Cleaning and exploration of Lao PDR NFI cycle 4 data

ຂໍ້ມູນການສຳຫລວດປ່າໄມ້ແຫ່ງຊາດຄັ້ງທີ່ 4 ທີ່ຈັດລຽງແລ້ວ

2026-09-03

Table of contents

# 1. Objective

The objective of this document is to prepare calculations of biomass for the entity ‘tree’.

## 1.1 List of calculations:

* Aboveground biomass
* Tree DBH class

Before working with the tree data we will check if the data is clean.

## 1.2 Step by Step instructions

1. Setup
   1. Define user inputs
   2. Prepare environment
2. Explore data
   1. Basic information
   2. Error check
   3. Join treeplot information with tree
3. Add AGB and DBH class to each tree

# 2. Setup

Setup has 2 parts: (1) define the user inputs to specify what data to use and (2) prepare the working environment with packages, functions and data.

## 2.1 User inputs

We need to define user inputs for the scripts:

* Do we need to prepare the data? **YES**
* Do we need to download the initial Raw data? **NO**
* If not, what is the file name of the raw data (can be zip or master CSV)? **4th\_NFI\_up to2025\_02\_24\_withpath.csv**
* Do we erase all data (only do this if you download new data)? **NO**
* What is your timezone? **Asia/Bangkok**

usr <- list()  
usr$get\_new <- TRUE  
usr$get\_auto <- FALSE  
usr$get\_filename <- "4th\_NFI\_up to2025\_02\_24\_withpath.csv"   
usr$clean\_all <- FALSE  
usr$time\_zone <- "Asia/Vientiane"

## 2.2 Environment preparation

List of actions:

1. Load packages and functions
2. Set paths to data and results
3. Get sampling design variables
4. Load ancillary data
5. Get harmonized data

In this project, each of these steps is handled by a different script stored in “*R/setup*”.

These scripts have if (condition) { action } else { action } statements to adapt to the source of data. There are **three sources of data** considered:

1. **master CSV from ONA** (automatic download),
2. **master CSV from a local file** or
3. **ZIP folder from a local file**.

More details on how the data are prepared can be found in “*R/setup/get-data.R*”

At this stage, the R environment contains:

* a list of user inputs.
* a list of paths to read data and write results in the correct sub-folders.
* a list of sampling design information.
* a list of data frame with the harmonized data, meaning they are structured by entity and column names are made more explicit (ex. tree\_data\_basic\_nest2\_\_t\_az\_nest2 is renamed to tree\_azimuth.

# 3. Data exploration

Goals:

* Get some information on the data:
* Check if errors in the data
* Join land cover type and class, and canopy height from treeplot to tree

Note: the harmonized data are stored in a list of entities

exists("tree")

[1] FALSE

exists("data\_harmo")

[1] TRUE

class(data\_harmo)

[1] "list"

names(data\_harmo)

[1] "tree" "stump" "dw" "treeplot" "plot" "ntfp"

class(data\_harmo$tree)

[1] "tbl\_df" "tbl" "data.frame"

class(data\_harmo$stump)

[1] "tbl\_df" "tbl" "data.frame"

## 3.1 Basic information on data

### 3.1.1 Tree data

tree = ຕົ້ນໄມ້

data\_harmo: harmonized data = ສັງລວມຂໍ້ມູນໃຫ້ເປັນ ລະບົບ ແລະ ໄຈແຍກປະເພດຂໍ້ມູນເຊັນວ່າ: ໄມ້ເປັນ, ໄມ້ຕາຍຢືນ, ໄມ້ນອນຂອນຕາຍ ແລະ ອື່ນໆ

Number of trees in the data / **ຈຳນວນຕົ້ນໄມ້ທັງຫມົດໃນຖານຂໍ້ມູນ**

nrow(data\_harmo$tree)

[1] 9833

Number of treeplots / ຈຳນວນດອນຍ່ອຍທັງຫມົດໃນຖານຂໍ້ມູນ

nrow(data\_harmo$treeplot)

[1] 1276

Number of treeplots with trees / ຈຳນວນດອນຍ່ອຍທີ່ມີຕົ້ນໄມ້ນອນຢູ່ໃນດອນຕົວຢ່າງ

names(data\_harmo$tree)

[1] "ONA\_parent\_index" "ONA\_index" "ONA\_no"   
 [4] "tree\_harmo\_src" "tree\_no" "tree\_stem\_go"   
 [7] "tree\_stem\_no" "tree\_distance" "tree\_azimuth"   
[10] "tree\_species\_code" "tree\_species\_newlocal" "tree\_dbh"   
[13] "tree\_pom" "tree\_dbh100\_url"

length(unique(data\_harmo$tree$ONA\_parent\_index))

[1] 1044

There are 1276 treeplots in the data but only 1044 treeplots with trees.

Manual numbers:

ຈຳນວນດອນຍ່ອນທັງຫມົດແມ່ນ 1260 (typo) ດອນຢູ່ຖານຂໍ້ມູນ, ແຕ່ມີພຽງດອນຍ່ອຍຈຳນວນ 1044 ດອນເທົ່ານັ້ນທີມີຕົ້ນໄມ້ຢູ່ໃນດອນ.

Automatic numbers:

ຈຳນວນດອນຍ່ອນທັງຫມົດແມ່ນ 1276 ດອນຢູ່ຖານຂໍ້ມູນ, ແຕ່ມີພຽງດອນຍ່ອຍຈຳນວນ 1044 ດອນເທົ່ານັ້ນທີມີຕົ້ນໄມ້ຢູ່ໃນດອນ.

Tree DBH distribution:

summary(data\_harmo$tree$tree\_dbh)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 10.0 13.4 20.0 25.9 34.5 139.4

Main species:

Requires several steps:

* group\_by tree species and summarise to get the count. HINT: in summarise() use n() to count the number of elements
* arrange the data in decreasing order and keep only the first 20 rows

main\_species <- data\_harmo$tree |>  
 group\_by(tree\_species\_code) |>  
 summarise(count = n()) |>  
 filter(tree\_species\_code != "0", tree\_species\_code != "9999") |>  
 slice\_max(n = 20, order\_by = count)  
  
print(main\_species)

# A tibble: 20 × 2  
 tree\_species\_code count  
 <chr> <int>  
 1 LT0181 737  
 2 LT0183 321  
 3 LT0334 146  
 4 LT0492 120  
 5 LT0076 90  
 6 LT0723 78  
 7 LT0343 76  
 8 LT0260 69  
 9 LT0464 68  
10 LT0080 59  
11 LT0739 59  
12 LT0184 55  
13 LT0104 53  
14 LT0369 51  
15 LT0117 49  
16 LT0188 47  
17 LT0628 46  
18 LT0140 45  
19 LT0475 44  
20 LT0725 43

We have species codes, let’s get the species names.

main\_species1 <- main\_species |>  
 left\_join(anci$species\_list, by = "tree\_species\_code")  
  
print(main\_species1)

# A tibble: 20 × 4  
 tree\_species\_code count tree\_species\_scientific\_name tree\_species\_local\_n…¹  
 <chr> <int> <chr> <chr>   
 1 LT0181 737 Castanea henryi (Skan) Rehder… ກໍ່ແດງ   
 2 LT0183 321 Castanopsis acuminatissima (B… ກໍ່ເດືອຍ   
 3 LT0334 146 Diospyros sylvatica Roxb. ໜັງດຳ   
 4 LT0492 120 Lagerstroemia paniculata S.Vi… ເປືອຍ   
 5 LT0076 90 Aporosa cardiosperma (Gaertn.… ເໝືອດແອະ   
 6 LT0723 78 Schima khasiana Dyer ຄາຍໂຊ້   
 7 LT0343 76 Dipterocarpus tuberculatus Ro… ກຸງ   
 8 LT0260 69 Cratoxylum maingayi Dyer ຕີ້ວໜາມ   
 9 LT0464 68 Hopea ferrea Laness. ແຄນຫີນ   
10 LT0080 59 Aporosa villosa (Lindl.) Bail… ເໝືອດຂົນ   
11 LT0739 59 Shorea obtusa Wall. ex Blume ຈິກໂຄກ   
12 LT0184 55 Castanopsis argentea (Blume) … ກໍ່ຂ້າວ   
13 LT0104 53 Arytera litoralis Blume ກະດູກ   
14 LT0369 51 Engelhardtia serrata Blume ພ່າວຂຽວ   
15 LT0117 49 Bauhinia variegata L. ບານ   
16 LT0188 47 Castanopsis calathiformis (Sk… ກໍ່ໜາມ   
17 LT0628 46 Parashorea dussaudii Tardieu ຊີ   
18 LT0140 45 Callerya atropurpurea (Wall.)… ຂີ້ໝູປ່າ   
19 LT0475 44 Irvingia malayana Oliv.ex Ben… ກະບົກ   
20 LT0725 43 Schima wallichii (DC.) Korth. ໝີ່ຄາຍ   
# ℹ abbreviated name: ¹​tree\_species\_local\_name

write\_csv(main\_species1, here(file.path(path$res$tab, "tree-calc-main-species.csv")))

We can correct the table of main species by adding the weight of the trees due to sampling design (scale factor). The scale factor is the weight of each tree if they were measured in a 1 ha plot:

with the plot radius in meters.

R chunk description:

1. Add new columns with scale factor and tree count (1) with mutate().
2. Group the data by species code.
3. Calculate the sum of tree count and the sum of scale factors to get the initial number of trees per species count and the corrected number of trees per species if all trees were measured in 1 ha plots count\_sf.
4. Filter out the missing species codes.
5. Keep the 20 most represented species (order\_by can be switched between count and count\_sf but only count\_sf is the correct dimension).
6. Add the species names from the ancillary table in anci$species\_list.

main\_species\_corr <- data\_harmo$tree |>  
 mutate(  
 tree\_plot\_radius = if\_else(tree\_dbh < 30, 8, 16),  
 tree\_scale\_factor = 10000 / (pi \* tree\_plot\_radius^2),  
 tree\_count = 1  
 ) |>  
 group\_by(tree\_species\_code) |>  
 summarise(  
 count = sum(tree\_count),  
 count\_sf = sum(tree\_scale\_factor)  
 ) |>  
 filter(tree\_species\_code != "0", tree\_species\_code != "9999") |>  
 slice\_max(n = 20, order\_by = count\_sf) |>  
 left\_join(anci$species\_list, by = "tree\_species\_code")  
  
## Check  
# table(main\_species\_corr$tree\_plot\_radius, main\_species\_corr$tree\_scale\_factor, useNA = "ifany")

### 3.1.2 Treeplot data

How many treeplots:

nrow(data\_harmo$treeplot)

[1] 1276

How many land covers:

Initial result can be wrong if there are missing land covers. Need to filter out the NAs first.

length(unique(data\_harmo$treeplot$treeplot\_lc\_class))

[1] 22

tt <- data\_harmo$treeplot |> filter(!is.na(treeplot\_lc\_class))  
length(unique(tt$treeplot\_lc\_class))

[1] 21

How many treeplot per land cover (HINT: use table() or group\_by() and summarise()):

table(data\_harmo$treeplot$treeplot\_lc\_class, useNA = "ifany")

11 12 13 15 16AC 16EC 16OTH 16RB 16TK 21 22 31 32   
 22 610 43 14 1 12 5 31 3 133 264 1 1   
 42 51 61 62 63 71 80 81 <NA>   
 1 52 4 3 5 2 9 4 56

sort(table(data\_harmo$treeplot$treeplot\_lc\_class, useNA = "ifany"), decreasing = T)

12 22 21 <NA> 51 13 16RB 11 15 16EC 80 16OTH 63   
 610 264 133 56 52 43 31 22 14 12 9 5 5   
 61 81 16TK 62 71 16AC 31 32 42   
 4 4 3 3 2 1 1 1 1

data\_harmo$treeplot |>   
 filter(!is.na(treeplot\_lc\_class)) |>  
 group\_by(treeplot\_lc\_class) |>  
 summarise(count = n()) |>  
 arrange(desc(count))

# A tibble: 21 × 2  
 treeplot\_lc\_class count  
 <chr> <int>  
 1 12 610  
 2 22 264  
 3 21 133  
 4 51 52  
 5 13 43  
 6 16RB 31  
 7 11 22  
 8 15 14  
 9 16EC 12  
10 80 9  
# ℹ 11 more rows

## 3.2 Error Check

### 3.2.1 Unrealistic DBH

We can check that there is no outlier (very small or very big tree) and that the DBH distribution per quartile is realistic with summary().

summary(data\_harmo$tree$tree\_dbh)

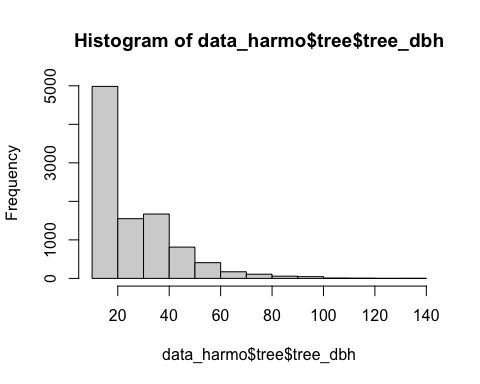
Min. 1st Qu. Median Mean 3rd Qu. Max.   
 10.0 13.4 20.0 25.9 34.5 139.4

Checklist:

* No NA: **OK**
* No trees with DBH < 10: **OK**
* No trees with unrealistic DBH (400+, they exist but are very rare): **OK**
* Median and quartiles are towards the small trees: **OK**

We can also make a histogram to see the distribution per diameter class

hist(data\_harmo$tree$tree\_dbh)



WARNING: in this histogram there are more trees in 30-40 cm DBH than 20-30 cm. This is irregular for large inventory, but in this case it comes from the different scale factors not being considered.

Let’s make a proper DBH distribution histogram. The steps are:

1. add scale factors to trees
2. add the DBH class
3. group by dbh class and get the sum of scale factors
4. make a ggplot

For steps 1 and 2:

gg\_dbh <- data\_harmo$tree |>  
 mutate(  
 tree\_treeplot\_radius = if\_else(tree\_dbh < 30, 8, 16),  
 tree\_scale\_factor = 10000 / (pi \* tree\_treeplot\_radius^2),  
 tree\_dbh\_class = case\_when(  
 tree\_dbh < 10 ~ "000-09",  
 tree\_dbh < 20 ~ "010-19",  
 tree\_dbh < 30 ~ "020-29",  
 tree\_dbh < 40 ~ "030-39",  
 tree\_dbh < 50 ~ "040-49",  
 tree\_dbh < 60 ~ "050-59",  
 tree\_dbh < 70 ~ "060-69",  
 tree\_dbh < 80 ~ "070-79",  
 tree\_dbh < 90 ~ "080-89",  
 tree\_dbh < 100 ~ "090-99",  
 TRUE ~ "100+"  
 ),  
 tree\_dbh\_class2 = floor(tree\_dbh / 10) \* 10,  
 tree\_dbh\_class2 = if\_else(tree\_dbh\_class2 < 100, tree\_dbh\_class2, 100)  
 )  
  
table(gg\_dbh$tree\_dbh\_class, useNA = "ifany")

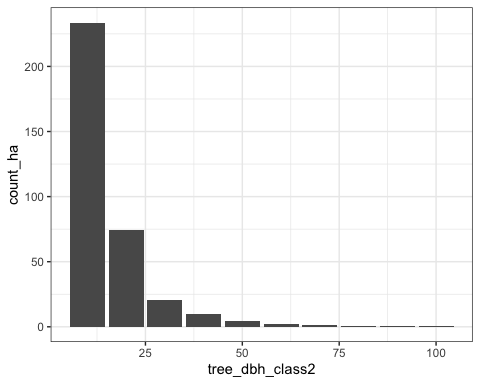
010-19 020-29 030-39 040-49 050-59 060-69 070-79 080-89 090-99 100+   
 4906 1553 1707 836 400 189 111 54 52 25

table(gg\_dbh$tree\_dbh\_class2, useNA = "ifany")

10 20 30 40 50 60 70 80 90 100   
4906 1553 1707 836 400 189 111 54 52 25

For steps 3 and 4:

n\_treeplot <- length(unique(data\_harmo$tree$ONA\_parent\_index))  
  
gg\_dbh2 <- gg\_dbh |>  
 group\_by(tree\_dbh\_class2) |>   
 summarise(  
 count\_ha = sum(tree\_scale\_factor) / n\_treeplot,  
 count = n()  
 )  
  
gg\_dbh3 <- ggplot(gg\_dbh2) +  
 geom\_col(aes(x = tree\_dbh\_class2, y = count\_ha))  
  
print(gg\_dbh3)



### 3.2.2 Distance or azimuth error

We can check issues in azimuth in the summary(): Min and max should be 0 and 360, and as we cummulate many trees, the median and quartiles should be aligned with the cardinal points:

* 1st quartile ~ 90
* 2nd quartile ~ 180 (median)
* 3rd quartile ~ 270

summary(data\_harmo$tree$tree\_azimuth)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.0 89.0 178.0 178.4 267.0 360.0

Checklist:

* No tree less than 0 or more than 360: **OK**
* 1st quartile ~ 90: **OK**
* median ~180: **OK**
* 3rd quartile ~ 270: **OK**

### 3.2.3 Small tree measured in the large circle

We want to count how many trees have DBH < 30 but distance > 8. HINT: use filter() and nrow().

# class(data\_harmo)  
# class(data\_harmo$tree)  
# class(data\_harmo$tree$tree\_dbh)  
# with(data\_harmo$tree, tree\_dbh)  
  
check\_dist <- data\_harmo$tree |>  
 filter(tree\_dbh < 30) |>  
 filter(tree\_distance > 8)  
  
nrow(check\_dist)

[1] 0

Checklist:

* number of trees with DBH < 30 and distance > 8 = 0: **OK**

PRACTICE filter() and nrow():

* number of trees with DBH < 10
* dbh10 <- data\_harmo$tree |>  
   filter(tree\_dbh < 10)  
    
  nrow(dbh10)
* [1] 0
* number of trees with DBH class is 30-39 (tree\_dbh\_class2 = 30)
* class30 <- gg\_dbh |>  
   filter(tree\_dbh\_class2 == 30)  
    
  class30\_new <- gg\_dbh |>  
   filter(tree\_dbh\_class == "030-39")  
    
  nrow(class30)
* [1] 1707
* nrow(class30\_new)
* [1] 1707
* number of trees with stem number > 1
* stem1 <- data\_harmo$tree |>  
   filter(tree\_stem\_no > 1)  
    
  nrow(stem1)
* [1] 474

### 3.2.4 Errors in stem number

Tree with multiple stems count 1 for:

* *tree distribution per DBH class (to be checked)*
* importance of species,

but count separately for basal area and AGB.

To check:

1. all trees have stem number at least 1
2. all trees with stem number > 1 have a stem number 1.

EX: table with the number of trees per stem number (show the NAs). HINT: use table().

# class(data\_harmo)  
# class(data\_harmo$tree)  
# class(data\_harmo$tree$tree\_stem\_no)  
  
table(data\_harmo$tree$tree\_stem\_no, useNA = "ifany")

1 2 3 4 5 6 7 10 <NA>   
 364 378 66 20 4 4 1 1 8995

table(data\_harmo$tree$tree\_stem\_go, useNA = "ifany")

multi one <NA>   
 888 8140 805

checklist:

* more stem 2 than stem 1: **NOT OK**
* NAs in stem\_no: **NOT OK**

We can correct when there is stem 2 but not stem 1

max(data\_harmo$tree$ONA\_index)

[1] 8212

summary(data\_harmo$tree$ONA\_index)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 1 1351 2827 3435 5506 8212

ID\_stem2 <- data\_harmo$tree |>  
 filter(tree\_stem\_no == 2) |>  
 select(ONA\_parent\_index, ONA\_index, tree\_stem\_no) |>  
 mutate(  
 ONA\_index\_unique = ONA\_parent\_index \* 1000 + ONA\_index,  
 ONA\_index\_unique\_stem1 = ONA\_index\_unique - 1  
 )  
  
ID\_stem12 <- data\_harmo$tree |>  
 mutate(ONA\_index\_unique = ONA\_parent\_index \* 1000 + ONA\_index) |>  
 filter(ONA\_index\_unique %in% c(ID\_stem2$ONA\_index\_unique, ID\_stem2$ONA\_index\_unique\_stem1))  
  
table(ID\_stem12$tree\_stem\_no, useNA = "ifany")

1 2 <NA>   
 332 378 38

ADDED OUTSIDE TRAINING

tt <- data\_harmo$tree

## 3.3 Join treeplot info

tree data doesn’t have land cover class, this information is in treeplot table. We need to get land cover class at tree level in order to apply the biomass allometric equations.

Step1: Make a treeplot table with only treeplot ID and land cover class

treeplot\_lc <- data\_harmo$treeplot |>   
 select(ONA\_treeplot\_id, treeplot\_lc\_class)

Step 2: Join this smaller treeplot with land cover table to the tree table

tree\_lc <- data\_harmo$tree |>  
 left\_join(treeplot\_lc, by = c("ONA\_parent\_index" = "ONA\_treeplot\_id"))  
  
table(tree\_lc$treeplot\_lc\_class, useNA = "ifany")

11 12 13 15 16AC 16EC 16OTH 16RB 16TK 21 22 32 51   
 309 6616 307 228 14 43 29 181 38 539 874 11 11   
 61 80 81 <NA>   
 4 18 5 606

Why NAs?

table(data\_harmo$treeplot$treeplot\_lc\_class, useNA = "ifany")

11 12 13 15 16AC 16EC 16OTH 16RB 16TK 21 22 31 32   
 22 610 43 14 1 12 5 31 3 133 264 1 1   
 42 51 61 62 63 71 80 81 <NA>   
 1 52 4 3 5 2 9 4 56

table(data\_harmo$treeplot$treeplot\_access, useNA = "ifany")

accessible danger distance other slope   
 1220 2 5 21 28

access\_lc <- data\_harmo$treeplot |>  
 filter(treeplot\_access == "accessible") |>  
 filter(is.na(treeplot\_lc\_class))  
  
nrow(access\_lc)

[1] 0

We have NAs in treeplot land cover but it is only in inaccessible plots. In tree table, we need to find why some treeplot are missing

vec\_lcna <- tree\_lc |>  
 filter(is.na(treeplot\_lc\_class)) |>  
 pull(ONA\_parent\_index) |>  
 unique() |>  
 sort()  
  
vec\_lcna

[1] 129 584 587 588 589 590 591 592 593 594 595 596 598 603 604  
[16] 605 606 607 608 609 610 774 775 776 777 778 779 780 781 783  
[31] 785 786 789 790 794 796 1007 1008 1009 1011 1012 1013 1014 1015 1016  
[46] 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1028 1029 1030 1031 1032  
[61] 1034 1151 1152 1153 1154 1155 1156 1157 1158

There too many missing treeplots so that the corresponding trees have missing land cover. Potentially, we are seeing a mix of entry treeplot code error and QAQC treeplots, that the harmonization script is automatically removing.

We can come back to the initial tables to get the IDs of QAQC treeplots and remove it from vec\_lcna.

vec\_qc <- data\_init$treeplot\_init |>   
 filter(plot\_info\_\_crew\_lead == "QC") |>  
 pull(ONA\_index) |>  
 sort()

Now we want to remove the QAQC plots from the tree table

tree\_clean <- tree\_lc |>  
 filter(!ONA\_parent\_index %in% vec\_qc)  
  
vec\_lcna <- tree\_clean |>  
 filter(is.na(treeplot\_lc\_class)) |>  
 pull(ONA\_parent\_index) |>  
 unique() |>  
 sort()  
  
vec\_lcna

integer(0)

# 4. Calculations

## 4.1 Tree level calculations

Use mutate() and case\_when() to assign AGB equation parameters a and b based on land cover type

# 5. Aggregation