Land-use data harmonization

Vincent Pellissier

27 November 2018

This report explains the steps taken to correct the land-use data. The masterfile used is the version 1.19, as found on November 27th 2018. The sheet in Grassplot 1.19.xlsx containing the LU information is the sheet 'datasets' and was read and saved as a .rds file to ease the process (faster loading)

```
df <- readRDS(file.path(path_grassplot, 'Grassplot 1.19_Data.rds'))
df <- df %>%
        mutate_at(.vars = c(95:104), funs(ifelse(. %in% c('NA', '[NA]'), NA, .)))
```

The columns 98 (mowing frequency) and 100 (grazing intensity) are renamed as `mowing_frequency` and `grazing intensity`:

```
names(df)[which(names(df) %in% c('Mowing frequency: cuts per year (2=2cut/yr; 1=1cut/yr; 0.5=1 cut/2 yr
```

The column 104 (Fertilized (1/0)) should be a binary column. Yet, it contains the values c("0", "0.5", "1", "0.3"), meaning that it is an intensity column. The column is thus renamed `fertilization_intensity`

```
names(df)[which(names(df) == 'Fertilized (1/0)')] <- 'fertilization_intensity'
```

PRELIMINARY NOTE: there are duplicated ID in the column `Grassplot ID of plot`. The following ID are duplicated DE__B__N001__0.0001aa, NA

1 Correction of the land use column

The file lookup_table_LU.xlsx contains two sheets:

- land_use_detail: lookup table between the column 96 and other column. This needs to be discussed further before I finalize the table.
- global_land_use: lookuptable matching the values in column 95 with a new classification. This new classification is a set of binary columns (allowing for mixed land-use) plus one declarative column. Each binary column correspond to a management practice (grazed, mown, burnt, fertilized, abandoned, natural_grassland), the declarative column stores additional information that cannot be put in one of the 6 columns (*i.e* trampled) and that will be used afterward. **NOTE** LAS, P, L were noted as land-use for the dataset PL__C and N was noted as land-use for the dataset RU__J. I could not find what these stand for. I assumed that these plots don't have a land-use but the info is still in the column 'other'.

```
lut <- read_excel(file.path(path_grassplot, "lookup_table_LU.xlsx" ),
                  sheet = 'global_land_use')
```

The global_land_use lookup table is used to harmonize the column 95 with the new binary columns:

```
df <- df %>%
    left_join(lut,
              by = c("Land use (5 standard categories: mown, grazed, abandoned, natural grassland, NA)"
```

2 Correction of the intensity/frequency columns

These three columns (mowing_frequency, grazing_intensity, fertilization_intensity) are presently coded as text columns
The column mowing_frequency contains 1 non-numeric values : x.

The column grazing_intensity contains 4 non-numeric values : low, middle, high, overgrazing.
The column fertilization contains 0 non-numeric values .

2.1 Mowing frequency

The mowing_frequency does have non-numerical values.
A closer look (manual) at the master files shows that there is some information in the column 96, that can be converted into mowing_frequency.
The column is converted to numerical, meaning that we lose the information for now, but it will be retrieved later using the column 96.

```
df$mowing_frequency <- as.numeric(df$mowing_frequency)
```

Post correction check: original number of plots with frequency - present number of plots with frequency (should be 0) = 15. **Manual note** The difference here is due to the removal of the plots with mowing_frequency ==‘x’.

2.2 Grazing intensity

The grazing_intensity does have non-numerical values. The verbal descriptors low, middle, high, overgrazing are replaced by numerical values and the column is converted to numerical:

- `overgrazing` and `high` are replaced by 1
- `middle` is replaced by 0.5
- `low` is replaced by 0.1

```
df[df$grazing_intensity %in% c('overgrazing', 'high'),]%<>%
  mutate(grazing_intensity = 1)

df[df$grazing_intensity %in% c('middle'),]%<>%
  mutate(grazing_intensity = 0.5)

df[df$grazing_intensity %in% c('low'),]%<>%
  mutate(grazing_intensity = 0.1)

df$grazing_intensity <- as.numeric(df$grazing_intensity)
```

Post correction check: original number of plots with intensity - present number of plots with intensity (should be 0) = 0.

2.3 Fertilization intensity

The fertilization_intensity column does not have non-numerical values.
The column is converted to numerical.

```
df$fertilization_intensity <- as.numeric(df$fertilization_intensity)
```

Post correction check: original number of plots with intensity - present number of plots with intensity (should be 0) = 0.

3 Matching new binary columns and frequency/intensity columns.

In this step, we identify datasets with discrepancies between the new binary columns and the matching intensity column to:

- Manually check in the original datasets or publications
- Correct the discrepancies.

NOTE in the following, I assumed that if the intensity/frequency/duration column was non empty, but the binary was, the intensity/frequency/duration column takes precedence (*i.e* if the frequency/intensity/duration, the binary column **HAS** to be == 1)

3.1 Mowing and mowing intensity

3.1.1 *Datasets containing plots for which mowing intensity > 0 & mown != 1 (here and after, we refer to the new binary columns)*

- The datasets CZ__J, EU__K contain 225 plots having a mowing intensity but classified as mown == 0. These plots have been manually checked the 8/11/2018 (masterfile v1.8) and contains not errors or additional information. They are thus reclassified as mown == 1.
  NOTE If necessary, we can discuss a frequency threshold the classify a plot as mown or not.

```
df[df$mowing_frequency != 0 & !is.na(df$mowing_frequency) & df$mown == 0,] %<>%
  mutate(mown = 1)
```

Post reclassification check: number of plots with a discrepancy - number of plots reclassified (should be 0) = 0

3.2 Grazing and Grazing intensity

3.2.1 *Datasets containing plots for which grazing intensity > 0 & grazed != 1:*

- The datasets AZ__B, EU__K, IR__A, PL__D, TR__B, UA__G contain 1189 plots having a mowing intensity but classified as grazed == 0. These plots have been manually checked the 8/11/2018 (masterfile v1.8) and contains not errors or additional information. They are thus reclassified as grazed == 1.

```
df[df$grazing_intensity != 0 & !is.na(df$grazing_intensity) & df$grazed == 0,] %<>%
  mutate(grazed = 1)
```

Post reclassification check: number of plots with a discrepancy - number of plots reclassified (should be 0) = 0

3.3 Fertilization and fertiliation intensity

3.3.1 *Datasets containing plots for which fertilization intensity > 0 & fertilized != 1 (here and after, we refer to the new binary columns)*

- The datasets ES__P, EU__K, PL__D, TR__A, TR__B contain 325 plots having a fertilization intensity but classified as fertilized == 0. These plots have **not** been manually verified (to be done). They are thus reclassified as fertilized == 1.

```
df[df$fertilization_intensity != 0 & !is.na(df$fertilization_intensity) & df$fertilized == 0,] %<>%
  mutate(fertilized = 1)
```

Post reclassification check: number of plots with a discrepancy - number of plots reclassified (should be 0) = 0

4 Matching new and old binary columns

In this step, we identify and correct discrepancies between the newly created and corrected binary column, and the former ones. Here, we consider that mown == 1 or grazed == 1 is always correct, since it is based on the above correction (based either on the broad land-use or on the mowing / grazing intensity). Thus, we only identify (and potentialy correct) plots for which the old binary column is 0 or NA and the new one is not 0 or NA.

4.1 Mowing

Table 1: Contingency table of columns mown and Mowing (1/0)

Mowing (1/0)	0	1	<NA>
?	NA	NA	6
0	7800	213	22
1	NA	622	NA
x	NA	NA	18
NA	150331	6807	22408

4.1.1 Datasets with discrepancies between mown and Mowing (1/0) (automatic report):

The dataset(s) AT_E contain(s) 18 plots with mown = NA and Mowing (1/0) = x

The dataset(s) PL_A contain(s) 6 plots with mown = NA and Mowing (1/0) = ?

4.1.1.1 Manual verification of discrepancies (has to be updated manually if necessary for each version of the masterfile)

4.1.1.1.1 *Datasets containing plots with mown = NA & Mowing (1/0) == ? (new vs former column):*

The following plots in PL_A have mown = NA and Mowing (1/0) == '?'

```
## [1] "PL_A_N003_0.0001a" "PL_A_N003_0.001a"  "PL_A_N003_0.01a"
## [4] "PL_A_N003_0.1a"    "PL_A_N003_10a"     "PL_A_N003_1a"
```

No additional information could be found, so the plots are left with mown = NA

4.1.1.2 Check

Table 2: Contingency table of columns mown and Mowing (1/0) after reclassification (check)

Mowing (1/0)	0	1	<NA>
?	NA	NA	6
0	7800	213	22
1	NA	622	NA
x	NA	NA	18
NA	150331	6807	22408

4.2 Grazing

Table 3: Contingency table of columns grazed and Grazing (1/0)

Grazing (1/0)	0	1	<NA>
?	NA	7	6
0	3441	13	17
1	136	6793	NA
2	NA	1	NA
probably	6	NA	NA
NA	125971	29405	22431

4.2.1 Datasets with discrepancies between grazed and Grazing (1/0) (automatic report):

The dataset(s) PL_A contain(s) 6 plots with grazed = NA and Grazing (1/0) = ?

The dataset(s) PL_A contain(s) 6 plots with grazed == 0 and Grazing (1/0) = probably

The dataset(s) PL_A, UA_L contain(s) 136 plots with grazed == 0 and Grazing (1/0) = 1

4.2.1.1 Manual verification of discrepancies (has to be updated manually if necessary for each version of the masterfile)

4.2.1.1.1 *Datasets plots with grazed = NA & Grazing (1/0) == '?':*

The following plots in PL_A have grazed = NA and Grazing (1/0) == '?'

```
## [1] "PL_A_N003_0.0001a" "PL_A_N003_0.001a"  "PL_A_N003_0.01a"
## [4] "PL_A_N003_0.1a"    "PL_A_N003_10a"     "PL_A_N003_1a"
```

No additional information could be found, so the plots are left as grazed = NA

4.2.1.1.2 *Datasets plots with grazed == 0 & Grazing (1/0) == 'probably':*

The following plots in PL_A have grazed == 0 and Grazing (1/0) == ''probably'

```
## [1] "PL_A_N004_0.0001a" "PL_A_N004_0.001a"  "PL_A_N004_0.01a"
## [4] "PL_A_N004_0.1a"    "PL_A_N004_10a"     "PL_A_N004_1a"
```

No additional information could be found, so the plots are left as grazed == 0.

4.2.1.1.3 *Datasets plots with grazed == 0 & Grazing (1/0) == '1':*

The following plots in PL_A have grazed == 0 and Grazing (1/0) == 1

```
##   [1] "PL_A_N005_0.0001b" "PL_A_N005_0.001b"  "PL_A_N005_0.01b"
##   [4] "PL_A_N005_0.1b"    "PL_A_N005_10b"     "PL_A_N005_1b"
##   [7] "UA_L_N001_0.0001a" "UA_L_N001_0.0001b" "UA_L_N001_0.001a"
##  [10] "UA_L_N001_0.001b"  "UA_L_N001_0.01a"   "UA_L_N001_0.01b"
##  [13] "UA_L_N001_0.1a"    "UA_L_N001_0.1b"    "UA_L_N001_100"
##  [16] "UA_L_N001_10a"     "UA_L_N001_10b"     "UA_L_N001_1a"
##  [19] "UA_L_N001_1b"      "UA_L_N002_0.0001a" "UA_L_N002_0.0001b"
##  [22] "UA_L_N002_0.001a"  "UA_L_N002_0.001b"  "UA_L_N002_0.01a"
##  [25] "UA_L_N002_0.01b"   "UA_L_N002_0.1a"    "UA_L_N002_0.1b"
##  [28] "UA_L_N002_100"     "UA_L_N002_10a"     "UA_L_N002_10b"
##  [31] "UA_L_N002_1a"      "UA_L_N002_1b"      "UA_L_N003_0.0001a"
##  [34] "UA_L_N003_0.0001b" "UA_L_N003_0.001a"  "UA_L_N003_0.001b"
##  [37] "UA_L_N003_0.01a"   "UA_L_N003_0.01b"   "UA_L_N003_0.1a"
##  [40] "UA_L_N003_0.1b"    "UA_L_N003_100"     "UA_L_N003_10a"
##  [43] "UA_L_N003_10b"     "UA_L_N003_1a"      "UA_L_N003_1b"
##  [46] "UA_L_N004_0.0001a" "UA_L_N004_0.0001b" "UA_L_N004_0.001a"
##  [49] "UA_L_N004_0.001b"  "UA_L_N004_0.01a"   "UA_L_N004_0.01b"
##  [52] "UA_L_N004_0.1a"    "UA_L_N004_0.1b"    "UA_L_N004_100"
##  [55] "UA_L_N004_10a"     "UA_L_N004_10b"     "UA_L_N004_1a"
##  [58] "UA_L_N004_1b"      "UA_L_N005_0.0001a" "UA_L_N005_0.0001b"
##  [61] "UA_L_N005_0.001a"  "UA_L_N005_0.001b"  "UA_L_N005_0.01a"
##  [64] "UA_L_N005_0.01b"   "UA_L_N005_0.1a"    "UA_L_N005_0.1b"
##  [67] "UA_L_N005_100"     "UA_L_N005_10a"     "UA_L_N005_10b"
##  [70] "UA_L_N005_1a"      "UA_L_N005_1b"      "UA_L_N006_0.0001a"
##  [73] "UA_L_N006_0.0001b" "UA_L_N006_0.001a"  "UA_L_N006_0.001b"
##  [76] "UA_L_N006_0.01a"   "UA_L_N006_0.01b"   "UA_L_N006_0.1a"
```

```
## [79] "UA_L_N006_0.1b"    "UA_L_N006_100"     "UA_L_N006_10a"
## [82] "UA_L_N006_10b"     "UA_L_N006_1a"      "UA_L_N006_1b"
## [85] "UA_L_N007_0.0001a" "UA_L_N007_0.0001b" "UA_L_N007_0.001a"
## [88] "UA_L_N007_0.001b"  "UA_L_N007_0.01a"   "UA_L_N007_0.01b"
## [91] "UA_L_N007_0.1a"    "UA_L_N007_0.1b"    "UA_L_N007_100"
## [94] "UA_L_N007_10a"     "UA_L_N007_10b"     "UA_L_N007_1a"
## [97] "UA_L_N007_1b"      "UA_L_N008_0.0001a" "UA_L_N008_0.0001b"
## [100] "UA_L_N008_0.001a"  "UA_L_N008_0.001b"  "UA_L_N008_0.01a"
## [103] "UA_L_N008_0.01b"   "UA_L_N008_0.1a"    "UA_L_N008_0.1b"
## [106] "UA_L_N008_100"     "UA_L_N008_10a"     "UA_L_N008_10b"
## [109] "UA_L_N008_1a"      "UA_L_N008_1b"      "UA_L_N009_0.0001a"
## [112] "UA_L_N009_0.0001b" "UA_L_N009_0.001a"  "UA_L_N009_0.001b"
## [115] "UA_L_N009_0.01a"   "UA_L_N009_0.01b"   "UA_L_N009_0.1a"
## [118] "UA_L_N009_0.1b"    "UA_L_N009_100"     "UA_L_N009_10a"
## [121] "UA_L_N009_10b"     "UA_L_N009_1a"      "UA_L_N009_1b"
## [124] "UA_L_N010_0.0001a" "UA_L_N010_0.0001b" "UA_L_N010_0.001a"
## [127] "UA_L_N010_0.001b"  "UA_L_N010_0.01a"   "UA_L_N010_0.01b"
## [130] "UA_L_N010_0.1a"    "UA_L_N010_0.1b"    "UA_L_N010_100"
## [133] "UA_L_N010_10a"     "UA_L_N010_10b"     "UA_L_N010_1a"
## [136] "UA_L_N010_1b"
```

- PL_A contains 6 plots (PL_A_N005_xxxxb) noted as abandoned with ‘occasionally grazed or trampled’ in the detailed column (column 96). These plots will be classified as grazed in the new binary column.

```r
df[df$`grazed` %in% '0' & df$`Grazing (1/0)` %in% '1',]%<>%
    mutate(grazed = 1)
```

4.2.1.2 Check

Table 4: Contingency table of columns grazed and Grazing (1/0) after reclassification (check

Grazing (1/0)	0	1	<NA>
?	NA	7	6
0	3441	13	17
1	NA	6929	NA
2	NA	1	NA
probably	6	NA	NA
NA	125971	29405	22431

4.3 Burning

Table 5: Contingency table of columns burnt and Burning (1/0)

Burning (1/0)	0	<NA>
?	6	NA
0	7196	62
1	187	3
x	NA	34
NA	158384	22355

4.3.1 Datasets with discrepancies between burnt and Burning (1/0) (automatic report):

The dataset(s) AT_E contain(s) 34 plots with burnt = NA and Burning (1/0) = x

The dataset(s) BY_A, CH_E, IR_A, IT_L, RS_A, RU_J, RU_L, TJ_A, UA_G, UA_I, UA_J, UA_L contain(s) 187 plots with burnt == 0 and Burning (1/0) = 1

The dataset(s) BY_A contain(s) 3 plots with burnt = NA and Burning (1/0) = 1

The dataset(s) PL_A contain(s) 6 plots with burnt == 0 and Burning (1/0) = ?

4.3.1.1 Manual verification of discrepancies (has to be updated manually if necessary for each version of the masterfile)

4.3.1.1.1 *Datasets with burnt == 0 and Burning (1/0) == 1*

The following plots have burnt = 0 and Burning (1/0) == '1':

```
##   [1] "BY_A_P003"          "BY_A_P004"          "BY_A_P005"
##   [4] "CH_E_N003_0.0001b" "CH_E_N003_0.001b"  "CH_E_N003_0.01a"
##   [7] "CH_E_N003_0.01b"   "CH_E_N003_0.1a"    "CH_E_N003_0.1b"
##  [10] "CH_E_N003_100"     "CH_E_N003_10a"     "CH_E_N003_10b"
##  [13] "CH_E_N003_1a"      "CH_E_N003_1b"      "IR_A_N004_0.0001a"
##  [16] "IR_A_N004_0.0001b" "IR_A_N004_0.001a"  "IR_A_N004_0.001b"
##  [19] "IR_A_N004_0.01a"   "IR_A_N004_0.01b"   "IR_A_N004_0.1a"
##  [22] "IR_A_N004_0.1b"    "IR_A_N004_1000"    "IR_A_N004_100a"
##  [25] "IR_A_N004_100b"    "IR_A_N004_10a"     "IR_A_N004_10b"
##  [28] "IR_A_N004_1a"      "IR_A_N004_1b"      "IR_A_N004_25a"
##  [31] "IR_A_N004_25b"     "IT_L_N004_0.0001a" "IT_L_N004_0.0001b"
##  [34] "IT_L_N004_0.001a"  "IT_L_N004_0.001b"  "IT_L_N004_0.01a"
##  [37] "IT_L_N004_0.01b"   "IT_L_N004_0.1a"    "IT_L_N004_0.1b"
##  [40] "IT_L_N004_100"     "IT_L_N004_10a"     "IT_L_N004_10b"
##  [43] "IT_L_N004_1a"      "IT_L_N004_1b"      "RS_A_N010_100"
##  [46] "RS_A_N011_0.0001a" "RS_A_N011_0.0001b" "RS_A_N011_0.001a"
##  [49] "RS_A_N011_0.001b"  "RS_A_N011_0.01a"   "RS_A_N011_0.01b"
##  [52] "RS_A_N011_0.1a"    "RS_A_N011_0.1b"    "RS_A_N011_100"
##  [55] "RS_A_N011_10a"     "RS_A_N011_10b"     "RS_A_N011_1a"
##  [58] "RS_A_N011_1b"      "RU_J_P020"         "RU_J_P021"
##  [61] "RU_L_N001_0.0001a" "RU_L_N001_0.0001b" "TJ_A_N007_0.0001a"
##  [64] "TJ_A_N007_0.0001b" "TJ_A_N007_0.001a"  "TJ_A_N007_0.001b"
##  [67] "TJ_A_N007_0.01a"   "TJ_A_N007_0.01b"   "TJ_A_N007_0.1a"
##  [70] "TJ_A_N007_0.1b"    "TJ_A_N007_100"     "TJ_A_N007_10a"
##  [73] "TJ_A_N007_10b"     "TJ_A_N007_1a"      "TJ_A_N007_1b"
##  [76] "TJ_A_N008_0.0001a" "TJ_A_N008_0.0001b" "TJ_A_N008_0.001a"
##  [79] "TJ_A_N008_0.001b"  "TJ_A_N008_0.01a"   "TJ_A_N008_0.01b"
##  [82] "TJ_A_N008_0.1a"    "TJ_A_N008_0.1b"    "TJ_A_N008_100"
##  [85] "TJ_A_N008_10a"     "TJ_A_N008_10b"     "TJ_A_N008_1a"
##  [88] "TJ_A_N008_1b"      "TJ_A_N014_0.0001a" "TJ_A_N014_0.0001b"
##  [91] "TJ_A_N014_0.001a"  "TJ_A_N014_0.001b"  "TJ_A_N014_0.01a"
##  [94] "TJ_A_N014_0.01b"   "TJ_A_N014_0.1a"    "TJ_A_N014_0.1b"
##  [97] "TJ_A_N014_100"     "TJ_A_N014_10a"     "TJ_A_N014_10b"
## [100] "TJ_A_N014_1a"      "TJ_A_N014_1b"      "UA_G_N008_100"
## [103] "UA_G_N009_1"       "UA_G_N009_10"      "UA_G_N009_100"
## [106] "UA_G_N010_1"       "UA_G_N010_10"      "UA_G_N010_100"
## [109] "UA_G_N011_1"       "UA_G_N011_10"      "UA_G_N011_100"
## [112] "UA_G_N012_1"       "UA_G_N012_10"      "UA_G_N012_100"
## [115] "UA_I_N005_0.001a"  "UA_I_N005_0.001b"  "UA_I_N005_0.01a"
## [118] "UA_I_N005_0.01b"   "UA_I_N005_0.1a"    "UA_I_N005_0.1b"
## [121] "UA_I_N005_100"     "UA_I_N005_10a"     "UA_I_N005_10b"
## [124] "UA_I_N005_1a"      "UA_I_N005_1b"      "UA_I_N006_0.0001a"
```

```
## [127] "UA_I_N006_0.0001b" "UA_J_N008_0.0001a" "UA_J_N008_0.0001b"
## [130] "UA_J_N008_0.001a"  "UA_J_N008_0.001b"  "UA_J_N008_0.01a"
## [133] "UA_J_N008_0.01b"   "UA_J_N008_0.1a"    "UA_J_N008_0.1b"
## [136] "UA_J_N008_100"     "UA_J_N008_10a"     "UA_J_N008_10b"
## [139] "UA_J_N008_1a"      "UA_J_N008_1b"      "UA_J_N009_0.0001a"
## [142] "UA_J_N009_0.0001b" "UA_J_N009_0.001a"  "UA_J_N009_0.001b"
## [145] "UA_J_N009_0.01a"   "UA_J_N009_0.01b"   "UA_J_N009_0.1a"
## [148] "UA_J_N009_0.1b"    "UA_J_N009_100"     "UA_J_N009_10a"
## [151] "UA_J_N009_10b"     "UA_J_N009_1a"      "UA_J_N009_1b"
## [154] "UA_J_P010"         "UA_J_P017"         "UA_J_P052"
## [157] "UA_J_P053"         "UA_J_P074"         "UA_J_P076"
## [160] "UA_J_P081"         "UA_J_P095"         "UA_L_N011_0.0001a"
## [163] "UA_L_N011_0.0001b" "UA_L_N011_0.001a"  "UA_L_N011_0.001b"
## [166] "UA_L_N011_0.01a"   "UA_L_N011_0.01b"   "UA_L_N011_0.1a"
## [169] "UA_L_N011_0.1b"    "UA_L_N011_100"     "UA_L_N011_10a"
## [172] "UA_L_N011_10b"     "UA_L_N011_1a"      "UA_L_N011_1b"
## [175] "UA_L_N012_0.0001a" "UA_L_N012_0.0001b" "UA_L_N012_0.001a"
## [178] "UA_L_N012_0.001b"  "UA_L_N012_0.01a"   "UA_L_N012_0.01b"
## [181] "UA_L_N012_0.1a"    "UA_L_N012_0.1b"    "UA_L_N012_100"
## [184] "UA_L_N012_10a"     "UA_L_N012_10b"     "UA_L_N012_1a"
## [187] "UA_L_N012_1b"
```

No other information are found. These plots are reclassified with burnt == 1

```r
df[df$burnt %in% 0 & df$`Burning (1/0)` %in% '1',]%<>%
    mutate(burnt = 1)
```

4.3.1.1.2 *Datasets with burnt = NA and Burning (1/0) == 1*

The following plots have burnt = NA and Burning (1/0) == '?':

```
## [1] "BY_A_P006" "BY_A_P007" "BY_A_P009"
```

No other information are found. These plots are reclassified with burnt == 1

```r
df[is.na(df$burnt) & df$`Burning (1/0)` %in% '1',]%<>%
    mutate(burnt = 1)
```

4.3.1.1.3 *Datasets with burnt = NA and Burning (1/0) == ?*

The following plots have burnt == 0 and Burning (1/0) == '?':

```
## [1] "PL_A_N005_0.0001a" "PL_A_N005_0.001a"  "PL_A_N005_0.01a"
## [4] "PL_A_N005_0.1a"    "PL_A_N005_10a"     "PL_A_N005_1a"
```

No additional information could be found, so the plots are left as burnt == 0.

4.3.1.2 Check

Table 6: Contingency table of columns burnt and Burning (1/0) after reclassification (check)

Burning (1/0)	0	1	<NA>
?	6	NA	NA
0	7196	NA	62
1	NA	190	NA
x	NA	NA	34
NA	158384	NA	22355

4.4 Fertilization

Non applicable so far, since the old binary column was an intensity one.

5 Proposal for a new land-use template

The column 96 provide a lot of information that is so far not fully exploited. To make use of it, it requires to manually check each of the value in this column, to fill other columns. After the Skype meeting (26th), I propose the following columns types:

5.1 Land-use

Already explained above, these are binary columns providing information on the land-use There is one extra column that contains verbal information not found in these five columns.

- mown
- grazed
- burnt
- fertilized
- abandoned
- natural
- other

5.2 Land-use intensity

5.2.1 Mowing

- mowing_frequency. Numeric, from 0 to infinite. Number of cut(s) per year

5.2.2 Grazing

- grazing_intensity. Numeric, from 0 to 1. Self reported relative intensity of grazing.
- grazing_load. Numeric, from 0 to infinite. Grazing animal load/ha./year (maybe expressed as equivalent cattle)

5.2.3 Burning

- burning_frequency. Numeric, from 0 to infinite. Number of burning(s) per year

5.2.4 Fertilization

- fertilization_intensity. Numeric, from 0 to 1. Self reported relative intensity of fertilization
- fertilization_type. Text, ‘synthetic’ or ‘natural’.
- fertilization_details. Text, free values. Details not present in the tzo previous column (**e.g.** frequency, amount, type of chemical fertilizer. . .)

5.3 Abandonement

- years_abandonment. Numeric, from 0 to infinity. Time since abandonment in years
- former_land_use: Text, 4 values that can be combined (arable, mown, grazed, burnt). Land-use before abandonment.

5.4 Grazing details

- grazing_animal: Text, free values that can be combined, spearated by ‘/’. New values should be added in agreement with the DB manager.Type of animal grazing the land.

5.5 Land destination

- land_destination: Text, 3 possible values (cropland, farmland, recreational). The primary intent behind the land-use.

5.6 Grassland type

- primary_grassland: Numeric, 0 or 1. Is the grassland primary (1) or secondary (0)?

Table 7: Summary of the proposed template (**to be discussed**)

Column type	Column name	Variable type	Possible values
Land use	• mown • grazed • burnt • fertilized • abandoned • natural • other	• binary • binary • binary • binary • binary • binary • text	• 0/1 • 0/1 • 0/1 • 0/1 • 0/1 • 0/1 • free
Land-use intensity	• grazing_intensity • grazing_load • mowing_frequency • burning_frequency • fertilization_intensity • fertilization_type • fertilization_details	• numeric • numeric • numeric • numeric • text • numeric • text	• 0 to 1 • 0 to infinity • 0 to infinity • 0 to infinity • 0 to 1 • synthetic/natural • free
Abandonment	• years_abandonment • former_land_use	• numeric • text	• 0 to infinity • arable, mown, grazed burnt
Grazing details	• grazing animal	• text	• free
Land destination	• land_destination	• text	• cropland, farmland, recreational
Grassland type	• primary_grassland	• numeric	• 0/1

5.7 Hamonizing the land-use details column

The following adds the columns not existing yet in the DB, and export the template, with one row per unique value of the land-use detail column.
This exported file is to be checked manually, in order to fill in the template. To help the sorting, the plots and/or dataset name is left.
In a second step, it will be reintegrated to the DB, to finalize the harmonization

```
names(df)[which(names(df) == 'Land use detail (e.g. number of cuts or years since abandonment)')] <- "la

template_names <- c('mown','grazed','burnt','fertilized','abandoned',
                    'natural','other','grazing_intensity','grazing_load',
                    'mowing_frequency','burning_frequency',
                    'fertilization_intensity','fertilization_type',
                    'fertilization_details','years_abandonment',
                    'former_land_use','grazing animal','land_destination',
                    'primary_grassland')

df_details <- df[match(unique(df$land_use_details), df$land_use_details),]

for(colname in template_names){
  if(!colname %in% names(df_details))
    df_details[,colname] <- NA
}

df_details <- df_details[,c('GrassPlot ID of plot', 'land_use_details', template_names)]


write_xlsx(df_details, path = file.path(path_grassplot, 'df_details.xlsx'))
```