Seed module

Francesca

October 20, 2016

Introduction

How to use this document

This document is intended as a guide to understand more in detail the approach used in the seed module. In describes the most updated version of the module and it provides to the reader the state of the art in terms of methodology reflected into functionalities already implemented though R routines.

In the following we explicitly make reference to R programming language in order to guide the reader also in a deepper understanding of the functions stored in the Seed module repository link.

Setting up variables:

The model

As mentioned above, FAO collects data for seed and area sown through the FAO production questionnaire. However, while overall response rates to the questionnaire have been rising, not all countries provide estimates for all commodities. Where no official seed use information is available, seed use can be imputed, including by national FBS compilers. In practice, the necessary steps are:

- Impute area sown, when missing.
- Estimate seed use through a hierarchical linear model (if official data collected thaks to questionnaire is unavailable).

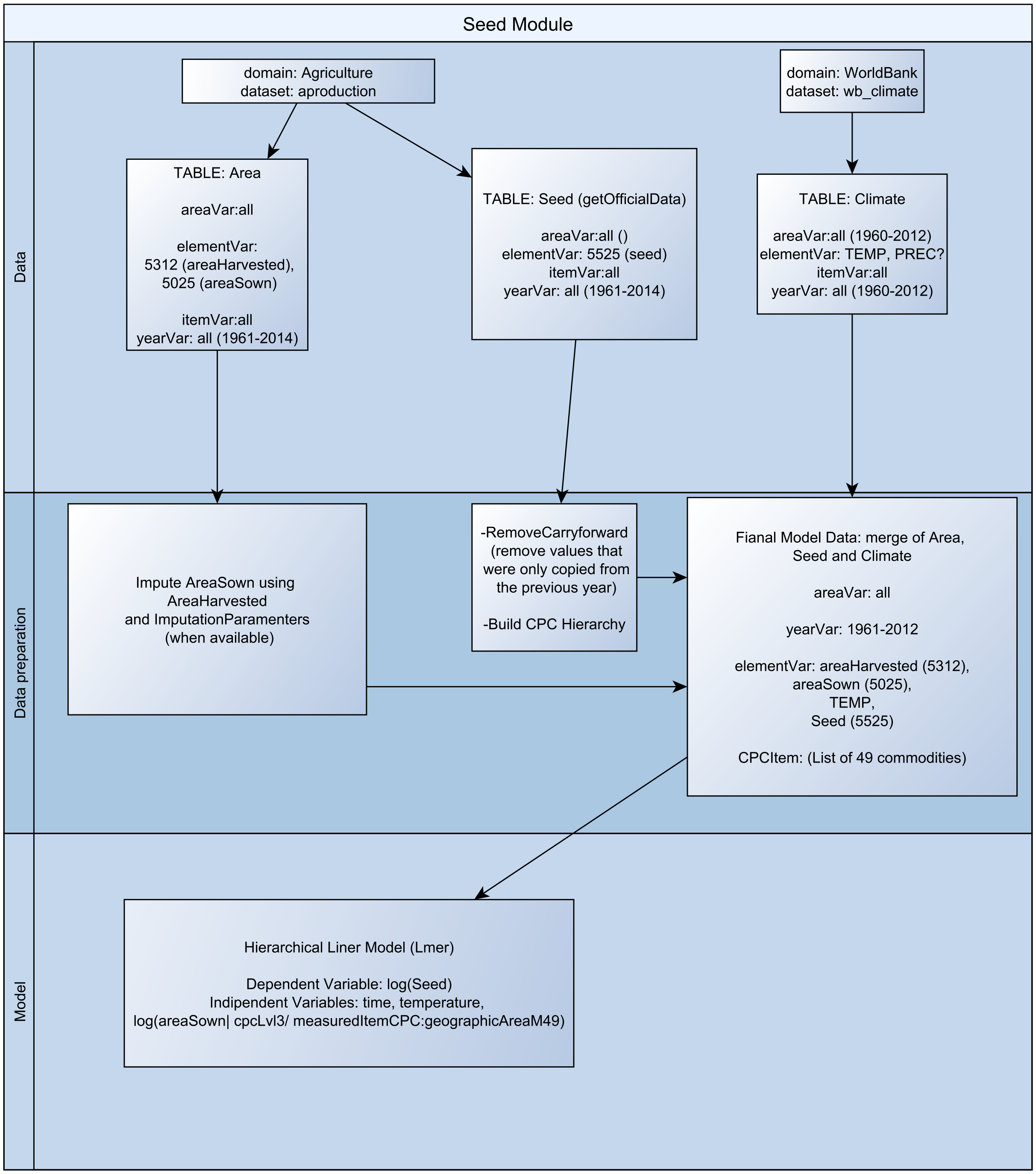
The estimation of seed rates is performed via a hierarchical linear model. The rationale for this model is that it is capable of capturing and modelling complicated trends when data is available. Moreover, the hierarchy of the model allows accurate imputation on countries with very sparse data by pooling together global data. The mathematical model can be written as follows:

$$log(Se_{i,j,k}) = \beta_0 + \beta_1 Temp_i + \beta_2 Time + \beta_{3,j,k}(AreaSown_{i,j,k} + \epsilon_{i,j,k}$$

$$\beta_{3,j,k} = \gamma_{3,0} + \gamma_{3,1,j,k}(CPC : Country)_{j,k} + \delta_{j,k}$$

$$\gamma_{3,1,j} = k_{3,1,0} + k_{3,1,1}(CPC)_j + \phi_j$$

Thus, the model estimates seed use proportional to the area sown. The model also accounts for changes over time and differences across countries; the latter are captured by the annual temperature variable, assuming essentially that seed rates need to be higher where production conditions are difficult, with a potential of late and frequent frosts (Russia), and can be lower where production conditions are more favourable (UK).



The flowchart highlights all the data (with ther relative sources) necessay to performed the imputation based on the hierarchical linear model. In particular we need

- Official seed data (response variable)
- AreaSown data to be derived from areaHarvested (covariate)
- Climate data to take into account the temperature to catch the country specificity (covariate)

Seed data

Import Official seed data:

```
seed = getOfficialSeedData()
head(seed)
```

```
##    geographicAreaM49 measuredItemCPC timePointYears
## 1:               100            0111           2008
## 2:               100            0111           2009
## 3:               100            0118           1980
## 4:               100            0118           1981
## 5:               100            0118           1982
## 6:               100            0118           1983
##    Value_measuredElement_5525 flagObservationStatus_measuredElement_5525
## 1:                     313000
## 2:                     282000
## 3:                          0
## 4:                          0
## 5:                          0
## 6:                          0
##    flagMethod_measuredElement_5525
## 1:                               -
## 2:                               -
## 3:                               -
## 4:                               -
## 5:                               -
## 6:                               -
```

Once seed has been pulled from the DataBase, the R module includes many functions from the Ensure package in order to performe a preliminary data validation. The checks are mainly addressed to:

1) ensure that the pulled data are in realible ranges: seed (5525) cannot be negative.

```
  ensureValueRange(data = seed,
                   ensureColumn = "Value_measuredElement_5525",
                   min = 0,
                   max = Inf,
                   includeEndPoint = TRUE,
                   returnData = FALSE,
                   getInvalidData = FALSE)
```

```
## All values withing specified range
```

2) ensure flag validity: this check might be unecessary because the getOfficialSeedData function embeds infomation in flags to be pulled:

```
   protectedFlag <- flagValidTable[flagValidTable$Protected == TRUE,] %>%
    .[, flagCombination := paste(flagObservationStatus, flagMethod, sep = ";")]
```

```
## All flag are valid
```

```
##Ensure   CorrectMissingValue not necessary because I pull only official data whose ObservationFlag

ensureCorrectMissingValue(data = seed,
                          valueVar = "Value_measuredElement_5525",
                          flagObservationStatusVar = "flagObservationStatus_measuredElement_5525",
                          missingObservationFlag = "M",
                          returnData = FALSE,
                          getInvalidData = FALSE)
```

```
## Data contains no mis-specified missing value
```

The seed data are then cleaned in order to:

- remove previous estimations besed on the carry forward procedure
- add three additional columns the CPC hierachy (the need of this passage will be clear when we build the model)

```
seed = removeCarryForward(data = seed,
                          variable = "Value_measuredElement_5525")

seed = buildCPCHierarchy(data = seed, cpcItemVar = itemVar, levels = 3)
```

Area data

Two additional variables are requested in order to build the model. Seed use depends on the area sown. Unfortunatly we do not have high quality data about area sown, in the following is described the procedure to estimate areaSown starting from areaHarvested. The following table shows the first 6 lines of the area data matrix, just to give an idea of its content.

```
area = getAllAreaData()
head(area)
```

```
##    geographicAreaM49 measuredItemCPC timePointYears
## 1:               100            0111           1961
## 2:               100            0111           1962
## 3:               100            0111           1963
## 4:               100            0111           1964
## 5:               100            0111           1965
## 6:               100            0111           1966
##    Value_measuredElement_5025 flagObservationStatus_measuredElement_5025
## 1:                    1323010
## 2:                    1253600
## 3:                    1189870
## 4:                    1195610
## 5:                    1146930
## 6:                    1142176
##    flagMethod_measuredElement_5025 Value_measuredElement_5312
## 1:                               -                    1323010
## 2:                               -                    1253600
## 3:                               -                    1189870
## 4:                               -                    1195610
## 5:                               -                    1146930
## 6:                               -                    1142176
##    flagObservationStatus_measuredElement_5312
## 1:
## 2:
## 3:
## 4:
## 5:
## 6:
##    flagMethod_measuredElement_5312
## 1:                               -
## 2:                               -
## 3:                               -
## 4:                               -
## 5:                               -
## 6:                               -
```

We report the structure of the area data matrix in order to summarise the name of columns (note 5312=area harvested; 5025= area sown)

```
## Classes 'data.table' and 'data.frame':   572832 obs. of  9 variables:
##  $ geographicAreaM49                         : chr  "100" "100" "100" "100" ...
##  $ measuredItemCPC                           : chr  "0111" "0111" "0111" "0111" ...
##  $ timePointYears                            : num  1961 1962 1963 1964 1965 ...
##  $ Value_measuredElement_5025                : num  1323010 1253600 1189870 1195610 1146930 ...
##  $ flagObservationStatus_measuredElement_5025: chr  "" "" "" "" ...
##  $ flagMethod_measuredElement_5025           : chr  "-" "-" "-" "-" ...
##  $ Value_measuredElement_5312                : num  1323010 1253600 1189870 1195610 1146930 ...
##  $ flagObservationStatus_measuredElement_5312: chr  "" "" "" "" ...
##  $ flagMethod_measuredElement_5312           : chr  "-" "-" "-" "-" ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "sorted")= chr  "geographicAreaM49" "measuredItemCPC" "timePointYears"
```

We perform the same quality checks already run for the OfficialSeedData. In particular:

```
areaPreProcessed= preProcessing(data= area,
                                normalised = FALSE)



areaConflict =areaRemoveZeroConflict(areaPreProcessed,
                                     value1= "Value_measuredElement_5025",
                                     value2= "Value_measuredElement_5312",
                                     observationFlag1= "flagObservationStatus_measuredElement_5025",
                                     methodFlag1= "flagMethod_measuredElement_5025",
                                     missingObservationFlag = "M",
                                     missingMethodFlag = "u"
)


 ensureValueRange(data = areaConflict,
                  ensureColumn = "Value_measuredElement_5312",
                  min = 0,
                  max = Inf,
                  includeEndPoint = TRUE,
                  returnData = FALSE,
                  getInvalidData = FALSE)
```

```
## All values withing specified range
```

```
  ensureValueRange(data = areaConflict,
                   ensureColumn = "Value_measuredElement_5025",
                   min = 0,
                   max = Inf,
                   includeEndPoint = TRUE,
                   returnData = FALSE,
                   getInvalidData = FALSE)
```

```
## All values withing specified range
```

```
areaConflictNormalised= normalise (areaConflict)
```

Unfortunatly, when we performe the flag validation procedure, the ensureFlagValidity function highlightes that there are still invalid flag combinations.

```
##    flagObservationStatus flagMethod
## 1:                     E          e
## 2:                     E          p
## 3:
```

```
##                       flagMethod
## flagObservationStatus           e     p
##                           4     0     0
##                     E     0 12380     1
```

```
##    flagObservationStatus flagMethod Valid Protected
## 1:                     E          e FALSE     FALSE
```

```
##    flagObservationStatus flagMethod Valid Protected
## 1:                     E          p FALSE     FALSE
```

It is evided the need to embed into the R module some authocorrection functions (very similar to the routine already developed in the production module) whose effects are:

```
## Number of (E, t) replaced with (E, -)4
```

```
## Number of (E, t) replaced with (E, -)2891
```

```
## Number of (E, e) replaced with (I, e)12380
```

```
## Number of (E, p) replaced with (E, f)1
```

Imputation of Area Sown

To impute the actual area sown, the following approach is taken:

- If values of the area sown and the area harvested are available, then an average ratio of the area sown to the area harvested is computed. Then, if the area sown is unavailable in one year, it is imputed by multiplying the area harvested in the following year by the average ratio. The idea is that the areaSown is always equal or greater than areaHarvested. The ratio represents the factor to be multiplied to the available areaHarvested in order to obtain the areaSown. The avarage is computed by Country-Commodity combinations. The discrepancy between areaSown and areaHarvested is strictly dependent on the commodity and some evironmental factors (climate, kind of soil..)
- If no prior information on the area sown is available or the ratio is erroneosly lower that 1, the corrective factors are imposed equal to 1 and consequently the area sown is assumed to be equal to the area harvested.

We show some lines of the dataset containg area data

Climate data

We pull climate data from the World Bank DB, and we fill the missing values with the average temperature (computed by country).

We are now ready to merge all tha data in a unique data table containing all the input for the model.

We are now ready to perform the imputation. The finalModelData containd all the ingredients..

```
head(finalModelData)
```

```
##    geographicAreaM49 timePointYears measuredItemCPC
## 1:               100           1989            0111
## 2:               100           1989            0112
## 3:               100           1989            0115
## 4:               100           1990            0111
## 5:               100           1990            0112
## 6:               100           1990            0115
##    Value_measuredElement_5525 flagObservationStatus_measuredElement_5525
## 1:                     341000                                          T
## 2:                      24000                                          T
## 3:                      84000                                          T
## 4:                     346000                                          T
## 5:                      23000                                          T
## 6:                      83000                                          T
##    flagMethod_measuredElement_5525 cpcLvl1 cpcLvl2 cpcLvl3
## 1:                               -       0      01     011
## 2:                               -       0      01     011
## 3:                               -       0      01     011
## 4:                               -       0      01     011
## 5:                               -       0      01     011
## 6:                               -       0      01     011
##    Value_measuredElement_5025 flagObservationStatus_measuredElement_5025
## 1:                    1138252
## 2:                     898317
## 3:                     360075
## 4:                    1162775
## 5:                     848745
## 6:                     359950
##    flagMethod_measuredElement_5025 Value_measuredElement_5312
## 1:                               -                    1138252
## 2:                               -                     563249
## 3:                               -                     360075
## 4:                               -                    1162775
## 5:                               -                     424428
## 6:                               -                     359950
##    flagObservationStatus_measuredElement_5312
## 1:
## 2:
## 3:
## 4:
## 5:
## 6:
##    flagMethod_measuredElement_5312 Value_areaSownRatio
## 1:                               -            1.006296
## 2:                               -            1.389649
## 3:                               -            1.007750
## 4:                               -            1.006296
## 5:                               -            1.389649
## 6:                               -            1.007750
##    flagObservationStatus_areaSownRatio flagMethod_areaSownRatio
## 1:                                   I                        e
## 2:                                   I                        e
## 3:                                   I                        e
## 4:                                   I                        e
## 5:                                   I                        e
## 6:                                   I                        e
##    Value_wbIndicator_SWS.FAO.PREC Value_wbIndicator_SWS.FAO.TEMP
## 1:                       40.42265                       10.96261
## 2:                       40.42265                       10.96261
## 3:                       40.42265                       10.96261
## 4:                       39.40342                       11.48194
## 5:                       39.40342                       11.48194
## 6:                       39.40342                       11.48194
```