Proceedings of the second training on R for forest statistics and modelling

Bangladesh Forest Department

31 January – 04 February 2016

UN-REDD PROGRAMME

Food and Agriculture Organization of the United Nations

UNDP

UN environment

The UN-REDD Programme, implemented by FAO, UNDP and UNEP, has two components: (i) assisting in developing countries to prepare and implement national REDD strategies and mechanisms; (ii) supporting the development of normative solutions and standardized approaches based on sound science for a REDD instrument linked with the UNFCCC. The programme helps empower countries to manage their REDD processes and will facilitate access to financial and technical assistance tailored to the specific needs of the countries.

The application of UNDP, UNEP and FAO rights-based and participatory approaches will also help ensure the rights of indigenous and forest-dwelling people are protected and the active involvement of local communities and relevant stakeholders and institutions in the design and implementation of REDD plans.

The programme is implemented through the UN Joint Programmes modalities, enabling rapid initiation of programme implementation and channelling of funds for REDD efforts, building on the in-country presence of UN agencies as a crucial support structure for countries. The UN-REDD Programme encourage coordinated and collaborative UN support to countries, thus maximizing efficiencies and effectiveness of the organizations' collective input, consistent with the "One UN" approach advocated by UN members.

The UN-REDD Bangladesh National Program is implemented by the Bangladesh Forest Department under the leadership of Ministry of Environment and Forests. United Nations Development Program (UNDP) and Food and Agriculture Organization (FAO) are the two implementing partners.

Contacts:

Rakibul Hassan Mukul
Project Director
UN-REDD Bangladesh National Programme
Bangladesh Forest Department
Email: pd-unredd@bforest.gov.bd

Matieu Henry
Chief Technical Advisor
Food & Agriculture Organization of the United Nations (FAO)
Email: matieu.henry@fao.org

Suggested Citation: **Sola, G. & Costello, L.** 2016. Proceedings of the second training on R for forest statistics and modelling. 31 January - 4 February 2016, Dhaka, Bangladesh Forest Department, Food and Agriculture Organization of the United Nations.

Disclaimer

This report is designed to reflect the activities and progress related to UNJP/BGD/057/UNJ UN-REDD Bangladesh National Programme. It does not reflect the official position of the supporting international agencies including FAO and UNDP should not be used for official purposes. Should readers find any errors in the document or would like to provide comments for improving the quality they are encouraged to contact one of above contacts.

Executive Summary

Following a first training on R for forest statistics and modeling, this training aimed to remind the basic notions to write R commands and to present new functions useful for compiling forest data. The Bangladesh NFI was the main training dataset. Total eleven Participants (nine male and two female) could use it to perform a rough quality assessment and estimate forest aboveground biomass. Practice was emphasised all along the training course so that most of the participants started to feel confident with the code and functions learned, mainly graphs using the ggplot2 package and a group of useful functions. Many of the plots presented were suitable to preparing model quality graphs after developing a model. The next step is the introduction to model fitting of linear and non-linear functions with R.

Contents

Executive Summary **3**

1. Introduction **5**

2. Summary of the sessions 1 to 4 ..... **5**

2.1 Session 1: General Introduction 5

1.1 Session 2: basic objects in R 6

1.2 Session 3: Simple calculations on data frames 10

1.3 Session 4: Graphs with R using the ggplot2 package 12

2 From NFI to Forest biomass estimates (Sessions 5 to 8) **15**

2.1 Basic information on the NFI data 15

2.2 Visual quality control 16

2.2.1 h-dbh relationship to detect outliers 16

2.2.2 Tree code of the outliers (high trees with small dbh) 17

2.2.3 Trees with missing plot coordinates 18

2.3 Estimating tree and forest biomass 18

2.3.1 Height-diameter relationship 19

2.3.2 Tree biomass and carbon stock per FAO biomes and land use 20

3 Bonus: Carbon stock average per land use and species correction for the NFI **22**

3.1 Carbon stock estimates per land use 22

3.2 Correction of tree species names with the taxonomic name resolution services (TNRS). 24

4 Evaluation of the training and Conclusion **24**

Appendix 1. Agenda **25**

Appendix 2. Participants list **26**

Appendix 3. Evaluation results **27**

1. Introduction

The second R training for forest calculation and modeling was implemented from January 31st to February 4th 2016 at the Bangladesh Bureau of Statistics in Dhaka. It was a follow up training from a first training on R during which the basics of R, the different objects and few functions useful to analyze national forest inventory (NFI) data were presented. The second training aimed to remind and practice the basics and useful function and emphasize on graphs and selection inside data frames. It was divided into eight half days sessions with the following objectives:

- Remind the basics of R, in particular the different objects,
- Remind the operations on data frame with an emphasis on data selection
- Practice calculation and data selection inside data frames
- Develop graphs with the ggplot2 package
- Apply the functions and data selection tools to calculate forest aboveground biomass estimates based on the NFI data.

2. Summary of the sessions 1 to 4

The training was organized into 8 sessions to cover the following elements:

- General introduction to forest modeling and statistics
- R basic objects from simple numbers to data frames
- Operations on data frames, including reading tables and selecting subsets of data
- Graphs with ggplot2
- NFI data to Forest biomass (4 sessions)

2.1 Session 1: General Introduction

The presentation focused on the context of REDD+ and the need for allometric equations to relate trees' characteristics to their biomass and sum trees' biomass to plot and forest. The whole chain of calculations introduces errors (sampling, measurement, modeling, etc.) and statistical knowledge is required both to develop models and to assess the error introduced at the different calculation steps (Figure 1).

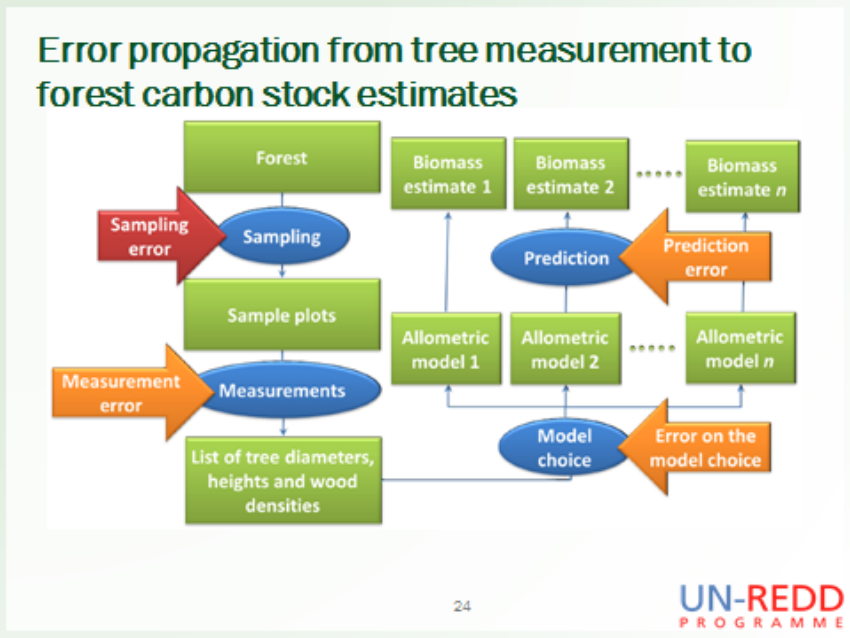


FIGURE 1. ERROR PROPAGATION FROM TREES TO FOREST BIOMASS ESTIMATES.

The participants to this training session attended the first training on R for forest statistics and modeling and their objectives was mainly to learn new tools, practice with R, apply it to forest data and ultimately develop new models. The questions and discussions were oriented towards modeling error and prediction as it is a key statistical concept.

1.1 Session 2: basic objects in R

When uploading a table into R, it is read as a **data frame**. Understanding what a data frame is, how R reads it and how to select data inside it is a key part of the training. The objectives of the first session were to present the graphic interface, the function help, basic formulas and then describe the main objects.

The different interfaces of Rstudio were reminded and the difference between text editor and calculation engine (console) (Figure 2). The help function was also presented: **help()**.

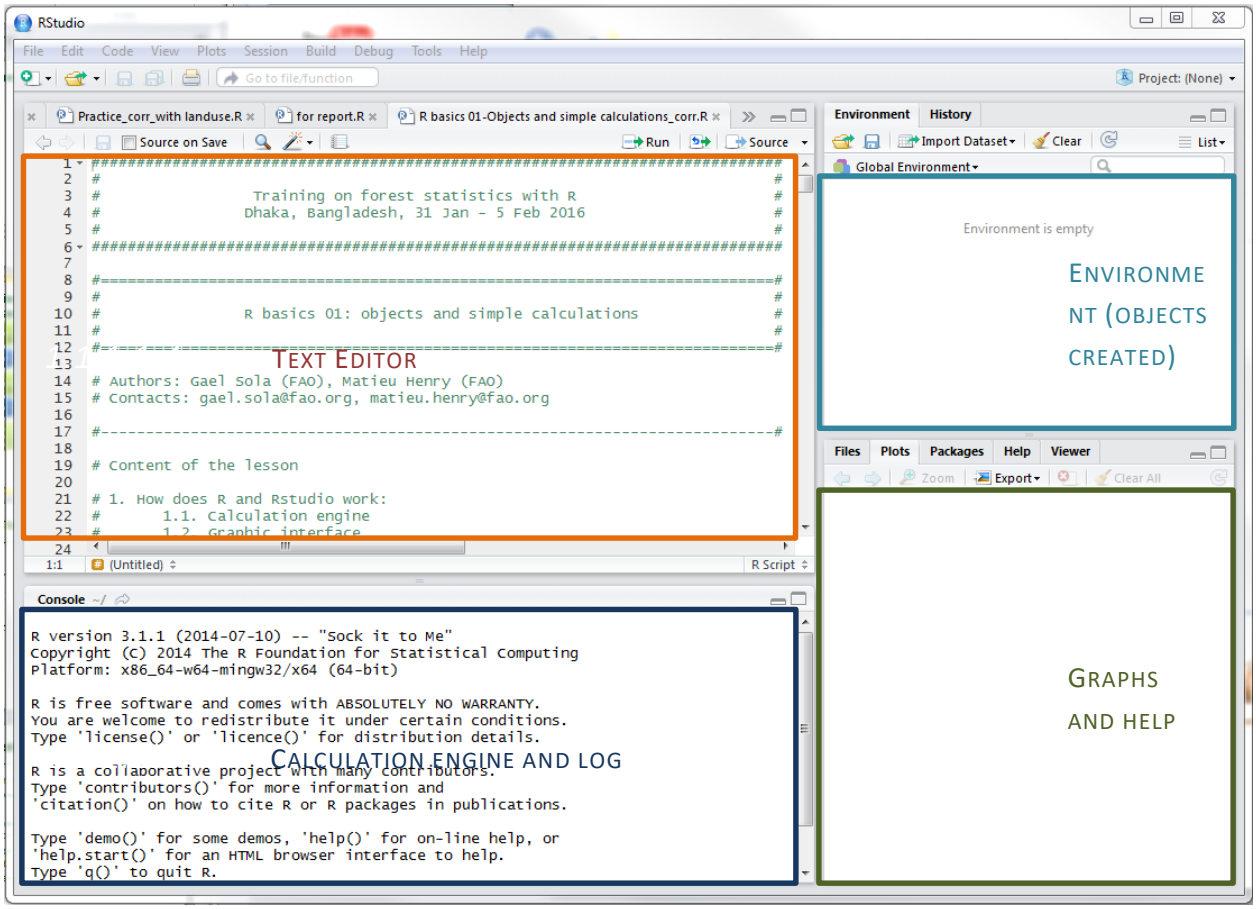


FIGURE 2. DEFAULT GRAPHIC INTERFACE IN RSTUDIO.

Simple formulas were presented such as, **log()** (natural logarithm), **log10()** (base 10 logarithm), **sqrt()** (square root), **exp()** (exponential).

```
> # Simple formulas
> 1+1
[1] 2
> 1-3
```

```
[1] -2
> exp(1)
[1] 2.718282
> log(exp(1))
[1] 1
> log10(100)
[1] 2
> sqrt(4)
[1] 2
> 24/12
[1] 2
> 5^3
[1] 125
> ((10 + 15) / 5) - 3*2
[1] -1
> pi
[1] 3.141593
```

Objects are defined by the name of the object and the data it contains. Creating an object with data is written with the sign "=" or "<-". In the following example, an object called "a" is created with the number 10 inside:

```
> a <- 10
> a
[1] 10
```

Objects can also contain text. Text must be written between brackets (""):

```
> b <- "Hello"
> b
[1] "Hello"
```

Besides text and numbers, R also has syntax for Booleans. The syntax is based on superior, inferior signs (< > <= >=) and is equal (==) or is different (!=). The answer to Boolean request is TRUE or FALSE. Booleans can also be stored as objects.

```
> ### Boolean
> x <- 3
> x==3
[1] TRUE
> x==4
[1] FALSE
> x!=4
[1] TRUE
> x>4
[1] FALSE
> c <- x==3
> c
[1] TRUE
```

Objects can also contain more than one element. Vectors contain one column of data (number, text, Booleans, etc.). To put together several numbers or text in one vector, the function **c()** (for concatenate) is used. To select one element inside a vector, the number of the row targeted can be written in between square brackets. For example **v[3]** corresponds to the third element of the vector **v**.

```
> ### Vectors
```

```
> v <- c(1,4,5.8)
> v
[1] 1.0 4.0 5.8
> v <- c("a","b","c")
> v
[1] "a" "b" "c"
> c <- 1:10
> c
 [1]  1  2  3  4  5  6  7  8  9 10
> # Data can be selected inside a vector
> c[3]
[1] 3
```

2.1.1.1.1.1 TIP 1: BRACKETS ARE IMPORTANT

The brackets are very important in R. In the Boolean example c is an object, and in the last example c() is the function concatenate and c[3] is the third element of the vector c.

Matrices can be created with the function **matrix()**. It creates a matrix with the specified number of columns and rows. To create a series of continuous integers between two numbers, the colon punctuation "**:**" can be used. In the following example an object **M** is created with continuous integers between 1 and 100 with 10 rows and 10 columns.

```
> M <- matrix(1:100, nrow=10)
> M
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    1   11   21   31   41   51   61   71   81    91
 [2,]    2   12   22   32   42   52   62   72   82    92
 [3,]    3   13   23   33   43   53   63   73   83    93
 [4,]    4   14   24   34   44   54   64   74   84    94
 [5,]    5   15   25   35   45   55   65   75   85    95
 [6,]    6   16   26   36   46   56   66   76   86    96
 [7,]    7   17   27   37   47   57   67   77   87    97
 [8,]    8   18   28   38   48   58   68   78   88    98
 [9,]    9   19   29   39   49   59   69   79   89    99
[10,]   10   20   30   40   50   60   70   80   90   100
```

Data can be selected inside matrices by specifying the number of the row and column in between square brackets, **M[a,b]** with **a** the number of the row and **b** the number of the column. One vector can be created from a matrix by selecting one entire row **M[a,]** or column **M[,b]**.

```
> M[1,2]
[1] 11
> M[9,]
 [1]  9 19 29 39 49 59 69 79 89 99
```

Matrices cannot contain both numbers and text for example and don't have column titles. List is a type of object that can store all kind of objects together, but in an unorganized way. Each object inside a list has a name and can be selected using the **$** sign: **List_name$object_name**.

```
> List1 <- list(a = 1:10,
+               b=c("a","b","c"),
+               c=matrix(1:25, nrow=5, byrow=T)
+               )
>
```

```
> List1
$a
 [1]  1  2  3  4  5  6  7  8  9 10

$b
[1] "a" "b" "c"

$c
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
[4,]   16   17   18   19   20
[5,]   21   22   23   24   25
> List1$a
 [1]  1  2  3  4  5  6  7  8  9 10
```

The most important object for forest data calculation is the data frame. A data frames is the combination of a matrix and a list. As such it has the structure of a matrix but it can contain both numbers and texts, and most importantly its columns have titles. To select an entire column inside a data frame, the number of the column can be entered in square bracket or the column title can be written after a $ sign.

```
> df <- data.frame(M[,c(1:3)])
> names(df) <- c("a","b","c")
> df
    a  b  c
1   1 11 21
2   2 12 22
3   3 13 23
4   4 14 24
5   5 15 25
6   6 16 26
7   7 17 27
8   8 18 28
9   9 19 29
10 10 20 30
> df$a
 [1]  1  2  3  4  5  6  7  8  9 10
> df[,1]
 [1]  1  2  3  4  5  6  7  8  9 10
```

New columns can easily be created as additional objects inside an existing data frame. In the following example a new column d is created in the data frame df with text.

```
> # Create a new column d with text
> df$d <- c("a","b","c","a","b","c","a","b","c","a")
> df
    a  b  c d
1   1 11 21 a
2   2 12 22 b
3   3 13 23 c
4   4 14 24 a
5   5 15 25 b
6   6 16 26 c
7   7 17 27 a
8   8 18 28 b
9   9 19 29 c
10 10 20 30 a
```

1.2 Session 3: Simple calculations on data frames

The objectives of this session were to introduce forest inventory data, read it with R and perform a first series of simple data selection and calculations. R can read any type of tables, but the functions to read directly .xls or .xlsx might generate unexpected errors. The simplest way to read tables with R is to read them from text (.txt) files. In text files there are no columns but separators. Separators can be comas, tabulations, semicolons, colons, etc. The other possible sources of errors are the special characters, the empty cells and the spaces between words in the title column.

TIPS: Preparing a table for R.

When preparing a table for R, users can open it with OpenOffice Calc or MS Excel and verify that:

- **There are no empty cells in the dataset. If there is NA should be entered.**
- **There are no special characters and punctuation signs inside cells (, # / etc., the #N/A from MS Excel should be replaced by NA).**
- **There are no spaces in the column titles (rename "species name" into "species_name" for example).**

Once the table is clean it can be saved as .txt file with separators: tabulation (from Menu > Save as…).

TIPS: R script starter.

When starting a script in R, three actions are important to avoid possible errors during the calculations:

1. **Clean the environment; remove the existing objects and graphs that are not related to the new script.**
2. **Launch additional libraries that are part of the R core but will be used during the calculations.**
3. **Set the working directory. The folder on the computer containing the data should be specified.**

The following command lines can be used at the beginning of each script:

```
#------------------------------------------------------------------------#
#                                    STARTER
#------------------------------------------------------------------------#

# Erase memory and graphs
rm(list=ls())
dev.off() # go to the console and press Ctrl+L

#install.packages("")
library(ggplot2)
library(nlme)

# Check working directory
getwd()

# If necessary set the working directory (copy paste from Windows explorer addr
ess bar + change the \ with \\ or /)
setwd("C:\\Mission 02-2016 Bangladesh R training\\Data")
```

The function used in R to read text files is **read.table()**. Users should specify the name of the file, the type of separators, if the first line corresponds to column titles and in the case of forestry data, add the command **stringsAsFactors = FALSE** to avoid converting all the text to factors (hierarchical text). Once the table has been successfully loaded in the environment three basic functions display the data or a summary of it, **str()**,

summary(). These functions can be applied to the whole table or only to one column. The **View()** function (First letter is uppercase) display the data frame in a read only table. Basic information for each column can be calculated with the functions **min()**, **max()**, **mean()**, **sd()** (for standard deviation).

```
> tree <- read.table("biomass_sample_modif.txt", header=T,sep="\t",stringsAsFac
tors = FALSE)
> str(tree)
'data.frame':  220 obs. of  18 variables:
 $ project      : chr  "RainForest-A" "RainForest-A" "RainForest-A" "RainForest
-A" ...
 $ region       : int  1 1 1 1 1 1 1 1 1 1 ...
 $ plot         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ tree         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ tree_id      : chr  "R01P01T01" "R01P01T02" "R01P01T03" "R01P01T04" ...
 $ sc_name      : int  52 11 9 65 37 31 58 57 9 52 ...
 $ family       : chr  "B" "G" "T" "Z" ...
 $ dbh_cm       : num  6.5 68.4 9.1 16.4 17.2 10.6 42.8 6.2 11.4 10.2 ...
 $ crown_d_m    : num  3 16 2.8 4.4 5.8 3.5 6.8 1.8 3.4 3 ...
 $ crown_area_m2: num  7.07 201.06 6.16 15.21 26.42 ...
 $ h_m          : num  6.2 26.4 7.2 18.4 16.3 12.9 31.5 9.5 9.3 10.9 ...
 $ v_m3         : num  0.02 4.9 0.03 0.21 0.19 0.07 3.02 0.02 0.05 0.05 ...
 $ b_branch_kg  : num  4.79 1459.5 4.24 21.22 36.47 ...
 $ b_leaves_kg  : num  1.52 118.82 1.03 1.13 12.27 ...
 $ b_stem_kg    : num  8.36 2316.14 14.57 78.08 87.33 ...
 $ agb_kg       : num  14.7 3894.5 19.8 100.4 136.1 ...
 $ wd_gcm3      : num  0.59 0.57 0.49 0.57 0.59 0.57 0.63 0.59 0.54 0.59 ...
 $ bef          : num  1.98 1.88 1.72 1.71 1.66 1.55 1.51 1.5 1.47 1.47 ...
> summary(tree$dbh_cm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.90   11.71   20.86   27.07   37.58   87.70
> mean(tree$dbh_cm)
[1] 27.07186
```

For the text variables, the function **table()** calculates the number of rows (each row is one tree in our examples) for each different word. For categories and hierarchical categories, the text can be transformed into factors.

```
> summary(tree$project)
   Length     Class      Mode
      220 character character
> table(tree$project)
RainForest-A RainForest-B
         110          110
> table(tree$project, tree$region)
                1   2
  RainForest-A 110   0
  RainForest-B   0 110
> tree$project <- factor(tree$project)
> summary(tree$project)
RainForest-A RainForest-B
         110          110
```

New columns can be created using the data frame syntax presented in session 2:

data_frame_name$new_column_name

For example:

```
> tree$count <- 1
> summary(tree$count)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       1       1       1       1       1
> tree$d2h_m3 <- (tree$dbh_cm/100)^2*tree$h_m
```

A key operation on data frames is selecting data inside a data frame. To select one column, naming it after the sign dollar and the name of the data frame is enough. To select several columns, the concatenate function **c()** should be used. Selecting one column creates a vector, selecting several columns creates another data frame.

```
> tree$dbh_cm
  [1]  6.50 68.40  9.10 16.40 17.20 10.60 42.80  6.20 11.40 10.20 49.50  6.80 1
2.00  9.60
 [15]  7.00  7.20 15.70 13.60 10.20 14.50  9.30 87.70 16.00 18.80  8.80 23.40 8
2.40 55.00
[…]
> tree[,c("dbh_cm","h_m")]
    dbh_cm   h_m
1     6.50  6.20
2    68.40 26.40
3     9.10  7.20
4    16.40 18.40
5    17.20 16.30
6    10.60 12.90
7    42.80 31.50
[…]
```

To select specific data and rows is different as rows don't have names. The data selection is based on the characteristics of the data, such as belonging to a group for text characteristics or belonging to a group of values or a range of values. This selection is based on the function **which()** and this function is placed inside the square brackets, to specify that the selection is on the rows. The syntax for the Boolean is used to specify is equal, is different, etc.

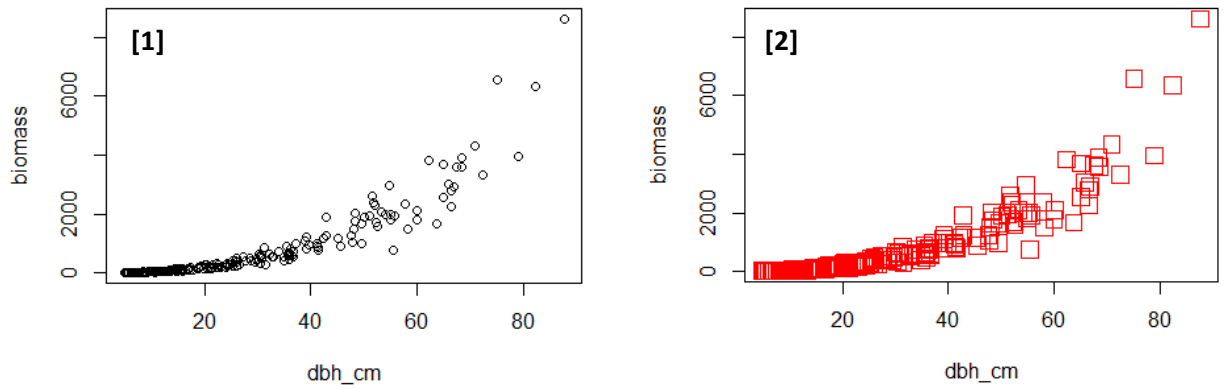
```
> tree_sub <- tree[which(tree$dbh_cm>=50),]
> dim(tree_sub)
[1] 35 20
> summary(tree_sub$dbh_cm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  50.11   54.40   60.09   62.15   67.18   87.70
> tree_sub <- tree[which(tree$project=="RainForest-A"),]
> max(tree_sub$dbh_cm)
[1] 87.7
```

The selection tool is particularly useful to detect outliers and focus on a group of data inside a potentially very large table.

1.3 Session 4: Graphs with R using the ggplot2 package

The basic scatterplot function in R is named **plot()**. It can be used to create simple graphs and the size, shape and colors of the dots can be changed manually with the command **cex**, **pch** and **col** respectively.

```
> tree$biomass <- tree$agb_kg
> plot(biomass~dbh_cm, data=tree) [1]
> plot(biomass~dbh_cm, data=tree, pch=22,cex=2, col='red') [2]
```



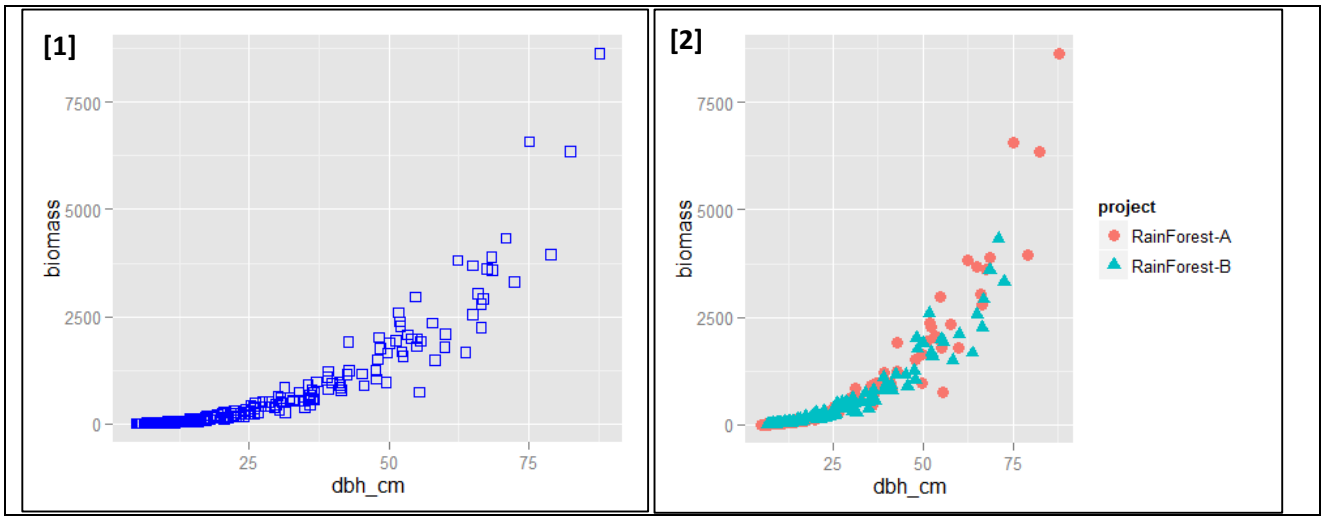
One of the main limitation of the plot() function is that it is quite difficult to change the color of the dots based on one variable. In the previous example, it would be useful to change the color based on the project to see if it influences the h – dbh relationship. The ggplot2 package introduces a new syntax for creating plots using several functions to define:

- The data used: **ggplot()**.
- The type of graph: **geom_point()**, **geom_line()**, **geom_boxplot()**, **geom_abline()**, etc.
- The variables to be plotted and used to change the color, size or shape of the geometry used, called aesthetics **aes()**.
- Additional functions to change the display (axis, graphs shape, axis labels, title, etc.).

The aesthetics can be placed inside the **ggplot()** function or inside the **geom_xxxx()** function. The other functions are summed. To avoid representing the command as a long line of functions, an object **p** is created and each new function is added to **p**. Inside the aesthetics, **pch**, **cex** and **colour** are based on existing variables and will change depending on these variables. Outside the aesthetics, **pch**, **cex** and **col** are based on one value and it will apply to the whole dataset.

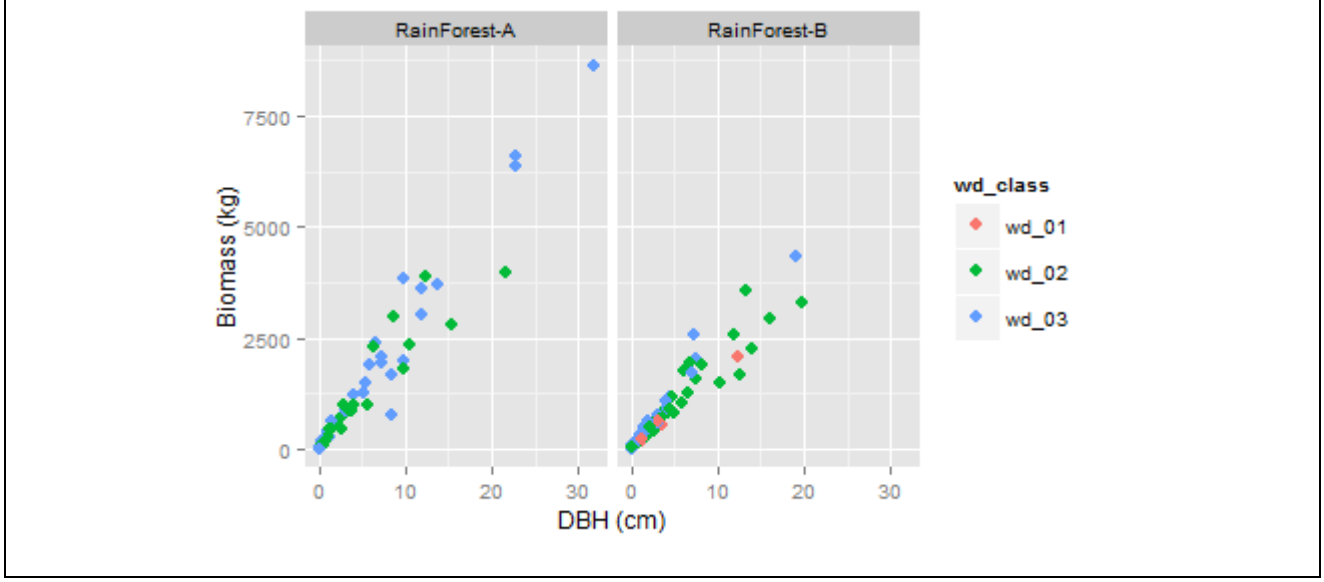
```
> library(ggplot2)
> p <- ggplot(tree, aes(x=dbh_cm, y=biomass))
> p <- p + geom_point(cex=3, pch=22,col="blue")
> p
[1]

> p <- ggplot(tree, aes(x=dbh_cm, y=biomass, pch=project, colour=project))
> p <- p + geom_point(cex=3)
> p
[2]
```



A more complex graph includes two plots in one figure, axis labels and dot color based on another variable:

```
> p <- ggplot(tree, aes(x=d2h_m3, y=biomass, colour=wd_class))
> p <- p + geom_point(cex=3)
> p <- p + xlab("DBH (cm)") + ylab("Biomass (kg)")
> p <- p + facet_wrap(~project)
> p
```



2 From NFI to Forest biomass estimates (Sessions 5 to 8)

After the introduction to the different types of objects in R, the knowledge on manipulating data frames and preparing graphs was used to analyze the data from Bangladesh National Forest Inventory (NFI) 2005.

2.1 Basic information on the NFI data

The functions **length()** and **unique()** were used to calculate the number of several variables (Table 1). Example:

```
> length(unique(tree$land_use))
[1] 23
```

Variable	Total number
trees	38993
tract	251
plot	702
land use	23
family	64
genus	209
species	295 (273 after correction)[1]

TABLE 1. BASIC INFORMATION ON THE NFI DATA.

The function **table()** was used to calculate the number of tree per land use (Table 2).

```
> table(tree$land_use)

  CA0   CA1   CA2   CP0   CP1   CP2    Fa    FB    FH    FM    HA    PL
 1365  1905  1084    23    39    52    41   934  3031  3892    26   410
   PM    PS    RL    Sh   SR0   SR1   SR2    SU   WHB    WP    WR
    7   140     2     3   131  7853 17493   358    12   166    26
```

Land use code	Land use	Number of trees
CA0	Annual crop without or with low tree cover	1365
CA1	Annual crop with tree cover; 0,1 – 0,5 ha	1905
CA2	Annual crop with tree cover; > 0.5 ha	1084
CP0	Perennial crop without or with low tree cover	23
CP1	Perennial crop with tree cover; 0,1 – 0,5 ha	39
CP2	Perennial crop with tree cover; > 0.5 ha	52
Fa	Wooded land with shifting cultivation (fallow)	41
FB	Bamboo or mixed Bamboo/broad-leaved forest	934
FH	Hill forest	3031
FM	Mangrove forest (saltwater)	3892
HA	Highways and other artificial areas	26
PL	Long rotation forest plantation: 40-60 years (Teak, Dipterocarp, Sal, Jam, etc.)	410

[1] See the section on species for more information on the species correction.

PM	Mangrove plantation	7
PS	Short/medium rotation forest plantation: 10-20 years (Acacia, Eucalyptus, Gamar, etc.)	140
RL	Rangeland/Pasture	2
Sh	Shrubs (or shrubs/trees)	3
SR0	Rural settlements without or with low tree cover	131
SR1	Rural settlements with tree cover; 0,1 – 0,5 ha	7853
SR2	Rural settlements with tree cover; >0.5 ha	17493
SU	Urban settlements	358
WHB	Haor & Baor	12
WP	Ponds	166
WR	Rivers	26

TABLE 2. NUMBER OF TREE PER LAND USE.

The function summary provides basic information on numerical variables.

```
> summary(tree$dbh)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   12.00   17.00   20.12   25.00  231.00
> summary(tree$h)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   6.000   8.000   8.931  11.000 123.000
```

⇨ The results show potential errors (tree h bigger than 60 m) and missing values (dbh or h equal to 0).

2.2 Visual quality control

2.2.1 h-dbh relationship to detect outliers

To understand better the relation between dbh and h and detect outliers the graph representing h against dbh is created (Figure 3). A group of trees is very different from the others, with a very big height but small dbh.

```
> p <- ggplot(tree)
> p <- p + geom_point(aes(x=dbh, y=h), cex=1.5)
> p <- p + xlab("Diameter at breast height (cm)") + ylab("Tree height (m)")
> p
```

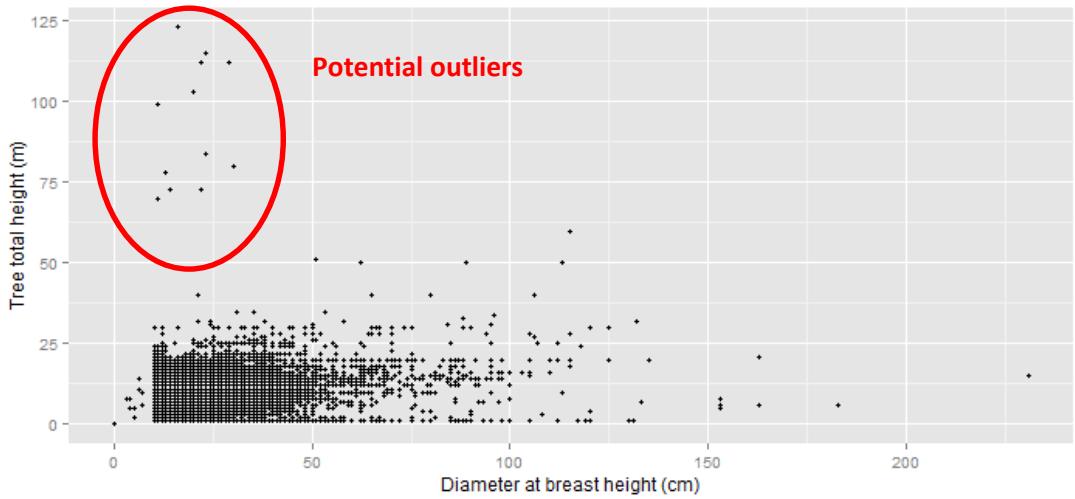


FIGURE 3. TREE HEIGHT-DIAMETER RELATIONSHIP TO IDENTIFY OUTLIERS.

2.2.2 Tree code of the outliers (high trees with small dbh)

A subset of the data for which tree dbh is smaller than 50 cm and tree h is bigger than 60 m can reveal the code of the outliers to control the field forms.

```
> tree[which(tree$h > 60 & tree$dbh < 50),c("unique_id","tract_id","plot_id
","tree_id","dbh","h")]
      unique_id tract_id plot_id tree_id dbh   h
182         182       33       2      26  20 103
338         338       33       3      99  29 112
3183       3183       25       3      14  16 123
4501       4501       63       4      41  23  84
13765     13765      119       2      15  22 112
19739     19739       49       3      33  23 115
20026     20026       49       3     320  11  99
24217     24217      276       2       7  13  78
26018     26018       95       2      43  11  70
28541     28541      155       1      76  14  73
29531     29531      133       4      16  22  73
34208     34208      147       4       1  30  80
```

- ⇨ As the field forms were not accessible, these trees were removed from the data to avoid potential overestimating of the tree, plot and forest biomass. Additionally, the trees with a dbh smaller than 10 cm were removed as they should not have been measured and the trees with a height smaller than 1.3 m were also removed.

```
> tree <- tree[which(tree$h <= 60 | tree$dbh >= 50),]
> tree <- tree[which(tree$dbh >= 10),]
> tree <- tree[which(tree$h > 1.3),]
```

The updated information on tree dbh and h is provided in Table 3 and Figure 4.

Variable	Number of trees	min	average	max
dbh	38417	10	19.95	231
h	38417	2	9.019	60

TABLE 3. UPDATED INFORMATION ON TREE DBH AND H.

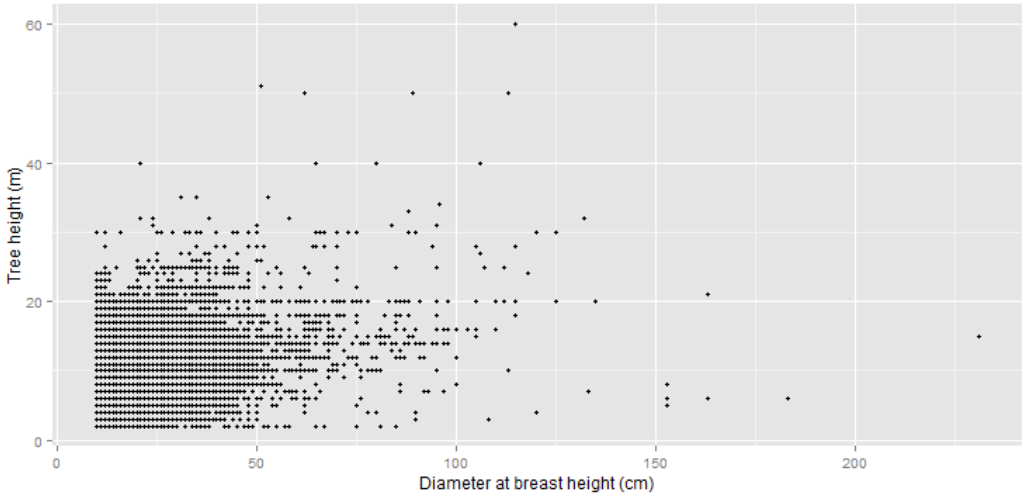


FIGURE 4. TREE H AGAINST DBH AFTER CORRECTION.

2.2.3 Trees with missing plot coordinates

Typos and errors in entering plot coordinates can be detected in the graphic representation of plot location. The graph revealed that at least one plot has a coordinates 0. It can be corrected by replacing these values with NA (meaning Not Available) so that R doesn't include it in the graph (Figure 5).

```
> p <- ggplot(tree)
> p <- p + geom_point(aes(x=plot_x, y=plot_y))
> p
> tree[which(tree$plot_x == 0),]$plot_x <- NA
> tree[which(tree$plot_y == 0),]$plot_y <- NA
```

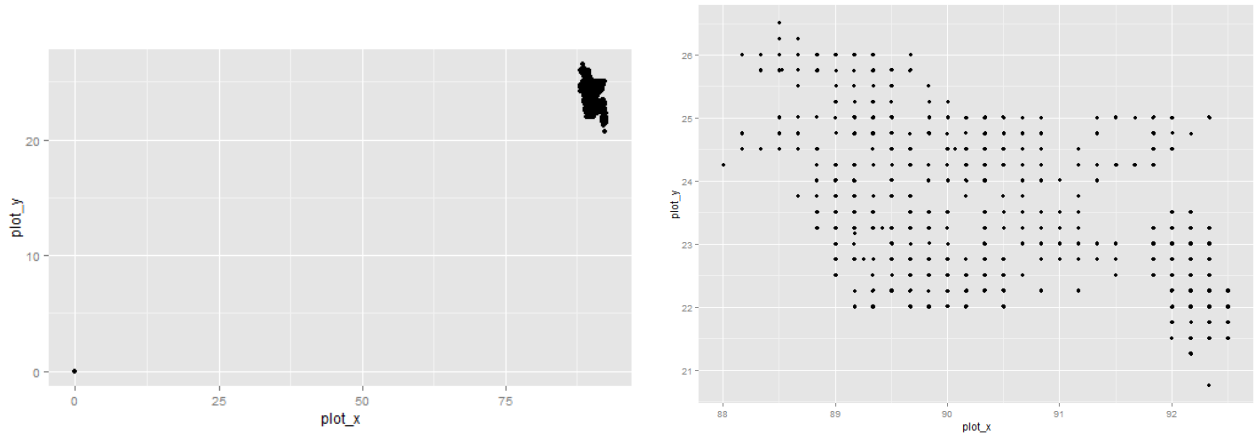


FIGURE 5. PLOT LOCATION BEFORE AND AFTER REMOVING THE ERRORS.

2.3 Estimating tree and forest biomass

This exercise is composed by 7 sections (Figure 6). The objective was to use existing forest inventory data, i.e. tree dbh, h and species, to calculate tree and forest carbon stock. Due to time constraints the part 1 to 3 were not implemented and the exercise started at step 4, estimating tree height by applying a tree h-dbh model. The part 5 was also not implemented as developing models training was not provided yet.

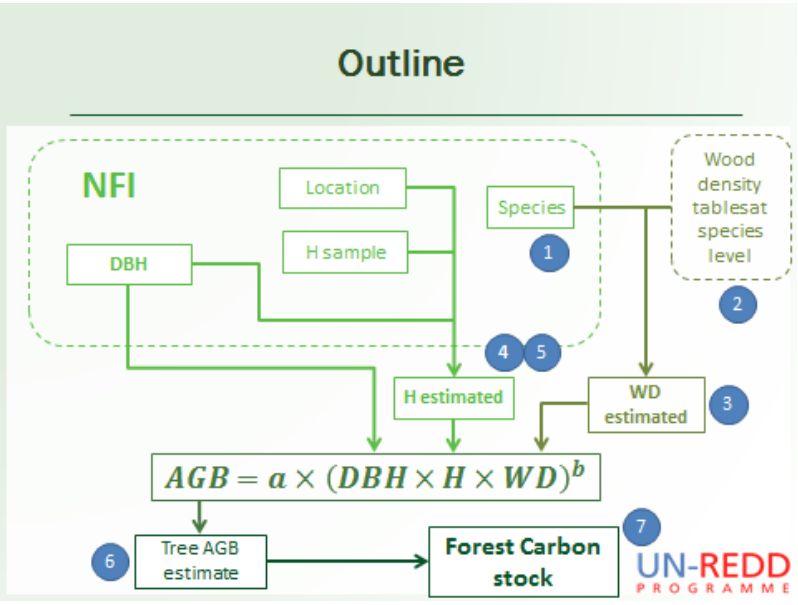


FIGURE 6. OUTLINE OF THE EXERCISE FROM NFI TO FOREST BIOMASS ESTIMATES.

2.3.1 Height-diameter relationship

In the Bangladesh NFI, tree height was measured for all trees, but in future inventories it might only be measured for few trees per plots. It is generally accepted that measuring the height of all trees is very time consuming and leads to more inaccuracy as it is often quickly estimated rather than carefully measured. This section presents the code and results for estimating tree height (h) with pan-tropical h-dbh relationships based on FAO biomes, tree dbh (in cm) and plot maximum height (hmax in m):

- Tropical dry forest: $h = 1.3 + (5.27 + 0.5*hmax)*exp(-2.06*exp(-0.07*dbh))$
- Tropical moist deciduous forest: $h = 1.3 + (6.37 + 0.5*hmax)*exp(-1.43*exp(-0.05*dbh))$
- Tropical mountain system: $h = 1.3 + (4.29 + 0.5*hmax)*exp(-2.35*exp(-0.09*dbh))$
- Tropical rainforest: $h = 1.3 + (11.66 + 0.5*hmax)*exp(-1.72*exp(-0.04*dbh))$

To calculate the maximum height per plot the functions **aggregate()** and **merge()** were used. The first function aimed to create a table containing the maximum height for each plot and the second one to merge the result to the tree table.

```
> h_plot <- aggregate(h ~ unique_plot_id, data=tree, FUN=max, na.rm=TRUE)
> # Verify that the table created as the same number of rows than the number of
 plots in the table tree
> dim(h_plot)
[1] 701   2
> length(unique(tree$unique_plot_id))
[1] 701
> # Change the name of the second column in h_plot
> names(h_plot)[2] <- "hmax"
> # Merge the h_plot table with tree table to associate the plot hmax to each t
ree
> tree <- merge(tree, h_plot, by= "unique_plot_id")
> # Cross check the result for one plot, for example the plot "62_2"
> tree[which(tree$unique_plot_id == "62_2"),c("h","hmax")]
       h hmax
31063  9   11
31064  4   11
31065 11   11
```

The package “rgdal” was used to collect the FAO biome for each plot based on the plot coordinate and the FAO Biome shapefile. This step can also be implemented with a GIS software. The function **ifelse()** was then used to associate to each tree the adequate model depending on the FAO biomes it is located in.

```
> tree$h_est <- ifelse(tree$fao_biome == "Tropical rainforest",
+                      1.3+(11.66+0.5*tree$hmax)*exp(-1.72*exp(-0.04*tree$dbh)),
+                      1.3+(6.37+0.5*tree$hmax)*exp(-1.43*exp(-0.05*tree$dbh))
+                      )
```

As a result each plot had a slightly different model depending on its maximum tree height and models differed a bit between FAO biomes (Figure 7). The estimated tree heights are not very different from the measured ones. The overall bias is 8.4 %.

```
> tree$h_err <- tree$h_est - tree$h
```

```
> bias <- sum(tree$h_err)/sum(tree$h)*100
> bias
[1] 8.432961
```

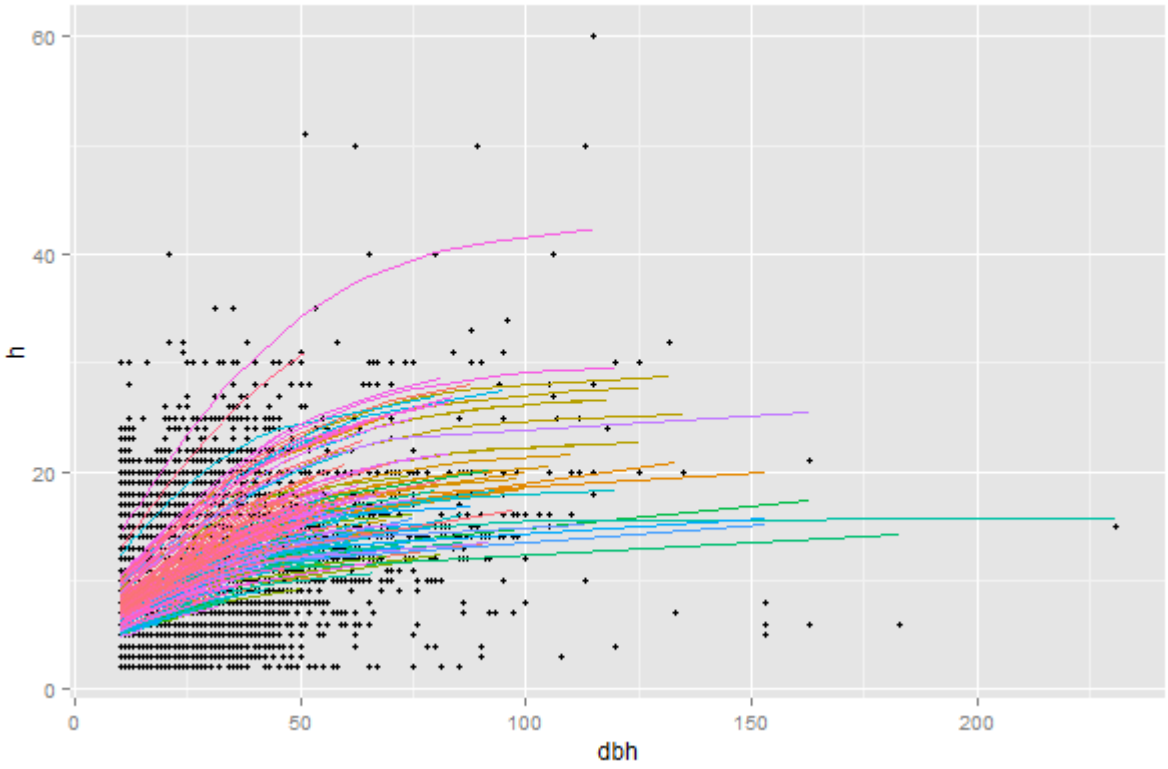


FIGURE 7. MEASURED (DOTS) AND ESTIMATES (LINES) TREE HEIGHT AGAINST DIAMETER AT BEAT HEIGHT.

2.3.2 Tree biomass and carbon stock per FAO biomes and land use

Wood density values were attributed to each tree based on existing wood density data and tree species. The wood density data comes from the global wood density database[2]. This part of the exercise was not implemented due to time constraints. Once tree height and wood density were estimated for all trees aboveground biomass (agb in kg) was estimated with the following formula, from Chave et al. 2014[3]:

- agb = 0.0673*(dbh^2*h*wd)^0.976

With dbh in cm, h in m, wd in g/cm3 and agb in kg.

As in the case of the Bangladesh NFI, h was measured for all the trees and it was used instead of the estimated tree height (h_es).

---

[2] Chave J, Coomes DA, Jansen S, Lewis SL, Swenson NG, Zanne AE (2009) Towards a worldwide wood economics spectrum. Ecology Letters 12(4): 351-366. http://dx.doi.org/10.1111/j.1461-0248.2009.01285.x

Zanne AE, Lopez-Gonzalez G, Coomes DA, Ilic J, Jansen S, Lewis SL, Miller RB, Swenson NG, Wiemann MC, Chave J (2009) Data from: Towards a worldwide wood economics spectrum. Dryad Digital Repository. http://dx.doi.org/10.5061/dryad.234

[3] Chave et al. Improved allometric models to estimate the above ground biomass of tropical trees. 2014. Global Change Biology. DOI: 10.1111/gcb.12629

```
> tree$agb_kg <- 0.0673*(tree$dbh^2*tree$h*tree$wd)^0.976
> summary(tree$agb_kg)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   3.422   37.910   70.520  166.900  172.100 18180.000
```

The next step was to sum the tree biomass for each plot with the function **aggregate()** and convert it to tons per ha. Then the information on FAO biome and basal area class were added to the **plotdata** table using the **unique()** and **merge()** functions. Finally the data represented in a boxplot graph (Figure 8).

```
> plotdata<- aggregate(agb_kg~unique_plot_id, data=tree, FUN=sum, na.rm=TRUE)
> plotdata$agb_t_ha <- plotdata$agb_kg/(0.5*1000)
> summary(plotdata$agb_t_ha)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.03597  3.22000  9.92700 18.29000 21.10000 237.70000
> plotinfo <- unique(tree[,c("unique_plot_id", "fao_biome", "ba_class")])
> plotdata2 <- merge(plotdata, plotinfo, by="unique_plot_id")
>
> plotdata2 <- merge(plotdata, plotinfo, by="unique_plot_id")
> p <- ggplot(plotdata2)
> p <- p + geom_boxplot(aes(x=ba_class, y=agb_t_ha, colour=fao_biome))
> p <- p + xlab("Basal area class") + ylab("aboveground biomass (t/ha)")
> p <- p + theme(legend.position="none")
> p
```

NB: the command **theme(legend.position="none")** removes the legend from the graph to increase the space allocated to the figure. In the above example the legend is quite long (FAO biomes) and it was better to describe it in the figure title rather than inside the figure.

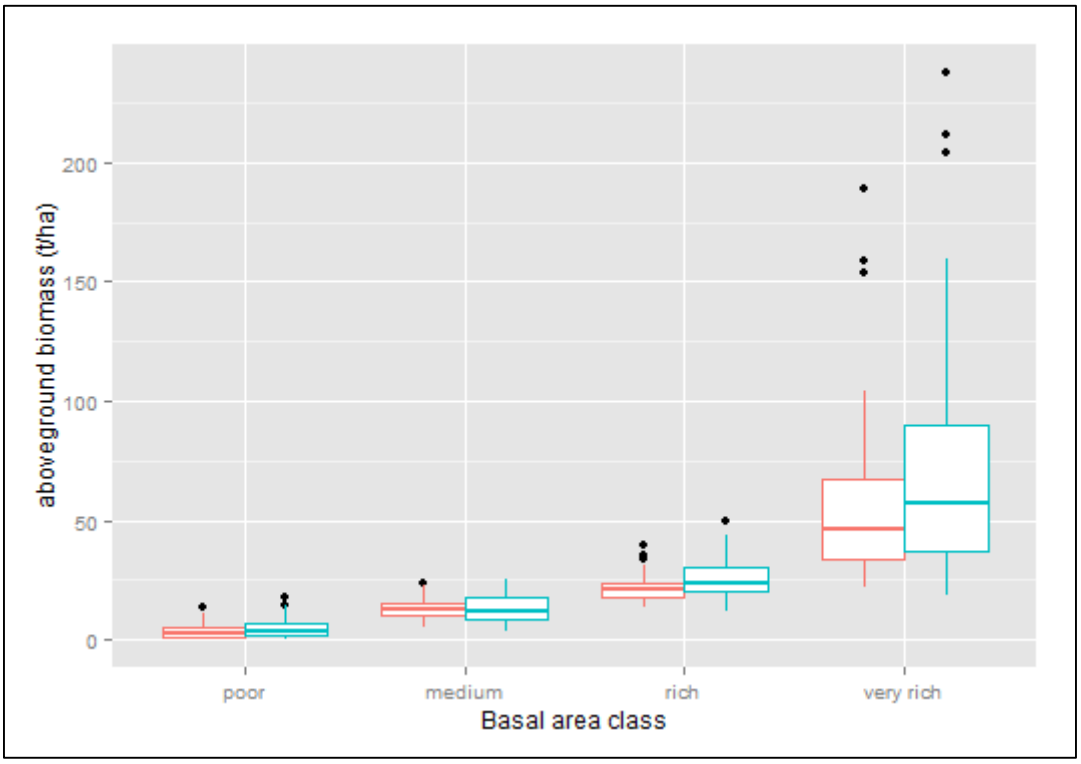


FIGURE 8. DISTRIBUTION OF PLOT CARBON STOCK PER BASAL AREA CLASS.

RED BOXPLOTS CORRESPOND TO INVENTORY PLOTS LOCATED IN TROPICAL MOIST DECIDUOUS FORESTS AND BLUE BOXPLOTS TO TROPICAL RAINFORESTS.

To calculate the average carbon stock per FAO biome and its standard deviation, the function **aggregate()** was used again.

```
> cstock_avg <- aggregate(agb_t_ha~fao_biome,data=plotdata2, FUN=mean, na.rm=TR
UE)
> names(cstock_avg)[2] <- "agb_t_ha_avg"
> cstock_sd <- aggregate(agb_t_ha~fao_biome,data=plotdata2, FUN=sd, na.rm=TRUE)
> names(cstock_sd)[2] <- "agb_t_ha_sd"
> cstock <- merge(cstock_avg, cstock_sd, by="fao_biome")
> cstock
                        fao_biome agb_t_ha_avg agb_t_ha_sd
1 Tropical moist deciduous forest     15.52901    21.22961
2             Tropical rainforest     23.76104    35.71705
```

As a conclusion forest carbon stock were not significantly different in the two FAO biomes. On average Tropical rainforest had a slightly higher carbon stock the standard deviations were of the same order than the estimates meaning that these differences were not significant and the overall average of 18.29 t biomass / ha could be used nationwide (see the result of the command **summary(plotdata$agb_t_ha)**). Other classifications might be better to separate the forest carbon stock in more meaningful categories in terms of biomass.

3 Bonus: Carbon stock average per land use and species correction for the NFI

3.1 Carbon stock estimates per land use

Aboveground biomass estimates per land use category were not calculated during the training as one plot can have two different land uses and the information on the area for each land use inside one plot was not available. To overcome this issue one main land use was attributed to each tree based on the number of tree per land use and per plot. The functions **aggregate()**, **merge()** and **unique()** were used to calculate the highest number of tree per land use inside each plot, use it as an identifier for the associated land use and expand this land use code to al trees of the same plot. The result was a table plotdata3 resulting from merging the information of the main land use per plot and the biomass per plot calculated earlier. The information was then aggregated to land use level to calculate the number of plots, the average and standard deviation of plot aboveground biomass (Table 4 and Figure 9).

```
> cstock_avg_lu <- aggregate(agb_t_ha~land_use_main,data=plotdata3, FUN=mean, n
a.rm=TRUE)
> names(cstock_avg_lu)[2] <- "agb_t_ha_avg"
> cstock_sd_lu <- aggregate(agb_t_ha~land_use_main,data=plotdata3, FUN=sd, na.r
m=TRUE)
> names(cstock_sd_lu)[2] <- "agb_t_ha_sd"
> cstock_count_lu <- aggregate(count~land_use_main, data=plotdata3, FUN=sum, n
a.rm=TRUE)
> cstock_lu <- merge(cstock_count_lu, cstock_avg_lu, by="land_use_main")
> cstock_lu <- merge(cstock_lu,cstock_sd_lu, by="land_use_main")
> str(cstock_lu)
'data.frame':  20 obs. of  4 variables:
 $ land_use_main: chr  "CA0" "CA1" "CA2" "CP0" ...
 $ count        : num  87 66 9 1 1 1 16 78 21 8 ...
 $ agb_t_ha_avg : num  3.19 7.04 20.53 3.55 4.02 ...
 $ agb_t_ha_sd  : num  6.08 7 14.51 NA NA ...
```

land_use_main	count	agb_t_ha_avg	agb_t_ha_sd
CA0	87	3.194	6.079
CA1	66	7.036	6.997
CA2	9	20.53	14.507
CP0	1	3.551	NA
CP1	1	4.021	NA
CP2	1	11.525	NA
FB	16	83.773	86.906
FH	78	27.733	28.186
FM	21	45.242	44.539
PL	8	24.477	28.886
PM	1	0.261	NA
PS	4	7.958	6.356
RL	1	0.076	NA
Sh	1	0.079	NA
SR0	11	3.368	5.869
SR1	178	11.473	10.448
SR2	204	24.66	25.469
SU	6	17.138	7.503
WHB	1	0.445	NA
WP	6	4.715	6.689
WR	1	0.205	NA

TABLE 4. AVERAGE AND STANDARD DEVIATION OF PLOT ABOVEGROUND BIOMASS (IN T/HA) PER LAND USE.

The same information was represented in a graph with the following command lines (). Three objects were needed to represent the averages as dot, the standard deviation as error bars and the number of plots as numbers.

```
> p <- ggplot(cstock_lu)
> p <- p + geom_point(aes(x=land_use_main, y=agb_t_ha_avg), cex=3.5)
> p <- p + geom_text(aes(x=land_use_main, y=agb_t_ha_avg, label=count, hjust=-
1, vjust=1.5, size=9))
> p <- p + geom_errorbar(aes(x=land_use_main, ymax=agb_t_ha_avg+agb_t_ha_sd, ym
in=agb_t_ha_avg-agb_t_ha_sd))
> p <- p + xlab("Land use code") + ylab("aboveground biomass (t/ha)")
> p <- p + theme(legend.position="none")
> p <- p + coord_flip()
> p
```

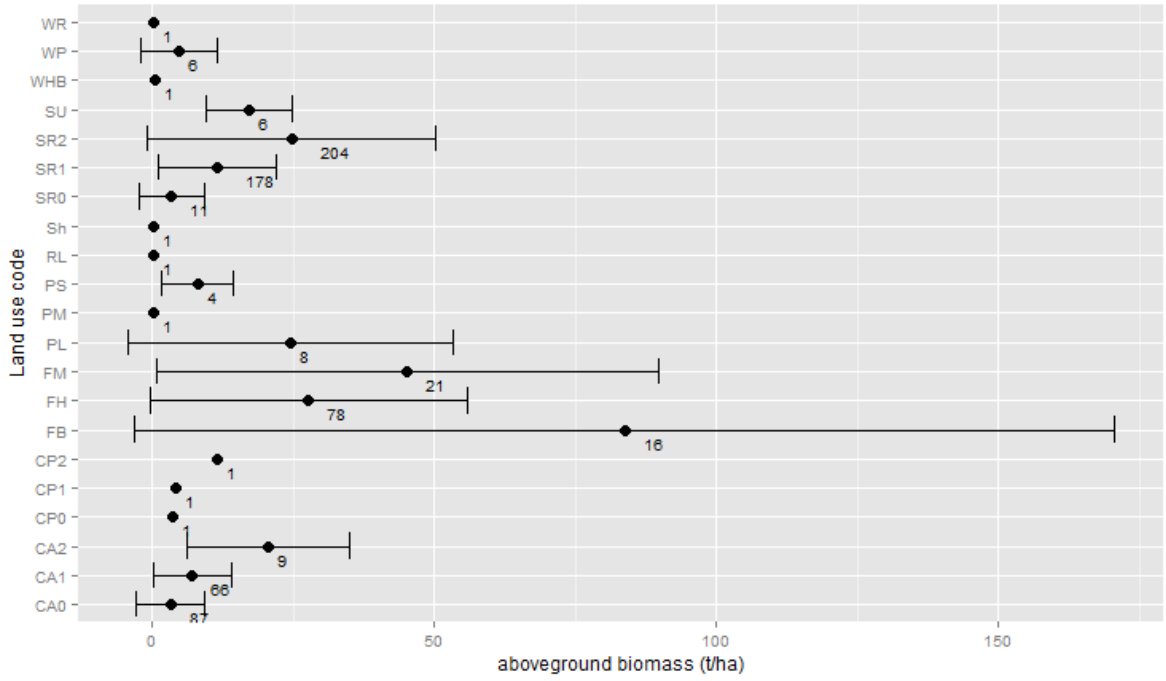


FIGURE 9. AVERAGE ABOVEGROUND BIOMASS PER LAND USE.

BARS REPRESENT THE STANDARD DEVIATION AND NUMBERS INDICATE THE NUMBER OF PLOTS FOR EACH LAND USE.

3.2 Correction of tree species names with the taxonomic name resolution services (TNRS).

In the process of estimating forest biomass, the tree wood density was attributed to each tree based on its scientific name and the availability of generic wood density values for the corresponding species. To ensure the quality of this process, the scientific names identified in the field data collection should be cross check with international databases to ensure the maximum of corresponding wood density values an avoid errors. Cross checking the scientific name to correct typos and synonyms can be a long and tedious exercise but it can be tremendously shorten with existing online tools such as the TNRS.

One website was presented: http://tnrs.iplantcollaborative.org/TNRSapp.html

It allows users to upload a list of species with a simple text file and send back via e-mail a list of proposed corrections based first on a typos correction and then a correspondence to their database, identifying synonyms, unknown species, unresolved names, etc. The use of this service was presented at the end of the workshop.

4 Evaluation of the training and Conclusion

The training was successful and most of the participants were able to apply, modify and develop the code presented during the training workshop. As the training was rather short and intensive, more practice was still needed to develop proficiency with R, but all the participants were enthusiastic to use the R software and very eager to learn all the new functions introduced. The meeting room was good and well equipped. The evaluation was overall very positive (see annex). Many participants still didn't feel comfortable enough to introduce the training contents to colleagues at the end of the week, but it was acceptable and follow up training and practice are planned to develop a community of R users able to develop advanced forest data analysis in Bangladesh.

Appendix 1. Agenda

Dates and venue: January 31st – February 4th 2016 at BBS in Dhaka

Date	Topic	Speaker / facilitator
Day 1	**Opening session**	
8:30	Registration	
9:00	Welcoming remarks Training objectives	UN-REDD Focal point / Gael Sola
9:30	Overview of the first R training	Liam Costello
9:45	Overview of the current data analysis process	RIMS
10:15	Break	
	Session 1: General introduction	
10:45-11:45	General introduction to forest modelling and statistics in the context of REDD	Gael Sola
11:45-12:15	• Verification of software and packages installation • Overview of the data	
	Session 2: hands on R – Objects and simple calculations	
14:00-17:00	• Simple exercises on R • Basic calculations.	
Day 2	**Session 3: hands on R – operations on data frames**	
8:30-12:00	• Creating a file ready for R • Reading the file with R • Adding new columns • Select columns • Aggregate data	
	Session 4: hands on R – graphs and visual interpretation	
14:00-17:00	• Basic graphs • Introduction to ggplot2 • Practice	
Day 3	**Session 5: NFI to forest biomass**	
8:30-12:00	• Correction of species names • Linking wood densities	
	Session 6: NFI to Forest biomass	
14:00-17:00	• Practice	
Day 4	**Session 7: NFI to forest biomass**	
8:30-12:00	• Applying a H-D model using FAO biomes and plot Hmax • Applying a pantropical biomass equation	
	Session 8: NFI to forest biomass	
14:00-17:00	• Aggregating tree biomass to plot and forest types	
Day 5	**Session 8: Conclusion**	
8:30-10:30	• Extraction of results into a report	
10:30	Break	
11:00-12:00	Conclusion	

Appendix 2. Participants list

Name	Organisation	Gender	Designation	Phone no.	e-mail
Md. Safat Ullah	BBS	M	Programmer	01556321393	ullahsafat@yahoo.com
Hossain Mohammad Nishad	FD	M	DFO	01715005677	hmnishad@gmail.com
S.M Zahirul Islam	BFRI	M	Research Officer	01837000010	zahir.fid.bfri@gmail.com
Md Tariq Aziz	FD	M	Research Officer	01790284328	tariqaziz9718@gmail.com
Md Baktiar Nur Siddiqui	FD	M	DFO	01711819670	baktiar1971@gmail.com
Afroza Begum	FD	M	Research Officer	01711283846	b.afroza@yahoo.com
Sourav Das	Shahjalal University	F	Lecturer	01711348630	souravdron@gmail.com
Md Raqibul Hasan Siddique	Khulna University	M	Assistant Professor	01716422182	raqibulhasan_fwt@yahoo.com
Akhter Hossain	Chittagong University	M	Assistant Professor	01827501435	akhter.hossain@cu.ac.bd
Mariam Akhter	FAO	F	FAO	01711170697	Mariam.akhter@fao.org
Liam Costello	FAO	M	FAO		Liam.costello@fao.org

Appendix 3. Evaluation results

Evaluation question	Average note from 11 participants (*)
The training was relevant to my daily work	1.2
I had enough previous knowledge to understand the contents of the event	1.4
The event provided me with new and useful knowledge/information	1.0
The training met my expectations in terms of the content and learning outcomes	1.1
I am interested to introduce this content to other people	1.0
I feel confident to be able to carry out the tasks described in the training without supervision.	1.6
Wordings used in the materials is understandable	1.1
Materials are relevant to Event content	1.0
Event materials were useful to my work	1.2
The resource person(s)' presentation indicated that they had made proper preparation for the Event	1.0
The resource person(s) was/were knowledgeable about the subject	1.0
The resource person(s) made clear and satisfactory presentation(s) on their topics	1.0
I was pleased with the venue/meeting room	1.3
I was happy with other support services (equipment, coffee break, lunch etc.)	1.5

(*) 1:yes 2:average / no opinion 3:novaluation