# README

1. Crop data: 1_Create_Yield_Dataframe_perSpecies.R

Reads the file « 0_all_crops_names_clean.txt » that contains a list of all crops file names. Here we removed durum wheat, rapeseed spring because of missing data. 16 crop are left in the cleaned file:

- `"bar_spr"`
- `"bar_win"`
- `"bar_tot"`
- `"mai_tot"`
- `"oat_spr"`
- `"oat_win"`
- `"oat_tot"`
- `"pot_tot"`
- `"rap_win"`
- `"rap_tot"`
- `"sug_tot"`
- `"sun_tot »`
- `"wht_spr"`
- `"wht_win"`
- `"wht_tot"`
- `"wne_tot"`

Note that rape_spring has been removed

Source needed: the anomaly_rev_CropSince1900.R function to compute all three types of anomalies The loop is on crop names because all along there is going to be one file per crop.

Compute one yield file per crop with all three types of prediction (polynomial, loess and spline) and three types of anomalies (raw, normalized by the prediction and standardized by the variance). The files also contain raw yield, area and production data.

Each file is stored in the result folder at first and then moved to « Computed_Total_Files_w_anomalies » folder in the Crop data. The 16 files are named: « TABLE_Yield_Pred_Anomalies_All_Detrend_Methods___NOM.csv »

This code end with: 1. a plot of the time series of area share of winter and spring crops that have a tot counterpart (see beginning of climate section) + the equality between winter + spring and total 2. a short section to make a map using the shp file departments. This works with yield area or production

---

2. Climate data: 2_Create_Clim_Dataframe_perSpecies.R

2a_ SAFRAN DATA (1958-2016)

Reads the file « data_clim_fr_new.csv» that contains SAFRAN data in the folder « DataClim_Crop_Since_1900 ».

Stores all growing months for each crop: http://agreste.agriculture.gouv.fr/IMG/file/dossier8_cultures_panorama.pdf http://agreste.agriculture.gouv.fr/IMG/pdf/calend_paru_gcult.pdf

The code is separated in spring and winter sections (because winter requires to attribute past fall to next harvest years). In the spring crops section are also computed all crops tot grown in the spring (e.g., maize) note that for barley there is a distinction between the north and the south part of the Loire, we take the longest time period, i.e. here sud de la loire starts in January and north in February so we take January for the total.

For winter crops there are two things to change each time the table is assembled: 1) the fact that rapeseed goes from 8 to 12 in the preceding winter while all other go from 10 to 12 at two different spots and 2) the crop name itself at two different spots. In the winter section we also select the month for all total crops that are grown either in the spring or in the winter. These are wheat, barley, oat and rapeseed. As a first step we consider growing seasons as if these crops were all winter crops with the idea that this also covers the spring period. (see the ratio plotted as the end of code 1 for further insight).

One table per crop species is stored as: Clim_NOM_SAFRAN.csv First in the result folder but than stored in the folder « DataClim_Crop_Since_1900 » in the folder named « Compute_SAFRAN_GrowingSeason_PerCrop »

16 crop species total:

bar_spr_SAFRAN.csv bar_tot_SAFRAN.csv bar_win_SAFRAN.csv mai_tot_SAFRAN.csv oat_spr_SAFRAN.csv oat_tot_SAFRAN.csv oat_win_SAFRAN.csv pot_tot_SAFRAN.csv rap_tot_SAFRAN.csv rap_win_SAFRAN.csv sug_tot_SAFRAN.csv sun_tot_SAFRAN.csv wht_spr_SAFRAN.csv wht_tot_SAFRAN.csv wht_win_SAFRAN.csv wne_tot_SAFRAN.csv

2b. Reanalysis data (1900-2010)

This is not done yet

---

3. No CC simulations: 3_Create_noCC_Dataset_LobellMethodology.R

Reads the climate files created in code 2 and stored in Compute_SAFRAN_GrowingSeason_PerCrop (see climate path)

Here you can choose lobell= TRUE or FALSE. if TRUE this condition computes the growing season average per climate variable only for the four month prior to harvest as done by lobell for wheat (i.e., just as Lobell does. (see SI page 2 " Finally, in the case of wheat, we used the dominant production season for each country and use the weather data for the 120 days prior to harvest. » If FALSE the average is computed over the totality of the growing season (winter included when relevant).

Loop on crop species (a character vector is created at the beginning of the code)

1. It computes Tmean as the average of Tmax and Tmin (because this is the variable used by Lobel) 1bis. It computes a mean over all months (i.e., a growing season mean) or only over the last four month (if the condition is TRUE) for each of the climatic variables in the file. This is stored in object TAB.mean 1ter. The code computes a Tmean using Tmax and Tmin - to fit with what Lobell does (Lines 39-45, note that we here have to separate and merge tables to compute a mean within the data frame)

2. Computes two types of trends in each of the variables, using (i) poly and (ii) loess. A plot is here to check that it works well (data+trend-residuals)

3. It simulates time series with no CC for each variable. (i) based on the first year of the time series and then using several start years: first=1958 + 1965,70,75,78,80,82 It thus creates 7 new vectors for each climatic variable. If start year is not the first year (for example 1958 for safran data), then the vector contains the observation up to the start date and detrended data afterwards (residuals + start year). For example, if the start year is 1980 the new vector contain: data for 1958-19780), (data-trend)+trend value in 1980 for the period 1981-2016. A plot illustrated the importance of the startyear on the time series.

4. it merges the results from poly and loess in order to have only one table per crop species. Two tables are potentially saved: if lobell= TRUE a table ‘TABLE_CLIM_OBS_&*LOBELL_4month*’ is saved if lobell = FALSE, a table ‘TABLE_CLIM_OBS_&*LOBELL*’ is saved.

At the end of the code all 16 tables are saved automatically in the Result folder. Than I move them to a folder named: « Computed_noCC_SAFRAN_PerCrop »

Note that the code is of course easily adaptable to other climate data (i.e., reconstruction).

---

4. Merge Crop and Climates: 4_Merge_Yields_Climate_Obs&noCC.R

Reads only the no climate change files because there are the observations averaged over the growing season in the no climate change files (this enables us to skip one merge step). Note that there are now two types of no OBS & no climate change files, the ones that are calculated over the totality of the time series and the ones that are calculated over the 4 last month before harvest.

It starts with the creation of a character vector that computes all yield and climate names in the same order so as to proceed in a loop. When computing the loop, the name of the climate and yield files are printed to check for the good correspondance. Then it merges using department-year. This creates repetitions since there is now several variables in each department year (for only one yield value).

The files are saved in the result folder and then in a folder « Computed_Merged_Files_Climate&Yields » depending on whether the lobell condition is TRUE or FALSE The files are named for example:

TABLE_MERGED_YIELD_CLIMATE_4monthTABLE_CLIM_OBS_&LOBELL_4month_bar_spr_SAFRAN.csv OR TABLE_MERGED_YIELD_CLIMATE_TABLE_CLIM_OBS&_LOBELL_bar_spr_SAFRAN.csv

!!!!!! i=1 the first line in table SIMU is all 0. To verify.

---

5. Lobell Models linking yield and climate: 5_Yield_Climate_Analysis.R

This code makes a regression of climate on yields with observation and with all the no climate change simulations computed in code 4.

In a first step we read the total merged table and then lists the content of the columns as comments.

1. It reads one complete table per species (TABLE.tot) if LOBELL is TRUE and another is LOBELL is FALSE. Then it creates a new data.frame with one column per variable (and not the climate variables as lines) to ease the glm. This is based on the function spread in dplyr. The new table is called TABLE.spread.

Before applying the model we change the null values to very small values (i.e., 1^-10) in order to remove the log(0) problem 2. It applies Lovell's model as: model<-glm(log(as.numeric(paste(yield)))~ departement*year_harvest+departement*I(year_harvest^2)+ Tx + (Tx^2) + PR + (PR^2), data=TABLE.spread )

3. It computes the new predictions using the model fitted on the observations for each 'scenario' (a scenario is a combination of a type of detrending of the climate data - linear or loess and a starting date) and new data, new data being the table containing all the simulations. Not that in the loop that does that (L ~90) the columns are renamed to be called Tmean and PR each time - so that the prediction can be calculated from model and on the simulations. This automatic renaming is done using the function sub with subtracts a suffix in a column name. This creates a table named TABLE.ALL.SIMUS containing for each year in each department a value for the actual model prediction from observations and a simulated yield (we take the exponential in the loop because the model is computed as a log). Note that here we compute the value of change for Temperature and precipitation together and not with the alternatives (i.e., either only Tmean or only PR as done by Lobell in his paper). One verification plot enables us to plot the time series of predicted yields without climate change compared to the regular one estimated with a regression on observations.
4. Difference with and without climate change In a double loop (per scenario and per department) it computes the average mean between the observations and the simulations as : (predicted yield from observations - predicted yield from simulation)/predicted yield from observations These are stored in a MEAN table.
5. Plots We compute a time-average using the base year « since » and then

(i) plot a box plot that represents the spatial spread for each scenario (and a inter-scenario mean computed in two steps)-. Note here to compute a production loss
(ii) plot a map to see the spatial patterns of losses

The table is saved in « LOBELL_SIMULATIONS_wht_win_SAFRAN » in a results clean folder.

In a fifth section, we compute average différences.

Growing seasons

bar_spr_gs <- c(2:8) bar_win_gs <- c(10:12,1:7) mai_tot_gs <- c(4:12) oat_spr_gs <- c(1:8) oat_win_gs <- c(10:12,1:8) pot_tot_gs <- c(4:10) rap_spr_gs <- c(2:7) rap_win_gs <- c(8:12,1:7) sug_tot_gs <- c(4:11) sun_tot_gs <- c(3:9) wht_spr_gs <- c(2:8) wht_win_gs <- c(10:12,1:8) wne_tot_gs <- c(3:10)

PBM !!!

# wdu_tot => pas distinction spr-win in data

wdu_spr_gs<- c(1:9) wdu_win_gs<- c(10:12,1:8) ########################################