eNFI Lesson 1: Overview

## Welcome!

Welcome to the first lesson of the **National Forest Inventory eLearning practice module**.

This interactive module is designed to complement the [NFI eLearning modules](https://bit.ly/3m5VEsE) with practical hands-on exercises. These exercises take place on a fictional island that just emerged in the middle of the Atlantic Ocean: Louland.

It features interactive lessons designed to practice National Forest Inventory (NFI) sampling and the exploratory analysis of NFI field data with the [R programming language](www.r-project.com). Although this fictional land is not an official country, we will use the acronym NFI when referring to its forest inventory as the method and formulas applied here are valid for nationwide forest inventories.

### Objectives

In this first lesson, **Overview of the preliminary data**, we will discover the land cover of Louland, and we will recalculate the average Aboveground Biomass (AGB) of a small inventory, carried out by the first exploration crew in one forest stand.

The objectives of this lesson are:

* familiarize yourself with how the interactive lessons of the NFI technical module 9 work,
* familiarize yourself with Louland, our fictional land to practice national Forest Inventory sampling design and data analysis,
* Calculate an average AGB estimate for one forest stand,
* Calculate the areas of different land covers, based on shapefile spatial data.

In this lesson, we will use the collection of packages grouped inside the package tidyverse for general data analysis and the package sf for spatial data analysis. We will also briefly use the package units to avoid confusion around area units.

library(tidyverse)  
library(sf)  
library(units)

Note that in the background we also use showtext to add Google Fonts to our figures, ggspatial to add the North arrow and scale to our maps and tmap for the interactive map at the end of the lesson.

Let’s get started!

### Interactions with R

Whenever you see “**Your turn!**”, followed by a small console window, these are actual R consoles. You can run R code in these consoles, and you will be presented with either pre-filled code sequences, with holes for you to complete, or an empty console.

Follow the instructions and try to complete or create the code necessary to get the table, value or figure requested. The lesson continue after the exercise is completed and the correct code is submitted.

**Instructions**

The interactive lessons are designed on a trial and error model. The code is first showcased then pre-filled code with holes is presented for you to complete in order to become familiar with the commands. Finally, after a few practice exercises, the last console for a group of commands will be empty.

Let’s test it out. Type 1+1 in the console below, Click **Run code** to see the results, and **Submit answer** to check if your code is correct.

## [1] 2

## [1] ""

## A new land has emerged

An island previously unknown to humankind has just been discovered and you are part of a team of scientists commissioned to inventory its forest resources, mainly its carbon stock.

In a future version of this interactive training module, you will be able to choose different land profiles with different (1) forest categories, (2) ratio of ocean covering the 90 x 90 km square frame in which the new land is created, (3) inclusion or exclusion of mangrove forests and even (4) the maximum altitude of Louland (it will have an impact on inaccessible plots)!

For now, Louland forests are divided into 5 categories: Evergreen forest, Mixed-Deciduous forest, Deciduous forest, Other Woodland and Mangrove Forest, also all the island is considered accessible.

The remote sensing team has developed land cover and topographic maps based on 30m resolution remote sensing images and initial forest plot measurements taken by the first exploration crew. Based on their observations, we will be able to have a general understanding of Louland’s forest cover. This data is called **auxiliary data**, it is not NFI data *per se*, but useful information to help design and implement an NFI.

Here is what Louland’s land covers look like with a map generated with the packages ggplot2, sf and ggspatial.

In the next section, we will have a quick look at the exploration crew’s forest inventory data, loaded in R as objects with the exfi prefix, and we will use the land cover maps to calculate the area of the main land cover types.

## First forest inventory

### Overview

First, let’s have a look at the results of 10 forest plots measured by the exploration crew. The data is stored in the object exfi\_agb. By printing the object name in the console, we will see the results:

print(exfi\_agb)

## # A tibble: 1 x 8  
## n\_plot n\_tree mean\_ba sd\_ba mean\_agb sd\_agb ci ci\_perc  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 10 303 15.0 9.33 130. 111. 69.0 53

Since we are using functions from the tidyverse collection of R packages, data tables in these interactive lessons are tibbles. They are simple rectangular tables (each column has the same number of rows), and when running their names in the console, we are given the number of rows and columns, the column names with the type of data and the first 10 values for each column.

The table (or tibble) exfi\_agb contains the number of plots n\_plot, the mean and standard deviation for aboveground biomass and basal area: mean\_agb and sd\_agb in ton/ha, mean\_ba and sd\_ba in m2/ha, and ci and ci\_perc represent the confidence interval around the mean AGB in ton/ha and percentage of the mean respectively.

The 10 plots come from a random sample from one of the Evergreen forests of Louland. The crew also shared the plot level aboveground biomass in the table name exfi\_pagb.

**Your turn!** Run the table name, exfi\_pagb, in the console below to see the table’s basic information.

## Error: <text>:1:7: unexpected input  
## 1: print(\_  
## ^

## # A tibble: 10 x 4  
## plot\_id count\_tree plot\_ba plot\_agb  
## <chr> <dbl> <dbl> <dbl>  
## 1 ev-ao6q 40 1.92 13.8  
## 2 ev-ax7d 136 9.45 70.3  
## 3 ev-hjq2 397 21.0 217.   
## 4 ev-jf9j 713 18.3 114.   
## 5 ev-jwe6 729 13.9 85.4  
## 6 ev-mre2 152 33.7 397.   
## 7 ev-pzf6 96 10.1 105.   
## 8 ev-ulq4 269 11.6 61.7  
## 9 ev-vwh1 402 23.8 179.   
## 10 ev-xu1z 96 6.42 53.6

## [1] ""

The table, exfi\_pagb, contains the ID plot\_id, number of trees per hectare count\_tree, basal area plot\_ba in m2/ha and aboveground biomass plot\_agb in ton/ha, for each plot.

### Recalculate the mean AGB

Given that the crew chose the plot location randomly, we can use simple aggregating functions to re-calculate the forest mean aboveground biomass and confirm their calculations. We will use a mix of base R and tidyverse functions, in particular the function summarise(). This function is very handy to aggregate numerical variables in different columns of our plot table and can even be used in combination with the group\_by() function to summarize numerical variables based on category variables of the same table (more on that later, with the full NFI data).

Let’s see, for example, how to calculate the mean basal area of the forest with its standard deviation and save the results in a table exfi\_ba. After creating the table, we run the table name in the console to display its content.

exfi\_ba <- exfi\_pagb %>%  
 summarise(  
 mean\_ba = mean(plot\_ba),  
 sd\_ba = sd(plot\_ba)  
 )

Which gives:

print(exfi\_ba)

## # A tibble: 1 x 2  
## mean\_ba sd\_ba  
## <dbl> <dbl>  
## 1 15.0 9.33

**Your turn!** Create a table, exfi\_agb2, with the columns mean\_agb and sd\_agb to create respectively the mean and standard deviation of plot level AGB, then display the results.

## Error: <text>:3:21: unexpected input  
## 2: summarise(  
## 3: mean\_agb = mean(\_  
## ^

## [1] ""

### Results

We have successfully found the same values for AGB: 129.65 ton/ha, on average, with a standard deviation of 111.29. You can confirm by visualizing your table. Printing its name in the console below:

## # A tibble: 1 x 2  
## mean\_agb sd\_agb  
## <dbl> <dbl>  
## 1 130. 111.

In the next section we will have an overview of the spatial data on Louland’s land cover classes.

## land cover types and areas

### Spatial objects

We can use the land cover shapefile provided to us, by the remote sensing analysts, to view the extent of the different land covers and calculate the area of the forest cover. The shapefile is loaded in the R environment under the name sf\_lc with the package sf. It contains the attribute table, the geometries and the coordinate reference system of Louland’s land cover. Running the object by its name in the console displays its summary information.

print(sf\_lc)

## Simple feature collection with 543 features and 4 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: 560363 ymin: 9919901 xmax: 650363 ymax: 10009900  
## Projected CRS: WGS 84 / UTM zone 27S  
## First 10 features:  
## lc lc\_id lc\_name id geometry  
## 1 EV 5 Evergreen 1 POLYGON ((566193.1 10009779...  
## 2 MD 4 Mixed-deciduous 2 POLYGON ((566803.7 10009814...  
## 3 DD 3 Deciduous 3 POLYGON ((570572.2 10009847...  
## 4 DD 3 Deciduous 4 POLYGON ((572188.9 10009853...  
## 5 WL 2 Woodland 5 POLYGON ((572457.7 10009848...  
## 6 NF 1 Non-forest 6 POLYGON ((619163.6 10009844...  
## 7 DD 3 Deciduous 7 POLYGON ((561303.3 10009842...  
## 8 NF 1 Non-forest 8 POLYGON ((562396.2 10009759...  
## 9 EV 5 Evergreen 9 POLYGON ((616198.6 10009770...  
## 10 EV 5 Evergreen 10 POLYGON ((567339.4 10009810...

This summary view regroups the Coordinate Reference System (CRS), the type of geometry (polygon), the number of polygons (features) and fields, and the detailed information on the first 10 polygons in the shapefile. The four fields represent the polygon IDs (id), the land cover IDs (lc\_id), codes (lc) and names (lc\_name).

To isolate the list of different land covers we first need to convert the spatial data into a tibble with the function as\_tibble(), then select only the columns related to land cover information with the function select(). We can use the distinct() function to keep only unique values across the newly formed data table. Finally, since the land cover IDs are numerical, we can sort the resulting table by increasing value of lc\_id with the function arrange().

lc\_tab <- sf\_lc %>%  
 as\_tibble() %>%  
 select(lc\_id, lc, lc\_name) %>%  
 distinct() %>%  
 arrange(lc\_id)  
  
print(lc\_tab)

## # A tibble: 7 x 3  
## lc\_id lc lc\_name   
## <dbl> <fct> <fct>   
## 1 0 WA Water   
## 2 1 NF Non-forest   
## 3 2 WL Woodland   
## 4 3 DD Deciduous   
## 5 4 MD Mixed-deciduous  
## 6 5 EV Evergreen   
## 7 6 MG Mangrove

With the tidyverse packages, we have the pipe operator, %>%. This operator allows us to apply functions one after the other, in a sequence and without needing to save each intermediate step in a new object.

Step-by-step the above code could be written as:

t1 <- as\_tibble(sf\_lc)  
t2 <- select(.data = t1, lc\_id, lc, lc\_name)  
t3 <- distinct(.data = t2)  
lc\_tab <- arrange(.data = t3, lc\_id)

Notice that with the pipe operator, %>%, we can place the commands one after the other and the resulting table of each function is silently passed to the next function. We don’t need to repeat the input .data =.

**Let’s set a few naming conventions.**

All forest inventory-related tables are stored in R objects with names related to their contents. For example the NFI plot and tree data are stored in tables plot and tree, respectively.

Using plot as a table name is not very good practice in R, as plot() is the base R function for plotting data. However, we still keep it because it is an informative name. The NFI tables in this tutorial are: plot, tree and species\_list.

The prefix exfi\_, from the tables shown in the previous section was added for the data from the first exploration crew’s forest inventory.

Similarly, all spatial data are handled mostly with the sf package, and the prefix, sf\_, was used as a naming convention. The spatial objects available for this tutorial are as follows: sf\_admin, sf\_lc and sf\_topo, respectively the administrative boundaries, land cover, and topography of Louland.

### Land cover areas

sf objects behave similarly to standard tibbles, making it easy to apply tidyverse functions, while keeping the spatial information, or to convert them to tibbles with the function as\_tibble(). The sf package also contains the st\_area() function that we are going to use to calculate the areas of the land covers.

In the following sequence, we combine the function mutate(), used to create or modify columns in our target table, with the function st\_area(). Since st\_area() takes the whole spatial object as input, we could use st\_area(sf\_lc) or st\_area(.), which is the standard way to call the current state of a tibble inside a sequence of functions linked with pipes.

area\_lc <- sf\_lc %>%  
 mutate(  
 area\_m2 = st\_area(.),  
 area\_ha = units::set\_units(area\_m2, value = ha)  
 ) %>%  
 as\_tibble() %>%   
 group\_by(lc) %>%  
 summarise(area\_ha = sum(area\_ha))

Which gives:

print(area\_lc)

## # A tibble: 7 x 2  
## lc area\_ha  
## <fct> [ha]  
## 1 WA 3883.  
## 2 NF 156830.  
## 3 WL 152703.  
## 4 DD 63978.  
## 5 MD 186385.  
## 6 EV 88912.  
## 7 MG 10362.

**Your turn!** Calculate the total area of land in Louland, all land cover classes together, and save it in an object area\_tot.

Hint: group\_by() is not necessary anymore.

## Error: <text>:3:23: unexpected input  
## 2: mutate(  
## 3: area\_m2 = st\_area(\_  
## ^

## [1] ""

### Results

Check the total area:

## # A tibble: 1 x 1  
## area\_ha  
## [ha]  
## 1 663054.

We have found that the total area of Louland is 663054 ha. From the table area\_lc, we know that 3883 and 156830 ha are Water and Non-forest respectively. This leaves 502340 ha of Forest land.

## Conclusion

This is the end of the lesson 1. In the next lessons, we will use these preliminary data to design different sampling frames and analyze tree and plot data. Forest inventory data will be assigned to our sampling design and we will use it to estimate several NFI main variables, such as carbon, biodiversity or production indicators.

Before moving on with the sampling designs and other lessons in this practice module, here is a dynamic view of Louland and the first 10 forest inventory plots, using the package tmap.

