**Package ‘faoswsLoss-package’**

9/20/2017

**Package:** faoswsLoss

**Type:** Package

**Title:** Package to perform the computation of food loss FAO food balance sheets

**Version:** 0.2

**Date:** 2017-09-22

Authors@R: c( person("Alicia", "English", role = "aut", email = “Alicia.english@fao.org", person( “Marco“ ,“Mingione”, role = "aut", email = “Alicia.english@fao.org"

**Description:** This package provides all the functions to perform computation of food loss percentages for the FAO food balance sheets. The first model was developed by

**Previous iterations:** Michael Kao and then modified by Natalia Golini and then replaced by the current version

**URL:** https://github.com/SWS-Methodology/faoswsLoss

**License:** FAO

Document:

[1. Information Flows 3](#_Toc501637825)

[2. SWS Dependencies 5](#_Toc501637826)

[3. External Inputs 5](#_Toc501637827)

[3.1. Loss Factors (Subnational/Stages) 5](#_Toc501637828)

[3.2. Input Data Tables 6](#_Toc501637829)

[3.3. Weights and commodity basket aggregation 7](#_Toc501637830)

[4. Routines 8](#_Toc501637831)

[4.1. Computational Parameters 8](#_Toc501637832)

[4.2. Input Parameters/Datasets 9](#_Toc501637833)

[finalModelData 9](#_Toc501637834)

[getProductionData 9](#_Toc501637835)

[getLossData 10](#_Toc501637836)

[4.3. Data Preparation 11](#_Toc501637837)

[addHeadingsCPC 11](#_Toc501637838)

[4.4. Modeling the loss percentages 11](#_Toc501637839)

[MultiExp 11](#_Toc501637840)

[FSC\_Markov 12](#_Toc501637841)

[VariablesAdd1 13](#_Toc501637842)

[LossModel 14](#_Toc501637843)

[WorldBankAPIData 15](#_Toc501637844)

[4.5. Calculating the GFLI/GFLP 15](#_Toc501637845)

[GFLI\_SDG 15](#_Toc501637846)

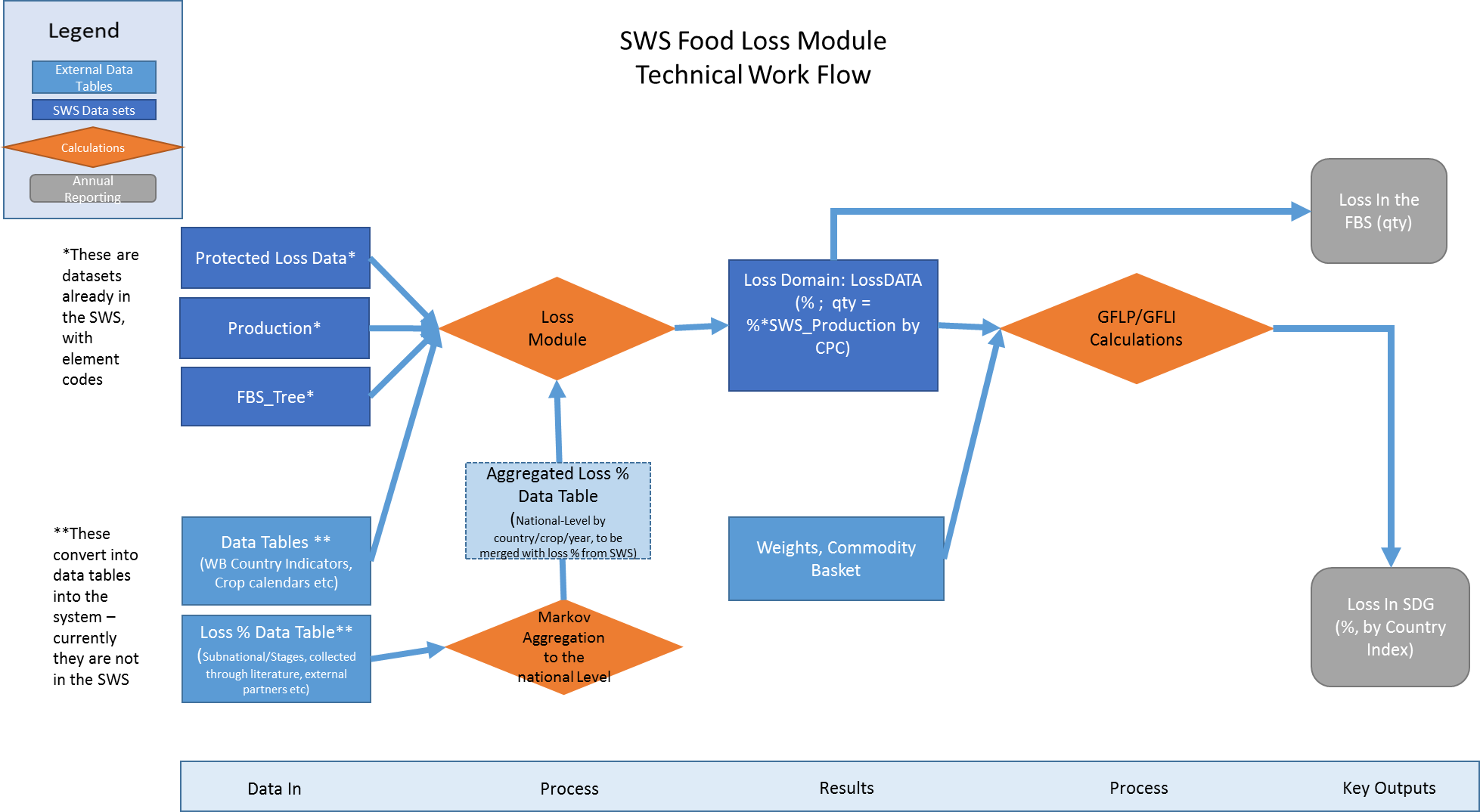
[4.6. Computational Parameters 15](#_Toc501637847)

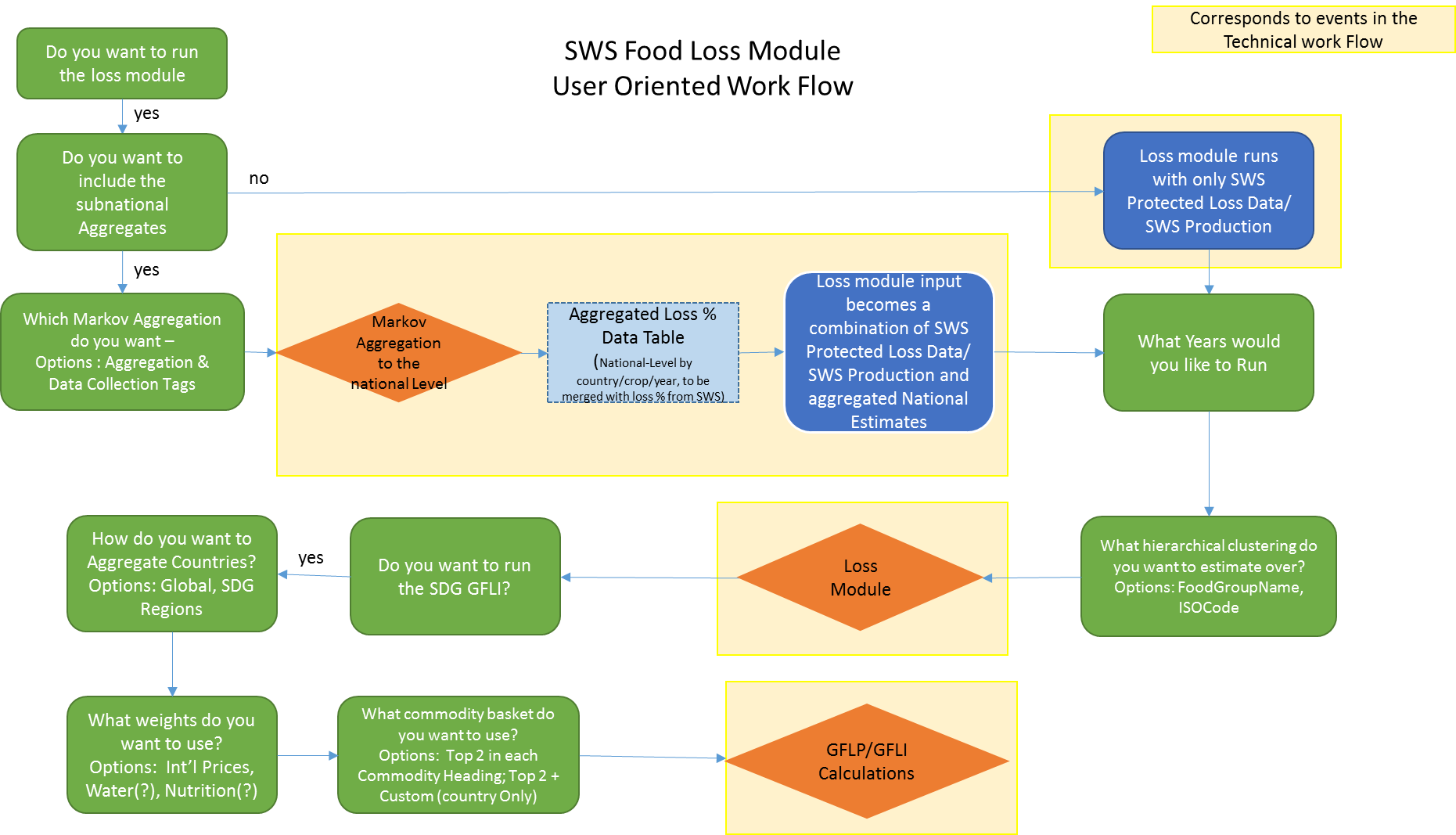
[GFLI\_SDG\_Fun 16](#_Toc501637848)

[5. Outputs 16](#_Toc501637849)

[6. Appendix 1. Variable Inputs 17](#_Toc501637850)

# Information Flows





# SWS Dependencies

faosws (>= 0.6.1)

swsDatasetDepends:

* agriculture:aproduction,
* agriculture:alosses (protected)
* fbsTree

swsModuleDepends:

* faoswsProduction,

swsModuleOutputs: LossDomain

* Percentages
* Quantities

The model will estimate losses for country, commodity and years combinations where it has existed in the SWS.

# External Inputs

## Loss Factors (Subnational/Stages)

The input for the dependent variables in the model is a combination of data from the SWS and the information collected from loss studies.

For the data coming from the external sources, the information. In the ***FLW\_LossPercFactors*** Excel file there are a couple of key columns that are required for merging the data with data in the model,

* Loss\_Per\_clean, This is the clean loss percentage. If for example a range is given for a study, the column calculates the average.
* Identifying: ISOCode, measuredItemCPC, Year
* Disaggregated by subnational stages along the Food Supply Chain (FSC\_Location)
  + The options that are used in the model are: farm, transport, storage, trader, wholesale, processing, retail, wholesupplychain, sws\_total (there should be no spaces)
  + The harvest and pre-harvest data collected will be useful for the next stages of modeling
  + This data is fed into the Markov Aggregation
* Tag\_DataCollection: This is to sort out data that is useful for the modeling efforts. For example, the rapid assessments, field and laboratory trials are useful for parameter estimation at the stages of the food supply chain but not for the aggregation at the national level – and are thusly excluded.

Additionally, there is metadata that has been collected along with the numbers – for example, sample size, region collected, reference and url, causes of losses, etc. The additional data allows for future modeling efforts, improvement to country level technical assistance in finding common causes across countries and so forth.

## Input Data Tables

There are 3 main sources of data for the model, these are collected at the national level by international partners, including,

* World Bank
* IFPRI
* IEA
* WTO
* World Bank/UN/IPCC

The description of all of the variables, including scope, timing of data collection and source which the data can be found, can be found in the supplemental table, including the original table names. These are the data inputs for modeling losses at the country level. The appendix will also be included as an additional excel file.

However, there are several that should be at gradually replaced by the common tables to be consistent across the organization. These are the tables are the ones that will be consistent within the SWS:

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Description | Data Folder | Source |
| CPC | CPC codes to create the set of data that needs to be included in the loss predictions | General | FAOSWS |
| FBSTable | The groupings of commodities into commodity groupings from the SWS | General | FAOSWS |
| Int\_$\_Prices\_2005 | Reference Prices for the index for all commodities for 2005 (2015 when available) | General | FAOSTAT |
| Crop Calendar | Covers crops and harvest months by country | General | FAO AIMS |

For data that is external to the organization but likely to have duplicates in other projects:

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Description | Data Folder | Source |
| Population | Population numbers by countries to impute per capita measures for aggregates | General |  |
| WB\_indicators\_Metadata | This is the listing of the World Bank Country Indicators that are available through the API. The API uses this list to download the necessary data | General | World Bank |
| WB\_UN\_CoutryCodes | This is the list of the WB country codes that are specific for the mapping between the UN and the World Bank. This is redundant to the mapping but a check to make sure to have all data needed | General | World Bank |
| SDG Country Mapping for the SDG/MDG/ISO3/M49 | Maps the countries to their respective geographic and SDG monitoring groups | General | FAO/WB |
| Rainfall | *Requirement:* The data needs to be monthly averages at the country level  This gridded historical dataset is derived from observational data, and provides quality controlled temperature and rainfall values from thousands of weather stations worldwide, as well as derivative products including monthly climatologies and long term historical climatologies. The dataset is produced by the Climatic Research Unit (CRU) of University of East Anglia (UEA), and reformatted by International Water Management Institute (IWMI). CRU-(Gridded Product). CRU data can be mapped to show the baseline climate and seasonality by month, for specific years, and for rainfall and temperature.; | General | World Bank/UN/IPCC/ |
| Temperature | *Requirement:* The data needs to be monthly averages at the country level  This gridded historical dataset is derived from observational data, and provides quality controlled temperature and rainfall values from thousands of weather stations worldwide, as well as derivative products including monthly climatologies and long term historical climatologies. The dataset is produced by the Climatic Research Unit (CRU) of University of East Anglia (UEA), and reformatted by International Water Management Institute (IWMI). CRU-(Gridded Product). CRU data can be mapped to show the baseline climate and seasonality by month, for specific years, and for rainfall and temperature. | General | World Bank/UN/IPCC/ |

## Weights and commodity basket aggregation

For the SDG Global Food Loss Index and aggregated percentages (GFLI/GLFP), there are a couple of additional inputs that are needed. These are

* The international Prices by commodities, coming from FAOSTAT
* Additional weighting schemes
* Population numbers by country

For the SDG monitoring, the international prices are applied to the SWS production data and then within the code, sorted for the highest value, and the top 2 for each of the commodity groupings is chosen for the compiling of the index.

For other baskets (e.g. based on calories or environmental impact), the prices can be switched out and the routine will run the same process as for the SDG reporting. The purpose of doing it internally in the code is to minimize the intermediate tables in the system calculations that would need to be updated.

# Routines

## Computational Parameters

These are options for the person running the loss module, these are defined at the beginning and impact functions in the rest of the module

*Binary*

* updateModel: Whether or not this is a model run, which will rewrite all the data in the loss module
* LocalRun: The model can either run in the SWS or from the github directory if the faoswsloss directory has been forked appropriately.
* SubNationalEstimates: is an option from the Information flows, allows for the user to decide whether or not to use and aggregate up the supply chain the literature review estimates. If 0, then only the protected loss estimates from the SWS are considered.
* graphLoss: Whether or not to output the graphs of the loss estimates by country

*Numeric (min,Max)*

* selectedYear = as.character(1991:2015): sets the years for the data pulls from the SWS
* selectedModelYear = as.character(1961:2015): Sets the years for the data to use from the literature reviews

*Predefined Strings*

* DataCollectionTags\_all <- c("SWS","APHLIS","Rapid Assessment","Expert Opinion",
* "Laboratory Trials","Field Trial","Survey","Declarative","Crop-Cutting","Case study")
* DataCollectionTags\_represent <- c("SWS","APHLIS","Expert Opinion","Survey","Declarative")
* ExternalDataOpt <- DataCollectionTags\_represent

These three options take the loss data from the data table in the SWS that contains the loss estimates from the literature and other studies. The \_all option has all of the “types” of data that were collected, and are based on the Global strategy guidelines. The representative data are the ones that the model will use to aggregate. *ExternalDataOpt,* given that there are different options for what the analyst may consider relevant, this can be set to the \_all or \_represent, as to not change any other code

* MarkovOpt <- "aveatFSP" # "model": For aggregating the subnational using the markov function, there are two options, at the moment the averaging as described in the function is "aveatFSP" , but in the case where a model may estimate losses probabilistically in the future, there is space and an option "model".
* HierarchicalCluster <- "foodgroupname" # "isocode", "SDG.Regions"

These are the clustering options for estimating the loss model. The foodgroupname uses the FBSTree to cluster the data based on it’s standard hierarchy. Other options have included the individual country and the sdg regional grouping. The country parameter may be adjusted in future iterations to model at the country level.

* VaribleSelection <- "RandomForest\_geo" # “Bayes”

In the present iteration of the model, the variables are selected through the Random Forest Algorithm. In the near future, the second option will be the Bayesian updating model.

## Input Parameters/Datasets

### finalModelData

**Description**

When the model updates the parameters for estimation, the finalModelData, pulls in the production dataset, the loss dataset and creates the data table of all the factors that will need to be estimated and written back into the system.

The ***keys*** for the model are "geographicaream49", "timepointyears", "measureditemcpc". Given the workings of the datatables from the SWS, these are in all lower case, as is most of the variables in the module.

**Usage**

For updating the entire dataset on losses

**Arguments**

none

**Returns**

timeSeriesDataToBeImputed , which is the full estimation dataset for the model.

Losses are also only estimated for the primary commodities, not the derived commodities or any other utilizations. And the matrix is formed based on the data from the production dataset that is pulled within this section of the model.

### getProductionData

**Description**

Pulls the agricultural production from the SWS. The only keys are "5510", and uses the line to access the production dataset faoswsUtil::getCompleteImputationKey(table = "production").

There are no restrictions on the production.

**Usage**

getProductionData ()

**Arguments**

none

**Returns**

Production data from the agricultural production dataset in the SWS

### getLossData

**Description**

Pulls the losses from the agricultural production from the SWS. The only keys are "5016", and uses the line to access the production dataset

DatasetKey(

domain = "agriculture",

dataset = "aproduction",

dimensions = list(

Dimension(name = "geographicAreaM49",

keys = GetCodeList(domain = "agriculture",

dataset = "aproduction",

dimension = "geographicAreaM49")[type == "country", code]),

Dimension(name = "measuredElement", keys = c("5016")), #"5126"

Dimension(name = "timePointYears", keys = as.character(1990:2015)),

Dimension(name = "measuredItemCPC",

keys = GetCodeList(domain = "agriculture",

dataset = "aproduction",

dimension = "measuredItemCPC")[, code]))

).

**Usage**

getLossData (Protected = TRUE)

**Arguments**

none

**Returns**

Protected Loss data from the agricultural production dataset in the SWS

Both of the results from these functions are merged by the ***keys*** and the protected losses are divided by agricultural production. This was an assumption that was discussed at length in the development process, that losses as they mostly occur are at the farmgate and transportation and storage, most imports directly go to processing and any losses at this stage will be captured in the conversion factors. The code is the following,

lossData <- merge(production,lossProtected, by.x = keys\_lower, by.y = keys\_lower, all.y= TRUE)

lossData[, loss\_per\_clean := 100\*(value\_measuredelement\_5016/value\_measuredelement\_5510)]

lossData[, fsc\_location := "SWS"]

## Data Preparation

### addHeadingsCPC

**Description**

Adds the appropriate number of zeros to the CPC code

**Usage**

addHeadingsCPC (measuredItemCPC)

**Arguments**

measuredItemCPC: The CPC codes that don’t have leading zeros

**Returns**

addHeadingsCPC

**Examples**

## Modeling the loss percentages

### MultiExp

**Description**

Allows the interaction of the varaible terms (X) to the specified degree, the function also sorts the data by the type of the variable, so that only the numerical data (and not factors) are interacted.

Necessary for the Random Forest to test the specification against non-linearities in the data, though to expansive data wise for the general data set. Use for the top variables. e.g. X1^1\*X2^0, X1^2\*X2^1. This was used in the testing phase for testing the nonlinear biases of the RandomForest. Given the memory constraints of running the model in full, this isn’t included in the final SWS model, though the code remains for future testing.

**Usage**

MultiExp (X, degree,depVar)

**Arguments**

X: Matrix of variables

Degree: the degree in power terms of the interaction

**Returns**

The expanded set of the multidimensional variables for the model

### FSC\_Markov

**Description**

Collapses the stages of the loss percentages along the subnational supply chains into singular observations at the country level for a given year. The options for this function are the simplified (aveatFSP) aggregation as described in the methodology. Under this heading, as the data progresses, additional options for probabilistic expansion can be considered.

**Usage**

FSC\_Markov (RawData,opt)

**Arguments**

RawData: Matrix of Conversion Factors

opt: Options for aggregating up to the national level for each country/crop/year

opt = aveatFSP

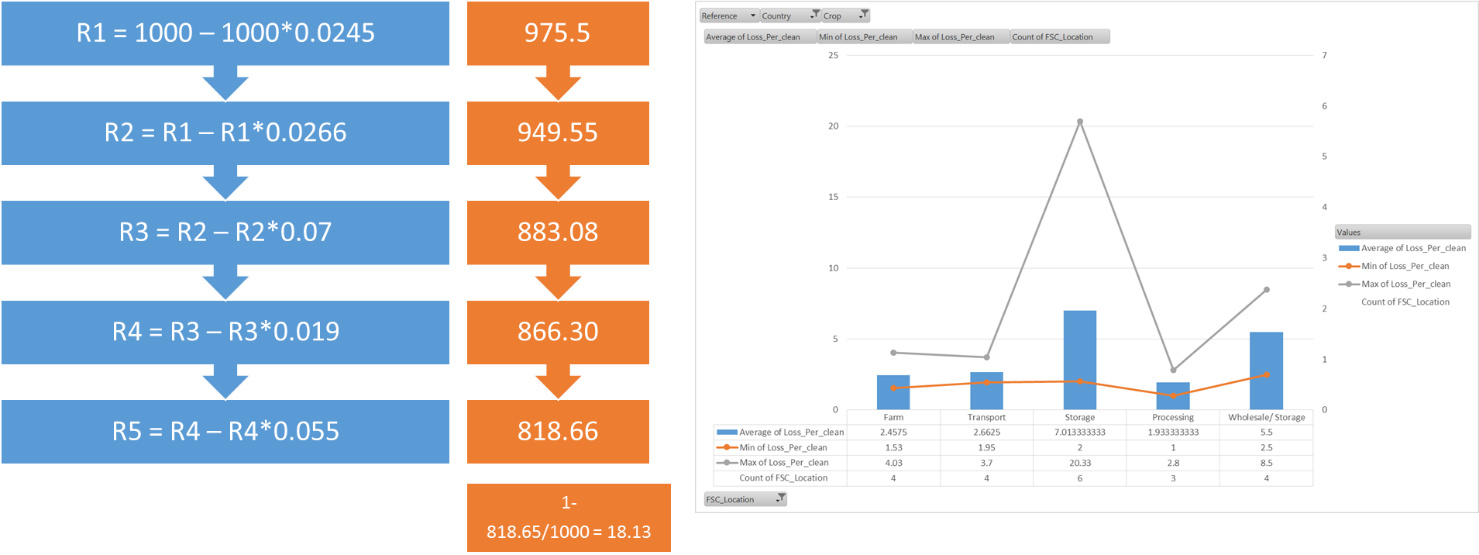
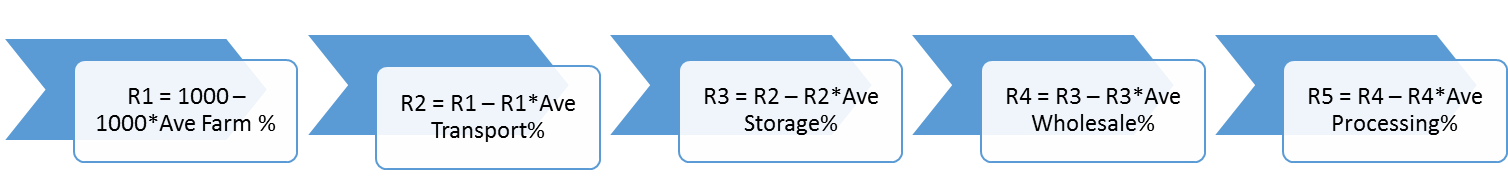
This option averages at each supply chain point for each country,crop,year - delineated by ("farm","storage", "processing", "retail","trader", "transport", "wholesale","whole supply chain", "sws\_total")

From the FLW\_LossPercFactors.xls the column FSC\_Location is the column that is to delineate the stages. If there are datapoints that cover multiple stages, the function splits it at the ‘/’ and attributes it to the first stage provided.

**Returns**

For each country, commodity and year, the function returns a single percentage for use in the loss modeling.

**Examples**



### VariablesAdd1

**Description**

Adds the factors for estimation to the loss factors, these are factors that are included from the literature. As well as other factors that are considered proxies for losses. This function merges the data tables to the loss estimates, by country, crop and year where relevant.

There are three stages to this function, which merge the data for the training or prediction sets with the data that has been assumed to be correlated with losses. The first step takes the crop calendar that is available and merges it with rainfall and temperature data, by the starting month of the harvest season by country. Since losses are at the country level, the rainfall/temperature parameter is averaged over the month and over the country.

The second stage of the function merges the data with the year specific variables. At the moment these country-time variables consist of the world bank pink sheets.

The third stage merges the country and year specific variables. The *LossTablelist\_ctryYr* is the list of datatables that need to be pulled from the working system. The merge is built first on the year/ctry combination and then loops through a merge of this larger table by three lagged years.

**Usage**

VariablesAdd1 (DataUseInt,keys\_lower,Predvar2)

It can be found in the main run file as well as in the loss module. For the Predictive set it only merges the data that was used in the model for the cluster (and not all variables).

**Arguments**

DataUseInt: Matrix of Loss percentages which is either data\_use\_train or data\_use\_pred

keys\_lower: "geographicaream49", "timepointyears", "measureditemcpc"

Predvar2: a list of blank, if being used for the training set, or the list of used variables for the predicitive set

**Returns**

The full data matrix for estimation

### LossModel

**Description**

The model operates in 3 parts,

1. Sets the clusters for estimating for countries without data. And it log transforming the data and setting the bounds just above 0 and below 1.
2. Random Forest variable selection. Runs the model across countries within the cluster to find the top preforming variables This has the option for using the Bayesian updating as a future addition.
3. Heirarchical model. Models the loss percentages on the results of the random forests variable selection and then applies them to the country/commodities/years needed in the output. This cluster uses the FBSTree id3 as the cluster that the model segments its runs on.
4. Model Saves. The data after estimation, takes the parameters for the country and crop dummy variables and for the coefficients of the model. It calculates the dummy variables first for all of the predictive set and sets the flags as I and e for the estimated data. This smaller set of the total set needed predictive values is then merged with the large set (timeSeriesDataToBeImputed), where protected values are excluded.

**Usage**

LossModel (Data,timeSeriesDataToBeImputed,lossData,HierarchicalCluster,keys\_lower)

**Arguments**

* Data is the data used for training the indicator. This should be the final data set, of the loss percentages by country, with data aggregated in the Markov model and with explanatory variables added.
* keys\_lower: "geographicaream49", "timepointyears", "measureditemcpc"
* timeSeriesDataToBeImputed is the data that needs estimates predicted (finalPredictData)
* HierarchicalCluster is for the group/cluster ("foodGroupName" was the best preformer)
  + Options: ("foodGroupName", "foodPerishableGroup","SDG.Regions",”ISOCode”)

**Returns**

For each country, commodity and year, the function returns the data series for the required crops in percentage terms for use in the FBS and the SDG reporting.

**Examples**

### WorldBankAPIData

**Description**

Can be used to add the variables from an external data API for the World Bank. Future iterations may have this data integrated already into the SWS.

**Usage**

WorldBankAPIData(LossPercentages)

**Arguments**

LossPercentages: Matrix of Loss percentages

**Returns**

The full data matrix for estimation

## Calculating the GFLI/GFLP

### GFLI\_SDG

**Description**

Is the main file for calculating the Global Food Loss Index and producing the required reporting requirements. It contains only the function for calculating the Global Food Loss index and the Country indices.

## Computational Parameters

These are options for the person running the loss module, these are defined at the beginning and impact functions in the rest of the module

*Binary*

GFLI\_calc : Calculate or not the new Index

*Predefined Strings*

* aggregation <- "geographicAreaM49" #"sdg\_region", "geographicAreaM49":

This is the aggregation for the regional index, it can either be calculated for just the individual countries or for the SDG regional aggregates.

* weights <- "intl\_prices"

These are the weights for the index. Currently, the weights are the international dollar prices for available commodities across the globe in 2005. As the prices are updated for 2015, the weights will be updated.

* basketN <- "top2perhead\_byCtry" # "top2perhead\_Globatop10","top2\_calories"
* This is how the basket of commodities are selected for the indexes. The "top2perhead\_byCtry" chooses the top 2 commodities in the FBSTree commodities groupings below. The "top2perhead\_Globatop10", chooses the top two by commodity groupings, but across the globe. The "top2\_calories", sorts the commodities by caloric impact.

#SDG Headings

fbsTree[foodgroupname %in% c(2905,2911), GFLI\_Basket :='Cereals & Pulses',]

fbsTree[foodgroupname %in% c(2919,2918), GFLI\_Basket :='Fruits & Vegetables',]

fbsTree[foodgroupname %in% c(2907,2913), GFLI\_Basket :='Roots, Tubers & Oil-Bearing Crops',]

fbsTree[foodgroupname %in% c(2914,2908,2909,2912,2922,2923), GFLI\_Basket :='Other',]

fbsTree[foodgroupname %in% c(2943, 2946,2945,2949,2948), GFLI\_Basket :='Animals Products & Fish and fish products',] # |foodGroupName == "PRODUCTS FROM FISH",

### GFLI\_SDG\_Fun

**Description**

Calculates the Global Food Loss Index and the country/regional index, with the parameters specified

**Usage**

GFLI\_SDG\_fun(keys\_lower,aggregation,weights,basketN,production)

**Arguments (see above)**

keys\_lower

aggregation

weights

basket

production

**Returns**

GFLI & FLI

# Outputs

See Attached files

# Appendix 1. Variable Inputs