Overall information

# Get ready

1. Install R and Rtools and R studio (very important to have all)
2. Install packages: tidyverse, ggplot2, zoo
3. Find your own folder and also look around the shared google drive. Make sure you have access.
4. Copy this document into your own folder, so that you can put notes in. Please also highlight the steps you have completed, like this, so that we know your progress.
5. If you have questions, you can either direct message a trainer, or leave the question in “Q&A question list” (a document on this google drive)

# Meta-data and files organisation

1. Upload your fine litter fall file into your folder and renamed it in the format “flf\_20190201\_XXX.csv”; **flf** - fine litter fall; **20190201** - the time you uploaded this file, as the version of the file, XXX - your plot code.
2. Find the example file of fine litter fall: flf\_20190201\_example.csv, open it and have a look; Think about how to convert your data into this format
3. Think about **uniqueness**. In file “flf\_20190201\_example.csv”; you can see there is a column called “unique\_id”, this is basically a combination of plot\_code + litterfall\_trap\_num + date. The rule is that, every row should have its own distinct unique\_id, in other word, for a given date, plot and litterfall trap, there must be only one value. With uniqueness defined, one row of information is also called “one entry”.
4. Think about **missing value**, as you can see, missing value here is termed “NA”, it means simply that you don’t have information. Please note that NA is different from 0, which means you know the amount and the amount is nil. Don’t leave it blank in Excel because it might confuse R.
5. The basic way to convert your data into a format like this is to copy and paste in Excel. However, clicking the mouse frequently including manually data editing and copy/paste tends to make mistakes. It is possible to use R codes to convert your data into the standard format. Using a package called “tidyverse”. Please get in touch with us to discuss further.

# Other tips

1. Ensure forward slashes when setting working directory
2. Clear session, workspace and plot history when rerunning same script for different plots to avoid weird errors.

# Try R

**Some quick and useful information on R (but I am sorry that they are all in English…)**

**I am writing this because now it is a great time that you step forward and learn more about R. In the past, we were presenting you a black-box of R codes, which accept your dataset and produce NPP. We are now trying to open the black-box and show you all the gadgets within. Let’s try to understand every process and even modify it according to your plot:**

1. Now you can find two R files in google drive folder “fine\_litter\_fall”, “finelitterfall\_teaching.rmd” is the main R file you need to run. The “functions.r” is the supplementary file that you will need when running the main R file.
2. This file will convert trap level fine litter fall into plot level flf. Try to follow the instruction in this file.
3. We recommend you to read the book “Wickham, Grolemund - 2017 - R for Data Science Import, Tidy, Transform, Visualize, and Model Data”, to learn more about Rstudio, you can follow some youtube video like (<https://www.youtube.com/watch?v=BoDTNbP7_OQ>)
4. What to do when I was stumbled by an R BUG[[1]](#footnote-0)? How to locate the reason for errors? Video: <https://www.rstudio.com/resources/rstudioconf-2018/debugging-techniques-in-rstudio/>
5. How do R understand “data”? How do R input, store and output them? How do I use R to manipulate data? This is the recommended book: <https://r4ds.had.co.nz/introduction.html>
6. Unfortunately, we have to rerun our script (R codes) lots of time - probably because you changed your raw data, or you find a mistake in previous code, or someone else (for example, trainers make some stupid error when deisnging the codes) tells you to do something different. Therefore, it is very important to make sure your R codes can be rerun easily - which is termed reproducible. Here are some tips:
   1. Workflows: the first part of R codes (1) explain the script, dates, author and version (2) library and set working directory (3) take in a csv as input data (4) set options or parameters; This is the part that you should customize yourself. The second part should check column type of the input dataframe, or do some descriptive stats, and format data accordingly. The third part is then analysis and calculation. The last part is plotting.
   2. backup

#######################

**Commençons avec l’analyse des pièges à litière**

# Préparation

1. Installez R et R studio
2. Installez le paquet R tidyverse (qui contient dplyr, ggplot et autres) et zoo
3. Copiez ce document dans votre dossier “drive” afin de le modifier comme bon vous semble. N’hésitez pas à prendre des notes et surlignez les points que vous avez complétés au fur et à mesure de votre avancement.
4. Si vous avez des questions, vous pouvez envoyer un message directement au formateur ou laisser une question dans la liste Q&A dans le document partagé sur le drive.

Organisation des fichiers et des méta-données

1. Trouvez dans l’explorateur votre fichier et le drive partagé
2. Transférer votre fichier contenant les données de dépôt de litière dans votre dossier de travail et nommez le selon le format (flf\_20190201\_ACJ.csv”, pour flf = “fine litter fall” (chute de litière fine), 20190201 - la date à laquelle le fichier a été transféré.
3. Inspirez vous du fichier flf\_20190201\_example.csv dans le drive, et encodez vos données pour qu’elles suivent la même organisation.
4. Faites attention à l’individualité de chaque unité d’observation. Dans le fichier exemple, il y a une colonne “unique\_id” qui est un nom qui caractérise chaque combinaison de plot + numéro de piège à litière + date. Il est fondamental que chaque ligne du tableau ait un idientifiant propre et unique. En utilisant ce type d’organisation des données, chaque ligne du tableau est appelée une entrée.
5. Pensez aux valeurs manquantes. Dans notre cas, chaque donnée manquante sera remplacée par les lettres NA, pour “Non-available”. Il est important de réaliser que une donnée manquante est différente d’un 0 (zéro). En aucun cas il ne faudra noter 0 si une donnée est manquante, car lors des analyses, R considère un 0 comme une valeur égale à zéro et pas comme l’absence de donnée.
6. En général, on peut remplir ce type de tableau en réalisant des copier-coller dans excel. Cependant, un grand nombre de copier-coller peut amener des erreurs. Il faudra donc être particulièrement vigilant pendant l’encodage et pendant l’organisation du fichier Excel. Il est possible (et souvent souhaitable) de modifier un tableau de données dans R pour le mettre au format standard. N’hésitez pas à nous contacter si vous voulez en discuter.

# Essayez R

1. Vous voyez maintenant 2 fichiers R dans le Drive partagé “fine\_litter\_fall : “finelitterfall\_teaching.rmd” est le fichier R principal que nous allons utiliser et qui contient la plupart des fonctions. Le fichier “functions.r” contient quelques fonctions supplémentaires qui permettent de faire fonctionner le fichier principal.
2. Le code contenu dans ce fichier permet de convertir les données de litières collectées au niveau d’un piège à litière en quantité de litière au niveau du plot.
3. Nous vous recommandons la lecture du livre de Wickham 2017 - R for Data Science Import, Tidy, Transform, Visualize, and Model Data”, ainsi que les référnces supplémentaires pour en apprendre plus à propos de R et ses diverses fonctions, ainsi que RStudio. Vous pouvez aussi trouver diverses vidéos sur youtube (p.ex. <https://www.youtube.com/watch?v=BoDTNbP7_OQ>)

NPP calculation workflow

Stem\_NPP by dbh census

# Workflow:

1. Input data: if your plot is not on ForestPlot, example datasheet is XXX\_02\_census\_data\_long\_format.csv. This is ForestPlot’s standard input datasheet. You should check the ForestPlot website for column names and other instructions. You then use BIEN\_Database.R and Teach\_join.R to pull in wood density. Wood\_density sourced from BIEN, by species, or genus, or family, or plot\_mean
2. Input data: if your plot is on ForestPlot, download XXX\_02\_PlotDump.csv from forestplot, then feed it into Organize\_census\_data\_from\_plotDump.R, it will produce a long format ready for analysis. ForestPlot has already filled in wood density.
3. stem\_npp\_teaching\_20210917.R is the main calculation script
4. Correct change of POM (point of measurement)
5. Remove dead tree dbh
6. Choose one of the plots (run one plot at a time)
7. Height propagation
8. Calculate biomass of each tree by a selected allometric equation, NPP of each tree was calculated as the difference in biomass between two given years. In any of the two years, if dbh was not recorded, the NPP of such a tree between these two years would be NA. Sum of all tree NPP is the plot level NPP for the given two years.
9. Error propagation on plot level, by R package BIOMASS. Error was not calculated as standard error of mean, but error propagation as introduced in BIOMASS. NPP error was not calculated for a single tree, only calculated for the whole plot.
10. Extrapolate to coarse root NPP and biomass, use a root:shoot ratio (R/S)
11. Data organisation and visualization

# Detailed info:

## Parameter

carbon\_content\_of\_wood=0.456

## Correct change of POM

This was done by ForestPlot, we chose the ‘DBH4’ column which has considered the change of POM.

## Height propagation (to solve NA in height)

There are two methods:

1. If you have height for more than 50 trees in your plot, We first pick the year with lots of height measurement, build a glmm model representing dbh-tree height correlation with this year, and fill NA in this year. Now every tree in this year has a height estimate. Any height recorded except for this year will be removed. Then we move on to other years, let height increase or decrease proportionally to the change of corresponding dbh of a given tree. Apart from glmm, we also tried brm() in R if possible. Otherwise, just simply log based glm.

Using one year only will ensure that NPP was calculated based on the change of dbh. Height is a rough estimate and the change of height from year to year may mess up NPP. See height option 1.

1. If you don’t have much height measurements for any year, we delete all height data, and replace them with height estimates as described by Feldpauch et al. (2012). See height option 2.

## Calculate biomass and NPP of each tree

* Carbon content of wood is 45.6%
* Several equations in Chave2014 and Chave2005 could be chosen
* Once biomass of each dbh record was calculated. Two censuses was extracted (any two years of data), the change of biomass of each tree is NPP. A tree is not considered for NPP calculation if it does not exist in one of the census (due to being too small, or dead). Stem NPP of a plot is the sum NPP of all trees.

## Error propagation

* Bayesian inference (<https://doi.org/10.1111/2041-210X.12753>) with BIOMASS package
* Error propagation done in grab\_total\_error.R, in order to compute systematic and random error independently, I created error\_propogation\_based\_on\_BIOMASS.r
* We used standard error estimated by BIOMASS, but not the biomass of tree estimated by BIOMASS, although the estimate is very similar to our stem\_npp\_teaching\_20210917.R. Therefore, we grab standard error from BIOMASS and scale according to our own estimate of tree biomass.
* We first do the above for tree biomass, with consideration of all sources of error (Diameter measurement error; Wood density error; Tree height error; Allometric model error; carbon content error)
* We use sd from year1 only, because the sd\_year1 - sd\_year2 is sometimes negative, sometimes positive, due to mente carlo simulation. We are going to scale sd according to the actual stem biomass of the plot anyway. It is ok to use one year only.
* Then we try to propagate errors for NPP. It is incorrect to simply add biomass error to get NPP error because it is a mixture of systematic error and random error.
* Following the idea that if biomass of a tree was overestimated for a year, then it must be an overestimation in next year, we know systematic error of NPP = Wood density + Tree height (because we use high propagation) + carbon content + allometric equation
* Also know random error only source from dbh measurement. As we have found in this study, dbh measurement error is the main source of error for stem NPP, although it is trivial in plot stem biomass.
* The standard deviation of dbh measurement was set as 0.2cm, sd of height is 2 meters. Sd of wood density source from BEIN database.
* npp\_std\_random = sqrt(biomass\_year1\_random\_std^2 + biomass\_year2\_random\_std^2)
* npp\_std\_systematic=(biomass\_year2\_system\_std - biomass\_year1\_system\_std)
* sd\_NPP = sqrt(npp\_std\_random^2 + npp\_std\_systematic^2)

# Output:

## KOG\_04\_stem\_NPP\_plot.csv

* KOG\_04 is plot code,
* Plot means plot average datasheet
* npp\_per\_ha\_MgC\_year is the plot level npp
* The standard deviation of the above is npp\_std
* Woody\_npp = npp\_per\_ha\_MgC\_year \* (1+above\_below\_ground\_ratio), above\_below\_ground\_ratio is a ratio that you set in R script stem\_npp\_teaching\_20210917.R
* The standard deviation of Woody\_npp is not displayed, but Woody\_npp\_std = npp\_std \* (1+above\_below\_ground\_ratio)
* Npp was calculated based on the change of stem diameter from Census1\_year to census2\_year
* Number\_of\_tree\_year1, the number of tree recorded in year 1, but some of them might be excluded from NPP calculation if it does not exist in year 2
* biomass\_year1\_per\_ha\_MgC total alive stem biomass in a plot
* The standard deviation of the biomass\_year1\_per\_ha\_MgC is biomass\_year1\_std

## KOG\_04\_stem\_NPP\_finest.csv

* Finest represent the finest resolution of NPP (no averaging), for stem NPP, it is NPP per tree

Fine\_litter\_fall

# Workflow

1. Fine litter fall data was categorized into leaves, twigs, flowers, fruits, seeds, bromeliads, epiphytes, palm leaves, palm flowers, palm fruits and others. Users could add as many categories as they want, by having new columns named as: newClass\_g\_per\_trap (replace newClass with your own category)
2. Time interval (DaysBetween) between collection was calculated based on collection date (therefore, it is important to record the litter fall trap setup date and note litter fall at that date as 0)
3. Users were reminded to check outliers and strange time intervals, for example, if the litter fall collection pause for a year.
4. NPP was calculated as Biomass\_g\_per\_trap / DaysBetween \* Unit\_conversion
5. Unit conversion from g per trap to Mgc\_ha\_month, with carbon content as 49%
6. Standard error calculated as the mean of data points (for whatever group we are averaging).

# Output files

XXX\_fine\_litter\_fall\_NPP\_finest.csv

1. The number of rows in XXX\_fine\_litter\_fall\_NPP\_finest.csv must equal to the number of rows in your input files
2. Plot\_code: same as your input file
3. Year month day date: same as your input file, the date when the sample was collected
4. Litterfall\_trap\_num, sub\_plot\_code, litterfall\_trap\_size\_m2, same as your input files
5. \_MgC\_ha\_month was converted from g\_per\_trap, which is the corresponding NPP given the DaysBetween
6. Quality\_code, comments, same as your input files

Ingrowth core

# Workflow

1. Ingrowth core data was sometime organised into different depth interval
2. Normally, four time steps were adopted for root estimate (see paper Metcalfe), however, for some plot, only one time step was used because all roots could be easily found in 10 minutes.
3. If root mass was recorded as the weight that a person found during a time step, we will correct it to accumulated mass.

Branch NPP by coarse woody debris sample

# Workflow

1. Size class (as you can see in cwd meta\_data.csv) was not considered in this script.
2. We first consider decay class by customizing decomposed\_wd\_g\_cm3 and Live\_wood\_density, for each decay class, we specify a wood density of sample after decomposition (the former one) and the wood density when the sample was alive in canopy (the later one). Then we could use WD\_ConversionFactorToAlive = decomposed\_wd\_g\_cm3/Live\_wood\_density to back calculate the dry mass of small branch from the somewhat decomposed sample we have collected
3. It is possible that there are some big branches that could not be weighted, for which diameter and length should be recorded. We calculate the dry mass of it by Volumn\_cm3\*decomposed\_wd\_g\_cm3 (see Branch\_npp\_teaching.R 4. estimate the weight of big branch from its size)
4. It is possible that there are too many small branches which require an excessive amount of oven time. In which case, only parts of the sample was sent to oven and moisture content was estimated. (see Branch\_npp\_teaching.R 5. weight of small branch from moisture content)
5. In 6.From biomass to NPP, we loop through each transect. The first record should be is\_stock = Y, which is considered necromass stock. The following record should be NPP. The user could exclude some branches (for example, from dead tree or it is a stem etc) by setting include\_as\_branch\_npp to N. After excluding, the remaining records will be used for NPP calculation, by NPP\_g\_day = dry\_mass\_when\_alive / DaysBetween. DaysBetween could be calculated by looking for the last record date (or it could be manually specified). Therefore, it is very important that for any transect with no branches, dry mass was recorded as 0, instead of NA.

1. Bug is an error in R that causes it to produce an incorrect or unexpected result. Sometime you will see warning as well, which indicates that the computer can still produce result but the software designer believes that there is something weird and your attention is needed [↑](#footnote-ref-0)