faoswsFlag:A package to perform flag aggregation and much more

Michael. C. J. Kao
Food and Agriculture Organization
of the United Nations

Abstract

This short documentation is intended to explain how observation flags are aggregated in the ESS Statistical Working System.

The methodology and tools are presented step by step, with code examples and explanations.

The paper also provide example of potential applications for integrating flag information.

Keywords: meta data, flag aggregation.

1. Introduction

Lets start by loading the required library into R.

```
## Load the required libraries
library(faoswsFlag)
library(ggplot2)
library(splines)
```

Since the introduction of the new statistical working system, the old symbol which reprents the collection and computation method of the data is now represented by two separate flags.

The first, an observation flag which is a description of the observation status, whether it be official, estimates or imputed value. While on the other hand, the methodology flag contains information of how it is collected or computed. It can be from survey, questionnaire or it can be obtained as a balance or estimated through statistical methodology.

The aim of this paper is to introduce a systematic way to aggregate observation flags and further incorporate this piece of information in to subsequent analysis.

Shown below is the corresponding table for the observation flags as of September 17, 2014 outlined in Annex 5 of FAO Statistical Standards:

The remaining of the paper is divided into three sections. First, we show how to crate a table which holds the rank information required for aggregation. The second part will illustrate how multiple flags can be aggregated by using functions provided in the package. Finally, the last section will present a simulated case how the use of these data can help us build better models.

2. Specification of a Mapping Table

In order to compute aggregation of flags, one must convert the symbol into a numerical type.

Table 1: Description of the Observation flag

Flags	Description
<blank>	Official Figure
E	Estimates
I	Imputed
M	Missing
T	Unofficial figure

The way this should be handled is assign a value based on the amount of information quantity and the reliability of the observation status.

Data obtained from reliable source should have a high information content and thus should be assigned a high value, while data based on human estimation should be assigned a low level of score to reflect that the data is not directly observed and error may result as a case.

Shown below is the default weights table for the new statistical working system.

```
## Printed here is the default flag conversion table shipped with the
## package.
faoswsFlagTable

##   flagObservationStatus flagObservationWeights
## 1                                         1.00
## 2                     T                   0.80
## 3                     E                   0.75
## 4                     I                   0.50
## 5                     M                   0.00
```

From this table, we have assigned 1 to official figure while 0 to missing values. Albeit the arbitrary selection of the values, it provides a rank of the information content which is necessary for the computation of flag aggregation.

Flag tables can be created for each separate application depending on the goal. For the aggregation of flag, only one restriction is applied and that is the value of the weights are unique.

3. Compute Flag Aggregation

In this section, we take the computation of yield for example and illustrate how to compute flag aggregation with the package.

The value of yield is computed based on production and area harvested which may come from different sources. Thus, when we compute a derived statistic which is unobserved such as the yield; it is important that the information quality reflect the lowest level that is used in the computation. For a set of aggregation, the minimum of the set is taken as the final observation flag.

A more concrete example is to say that we may have a production value recorded from official survey (), while the area harvested was collected from an unofficial external data base (T). Following this principle, the resulting flag for yield should return (T) to reflect the lower information content of the unofficial figure.

```
## The function works just like sum(), with an optional arguement for
## the flag table to be used.
aggregateObservationFlag("", "T", flagTable = faoswsFlagTable)

## [1] "T"
```

```
## Aggregation of multiple flag

## Simulate flag for production
simulatedProductionFlag =
    faoswsFlagTable[sample(1:NROW(faoswsFlagTable), 10, replace = TRUE),
                    "flagObservationStatus"]
simulatedProductionFlag

##  [1] ""  "T" "T" "I" "E" "M" "T" "T" "M" "I"

## Simulated flag for area harvested
simulatedAreaFlag =
    faoswsFlagTable[sample(1:NROW(faoswsFlagTable), 10, replace = TRUE),
                    "flagObservationStatus"]
simulatedAreaFlag

##  [1] "I" "E" "M" "T" ""  "M" "T" "E" "T" "E"

## Now compute the aggregation of flag
aggregateObservationFlag(simulatedProductionFlag, simulatedAreaFlag,
                         flagTable = faoswsFlagTable)

##  [1] "I" "E" "M" "I" "E" "M" "T" "E" "M" "I"
```

Currently, the weights of the flags are chosen as arbitrary mainly to preserve a rank order based on expert judgement. Nevertheless, this information can be estimated from the data and history of the flag as we wil discuss more in the improvement section.

4. Other Applications

The converesion of the symbol to a numeric value has various advantage than solely for the purpose of constructing aggregation. It can assist subsequent modelling by identifying the quality of data and enable an algorithm to take into account of the difference among various data source.

For example, instead of fitting a linear regression by treating all observation equally with the same source and identical quality, we can estimate a weighted regression which gives more weight to data which are of higher reliability.

4.1. Robust Fitting to Anormalies

The following artificial example illustrates how accounting for the information source can result in a better fit and incorporate poor data quality. The artificial data starts in 1991 and ends in 2014, with all the observation collected as unofficial figure except the last two which

were estimated. For illustrative purpose, the values were estimated by a poor algorithm and can be seen in the graph as anormalies. The illustration shows how accounting for the anormalies through the use of meta data can result in more robust model fitting than as treating all data have the same information quality.

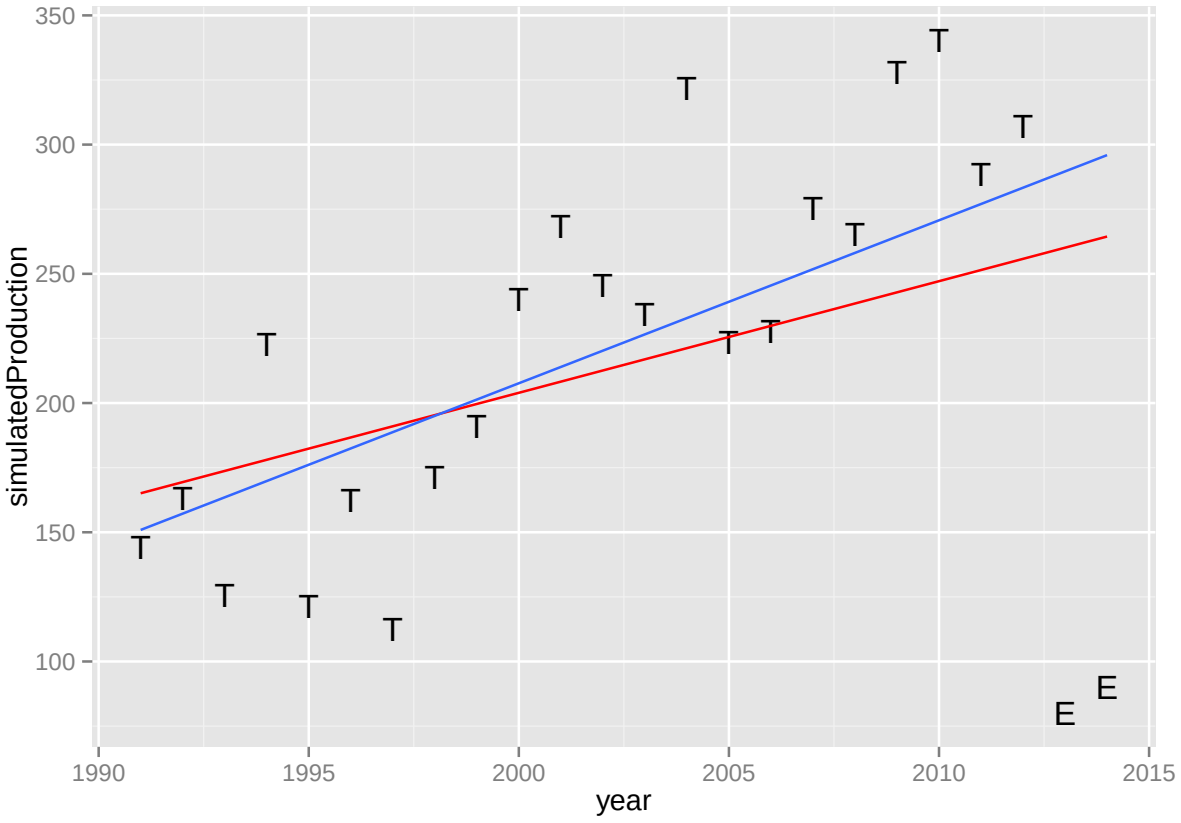
Figure below shows the value of simulated production with respect to time, they are labelled by their corresponding flag. The fit of the linear regression when all observation are treated equally is illustrated in red. On the other hand, the blue line corresponds to the fit of a weighted regression which gave less weight for the suspicious point as it was marked as estimated (E) by the flag and takes only half the weight of an official observation.

The dataset contains two flag, E and T which has weight of 0.4 and 0.8 respectively.

```
## New table for simulation.
simTable = faoswsFlagTable
simTable[simTable$flagObservationStatus == "E",
         "flagObservationWeights"] = 0.4

## Simuate a data set which has a single point that was imputed badly
## but still used for later analysis.
x = 1991:2014
y = 100 + 10 * (x - 1989) + rnorm(length(x), sd = 30)
f = rep("T", length(x))
y[23:24] = c(80, 90)
f[23:24] = "E"
simulated.df = data.frame(year = x, simulatedProduction = y, flag = f)

## Plot the data and show the two different fit when accounting for the
## source and quality of information.
ggplot(data = simulated.df,
       aes(x = year, y = simulatedProduction, label = flag)) +
    geom_text() +
    geom_smooth(method = "lm", formula = y ~ x,
                data = simulated.df, se = FALSE, col = "red") +
    geom_smooth(method = "lm", formula = y ~ x,
                aes(weight = flag2weight(flag, flagTable = simTable)),
                data = simulated.df, se = FALSE)
```



4.2. Weighted Source of Combination

Another potential application of weight is for combining data from various source to form an ensemble estimate.

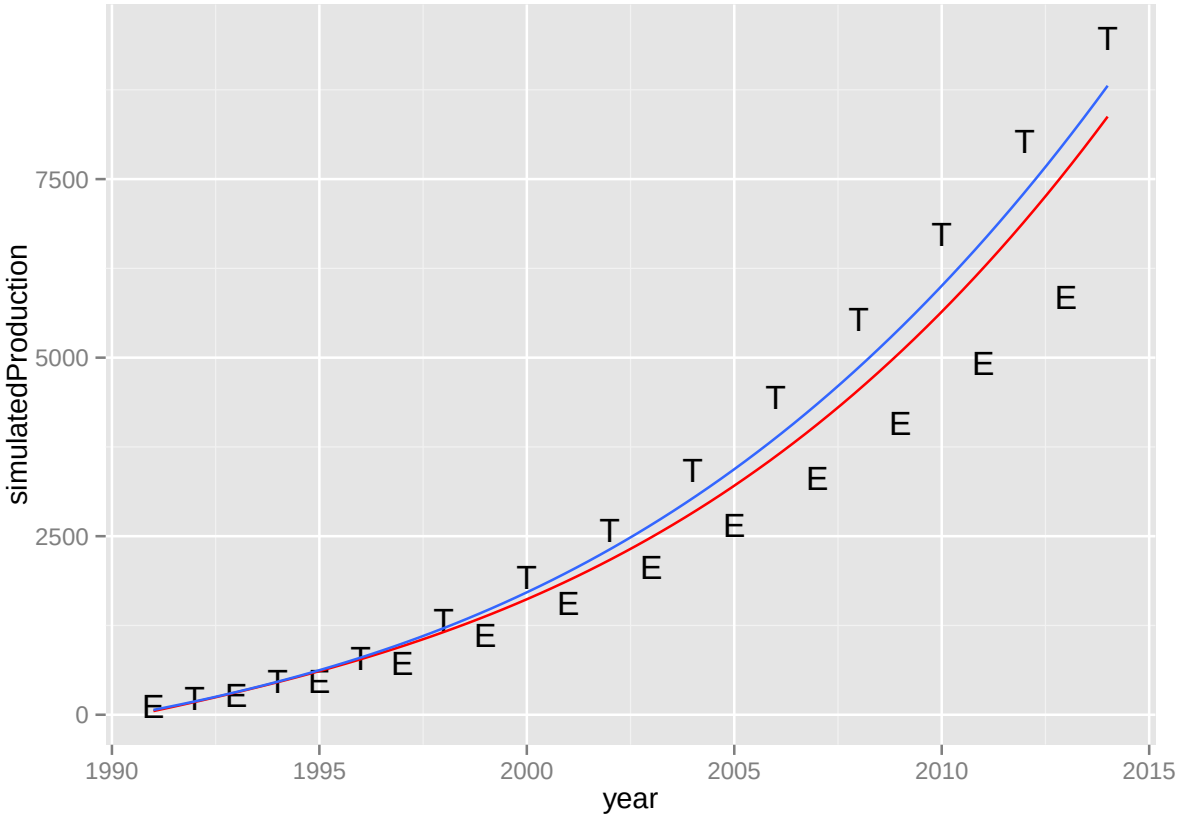
Here we generate another artificial dataset for illustration. Assuming we have two sources of data where each are collected on alternating years and we would like to estimate the growth rate.

Plotted below is the simulated data, again the red is uniform weight while the blue line represents the model which accounts for the asymmetry of information. Since we trust data which are marked with T with higher degree of believability, we can observe the estimated growth curve is closer to the observed value marked as T.

```
x = 1991:2014
y = 100 + c(10, 15) * (x - 1989)^2 + rnorm(length(x), sd = 20)
f = rep(c("E", "T"), length(x)/2)
simulated.df = data.frame(year = x, simulatedProduction = y, flag = f)

## Plot the data and show the two different fit when accounting for the
## source and quality of information.
ggplot(data = simulated.df,
       aes(x = year, y = simulatedProduction, label = flag)) +
    geom_text() +
    geom_smooth(method = "lm", formula = y ~ bs(x),
                data = simulated.df, se = FALSE, col = "red") +
    geom_smooth(method = "lm", formula = y ~ bs(x),
                aes(weight = flag2weight(flag, flagTable = simTable)),
```

```
                data = simulated.df, se = FALSE)
```



5. Computing Weights

To compute the weights objectively, we following the principle of minimum discrimination information.

The information states that given derived information set, a new distribution q should be chosen which is as hard to discriminate from the original distribution p as possible; so that the new data produces as small an information gain as possible.

In another word, if we have to choose another representation, the information set which result in the least amount of information gain or uncertainty should be chosen.

First of all, we take official figures as the desired distribution. Then we can measured the information gain when we replace data collection with imputed or estimated data.

The cross-entropy can be calculated as:

$$\mathrm{H}(P, Q) = \mathrm{H}(P) + D_{\mathrm{KL}}(P \| Q)$$

Where

$$H(P) = -\sum_i p(x_i) \log p(x_i),$$

$$D_{\mathrm{KL}}(P\|Q) = \sum_i \log\left(\frac{p(i)}{q(i)}\right) p(i).$$

However, since the entropy of $H(x)$ would be the identical. We can simple calculate the Kullback-Leibler Divergence $D_{\mathrm{KL}}(P\|Q)$. After calculating the Kullback-Leibler Divergence, we can compute the weight according to the information gain.

$$\omega_i = \begin{cases} 1/(1 + D_{\mathrm{KL}}(P\|Q_i)) & \text{if } D_{\mathrm{KL}}(P\|Q_i) \neq 0 \\ 1 - 1e^{-5} & \text{if } D_{\mathrm{KL}}(P\|Q_i) = 0 \end{cases}$$

The following function shows how the weights can be computed from historical information. This code is only executable when connected to the intranet.

```
## load the library
library(faosws)
library(faoswsExtra)
library(faoswsFlag)
library(data.table)
library(FAOSTAT)

## Set up the data query
newPivot = c(
    Pivoting(code= "geographicAreaM49", ascending = TRUE),
    Pivoting(code= "measuredItemCPC", ascending = TRUE),
    Pivoting(code = "timePointYears", ascending = FALSE),
    Pivoting(code= "measuredElement", ascending = TRUE)
    )

newKey = swsContext.datasets

getAllCountryCode = function(){
    ## 1062 is geographical world
    keyTree =
        unique(GetCodeTree(domain = swsContext.datasets[[1]]@domain,
                           dataset = swsContext.datasets[[1]]@dataset,
                           dimension = "geographicAreaM49",
                           roots = "1062")
               )
    allCountryCode =
        unique(adjacent2edge(keyTree)$children)
    allCountryCode[allCountryCode %in% FAOcountryProfile$UN_CODE]
}

## Create new key and download data, the history is for the whole
## world since 1970 for wheat.
newKey[[1]]@dimensions$geographicAreaM49@keys = getAllCountryCode()
newKey[[1]]@dimensions$timePointYears@keys = as.character(1970:2013)
```

```
newKey[[1]]@dimensions$measuredItemCPC@keys = "0111"

## Compute the table
history = GetHistory(newKey[[1]], newPivot)
history[, timePointYears := as.numeric(timePointYears)]
obsTable = computeFlagWeight(history, method = "entropy")
```

Affiliation:

Michael. C. J. Kao
Economics and Social Statistics Division (ESS)
Economic and Social Development Department (ES)
Food and Agriculture Organization of the United Nations (FAO)
Viale delle Terme di Caracalla 00153 Rome, Italy
E-mail: michael.kao@fao.org
URL: https://github.com/mkao006/sws_r_api/tree/master/faoswsFlag