HDMspp_NO, myCropRasters = cropRasters)
```

**Arguments**

myHDMspp\_NO      vector of TAXON\_IDs for species to be included in output  
 myCropRasters    list of rasters and indices and cell values created by function cropNAborder()

---

make\_Spp\_LU\_list      *Generate list of species abundance lookup arrays*

---

**Description**

Generate list of species abundance lookup arrays

**Usage**

```
make_Spp_LU_list(myHDMspp_NO = HDMspp_NO, myAbundDataLong = ExpertDataLong)
```

**Arguments**

myHDMspp\_NO      vector of VBA IDs for species to be included in analysis  
 myAbundDataLong      long format input lookup table of species abundance x YSF x EFG\_NO x FIRE-  
    TYPE\_NO

**Details**

function creates a list of Lookup arrays for each taxon (TAXON\_ID) for YSF x EFGNO x Fire-  
 TypeNo these are then used in spatial calculation of species abundance functions

**Value**

list of 3D arrays named by TAXON\_ID of relative abundance value for YSF x EFG x FIRE-  
 TYPE\_NO

---

notAllIn      *Checks whether all values in one vector are in another vector*

---

**Description**

Checks whether all values in one vector are in another vector

**Usage**

```
notAllIn(x, v = V)
```

**Arguments**

x      Vector of values to check if all are in second vector  
 v      Second vector of values that may or may not contain all values in x

**Value**

logical



---

|                 |   |
|-----------------|---|
| removeEmptyDirs | <i>remove empty directories from path</i> |
|-----------------|---|

---

**Description**

remove empty directories from path

**Usage**

```
removeEmptyDirs(rootDir = "./Results")
```

**Arguments**

|         |   |
|---------|---|
| rootDir | relative path to remove all empty subdirectories from. Default value <code>"./Results"</code> |
|---------|---|

**Details**

Removes all empty subdirectories from the nominated path does not remove the nominated path directory even if empty

---

|                   |  |
|-------------------|--|
| unlistPivot_wider | <i>Fix Pivot_wider list of lists columns</i> |
|-------------------|--|

---

**Description**

Fix Pivot\_wider list of lists columns

**Usage**

```
unlistPivot_wider(df)
```

**Arguments**

|    |  |
|----|--|
| df | wide format data frame with fields that are lists of lists |
|----|--|

**Details**

Supporting function to deal with pivot\_wider returning list of lists in some cases

**Value**

wide format data frame without fields that are lists of lists

# Index

`add_xystring`, [2](#)

`calc_DraftSpList`, [5](#)  
`calc_SpeciesRA`, [6](#)  
`calc_Spp_EFG_LMU`, [7](#)  
`calc_TFI_2`, [8](#)  
`calcBBTFI_2`, [3](#)  
`calcDeltaAbund`, [4](#)  
`calcU_All_Combs`, [4](#)  
`cellsToHectares`, [9](#)  
`cropNAborder`, [9](#)

`fhProcess`, [10](#)

`get_Spp_No`, [11](#)

`inputRasters`, [12](#)

`Join_Names`, [12](#)

`LBY_f`, [13](#)

`make_Spp_LU_list`, [16](#)  
`makeAbundDataLong`, [13](#)  
`makeGS_LU`, [14](#)  
`makeGS_Summary`, [14](#)  
`makeHDMVals`, [15](#)  
`makeHDMValsfromRasters`, [15](#)

`notAllIn`, [16](#)

`removeEmptyDirs`, [17](#)

`unlistPivot_wider`, [17](#)