# Package 'faoswsProduction'

April 6, 2015

**Type** Package

**Title** Package to perform the imputation of area harvested, production and yield for the FAO production domain.

**Version** 1.0.0

**Date** 2015-01-29

**Author** Joshua M. Browning <joshua.browning@fao.org>, Michael C. J. Kao <michael.kao@fao.org>

**Maintainer** Joshua M. Browning <joshua.browning@fao.org>

**Description** This package provides all the functions to perform imputation of area, production and yield for the FAO production domain.

**URL** https://github.com/mkao006/sws_imputation

**License** GPL (>= 3)

**Imports** lme4 (>= 1.1-7), reshape2 (>= 1.4), earth (>= 3.2-7), forecast (>= 5.5), zoo (>= 1.7-11), RColorBrewer (>= 1.0-5)

**Depends** faoswsFlag (>= 0.1.0), faoswsImputation, faoswsUtil, splines (>= 3.0.2), data.table (>= 1.9.2)

**LazyData** yes

**ZipData** no

**VignetteBuilder** knitr

**Suggests** knitr, ggplot2

## R topics documented:

## balanceAreaHarvested *Function to compute area harvested when new production and yield are given.*

### Description

Function to compute area harvested when new production and yield are given.

### Usage

```
balanceAreaHarvested(data, imputationParameters, processingParameters)
```

### Arguments

| | |
|---|---|
| data | The data.table object containing the data. |
| imputationParameters | A list of the parameters for the imputation algorithms. See defaultImputationParameters() for a starting point. |
| processingParameters | A list of the parameters for the production processing algorithms. See defaultProductionParameters() for a starting point. |

## balanceProduction *Function to compute production when new area harvested and yield are given.*

### Description

Function to compute production when new area harvested and yield are given.

### Usage

```
balanceProduction(data, imputationParameters, processingParameters)
```

### Arguments

| | |
|---|---|
| data | The data.table object containing the data. |
| imputationParameters | A list of the parameters for the imputation algorithms. See defaultImputationParameters() for a starting point. |
| processingParameters | A list of the parameters for the production processing algorithms. See defaultProductionParameters() for a starting point. |

`computeYield` *Function to compute and update yield*

## Description

Function to compute and update yield

## Usage

```
computeYield(data, newMethodFlag, flagTable = faoswsFlagTable,
  unitConversion = 1, processingParameters)
```

## Arguments

| | |
|---|---|
| `data` | The data.table object containing the data. |
| `newMethodFlag` | The flag to be used to update the yield method flag when imputation occurs. |
| `flagTable` | see data(faoswsFlagTable) in **faoswsFlag** |
| `unitConversion` | Yield is computed as (production) / (area) and multiplied by unitConversion. This parameter defaults to 1. |
| `processingParameters` | A list of the parameters for the production processing algorithms. See defaultProductionParameters() for a starting point. |

`defaultProcessingParameters`
*Default Processing Parameters*

## Description

This function can be used to generate the input parameters for the data pre-processing code. This is a good way to get a list of the required parameters and then modify parameters to match your particular configuration.

## Usage

```
defaultProcessingParameters()
```

## Details

Below is a description of the parameters:

- productionValue: The column name of the production variable.
- productionObservationFlag: The column name of the observation flag corresponding to the production variable.
- productionMethodFlag: The column name of the method flag corresponding to the production variable.
- yieldValue: The column name of the yield variable.

- yieldObservationFlag: The column name of the observation flag corresponding to the yield variable.
- yieldMethodFlag: The column name of the method flag corresponding to the yield variable.
- areaHarvestedValue: The column name of the area harvested variable.
- areaHarvestedObservationFlag: The column name of the observation flag corresponding to the area harvested variable.
- areaHarvestedMethodFlag: The column name of the method flag corresponding to the area harvested variable.
- yearValue: The column name for the year variable in data.
- byKey: The column name for the variable representing the splitting group. Usually, this is the country variable.
- removePriorImputation:
- removeConflictValues:
- imputedFlag:
- naFlag: How are missing values specified in the database? Usually, this is "M".

## Value

Returns a list of the default parameters used in the data pre-processing algorithm.

---

`ensureProductionInputs` *Ensure Production Inputs*

---

## Description

This function is designed to ensure that the provided dataset is valid. In particular, it coerces column types: all values are coerced to numeric ( instead of integer, which can cause problems) and all flags are coerced to character (instead of logical, which occurs if the flag is set to NA). Also, it ensures data is a data.table.

## Usage

```
ensureProductionInputs(data, processingParameters)
```

## Arguments

| | |
|---|---|
| `data` | A data.table containing the data. |
| `processingParameters` | A list containing the parameters to be used in the processing algorithms. See ?defaultProcessingParameters for a starting point. |

`getAllHistory` *Get All Production History*

## Description

This function is used to pull the historical production values from the database. One main purpose of this data is to train machine learning algorithms on invalid observations, and use those models to attempt to detect invalid observations in the current data.

## Usage

```
getAllHistory()
```

## Value

A data.table object with the following columns:

- geographicAreaM49: The code for the country of this observation.
- measuredElement: The code for the element of this observation (these match with the "Element" dimension in the agriculture domain, for example 5312 is area harvested, 5510 is production, etc.).
- measuredItemCPC: The CPC code for the commodity.
- timePointYears: The year for this observation.
- Version: ?
- StartDate: The initial date this value was entered into the system.
- EndDate: The date that this value was overwritten. If NA, this value is the current best estimate and is considered valid. If not NA, this value is considered invalid.
- Metadata: ?
- Metadata_Language: An abbreviation for the language of the metadata comments.
- Metadata_Group: ?
- Metadata_Element: ?
- Metadata_Value: ?
- Value: The value stored in the database for this observation.
- flagObservationStatus: The flag corresponding to the observation status of the current observation. See the faoswsFlag package for more details.
- flagMethod: The method flag for the current observation. Again, see the faoswsFlag package.
- valid: A Y/N value indicating whether or not the current value is valid. This is determined solely on the basis of whether or not this value was overwritten.
- productionValue: The value corresponding to production (element 5510) for this particular observation. This may be useful in training the classification ensemble.

`getProductionDomainData` *Get Production Domain Data*

## Description

This function is designed to pull production data from the working system. It is essentially a wrapper to the GetData function in faosws, but it massages the data from that function slightly.

## Usage

```
getProductionDomainData(key)
```

## Arguments

| | |
|---|---|
| `key` | A DatasetKey object, typically as created by GetTestEnvironment. See the argument with the same name in faosws::GetData. |

## Value

A data.table object containing the dataset of interest.

---

`imputeProductionDomain` *This function imputes the whole production domain.*

## Description

The function will impute production, area harvested and yield at the same time.

## Usage

```
imputeProductionDomain(data, processingParameters, yieldImputationParameters,
  productionImputationParameters)
```

## Arguments

| | |
|---|---|
| `data` | The data |
| `processingParameters` | A list of the parameters for the production processing algorithms. See defaultProductionParameters() for a starting point. |
| `yieldImputationParameters` | A list of the parameters for the yield imputation. See defaultImputationParameters() for a starting point. |
| `productionImputationParameters` | A list of the parameters for the production imputation. See defaultImputationParameters() for a starting point. |

## Details

Transformation in the yield formula is not allowed and will not be taken into account.

## okrapd *Example data for the documentations.*

### Description

The data containing global okra production, area harvested and yield from the year 1995 to 2013.

### Usage

```
data(okrapd)
```

### Format

A data.table object with 912 rows and 14 variables

## processProductionDomain *This is a wrapper for all the data manipulation step before the preparation of the imputation.*

### Description

This is a wrapper for all the data manipulation step before the preparation of the imputation.

### Usage

```
processProductionDomain(data, processingParameters)
```

### Arguments

| | |
|---|---|
| data | The data |
| processingParameters | A list of the parameters for the production processing algorithms. See defaultProductionParameters() for a starting point. |

## saveProductionData *Save Production Data*

### Description

This function takes the a seed data dataset and saves it back to the database.

### Usage

```
saveProductionData(data, areaHarvestedCode = "5312", yieldCode = "5421",
  productionCode = "5510")
```

## Arguments

| | |
|---|---|
| `data` | The data.table object containing the seed data to be written to the database. |

## Value

No R objects are returned, as this functions purpose is solely to write to the database.

# Index