# 01_Filter_data_Markdown

Daniela Linero

23/4/2020

## Filtering the PREDICTS data

In this document I am going to present the process to filter the PREDICTS database and get the records belonging to:

- Tropical bird and mammal frugivores
- Tropical plants with endozoochorous seed dispersal

### 1. Getting tropical data

1.1. Import and explore structure of PREDICTS data:

```
diversity <- readRDS("./data/PREDICTS2020/PREDICTS2020.rds")
```

```
# Total number of records
length(diversity$Source_ID)
```

```
## [1] 4054275
```

```
# Total number of species
length(unique(diversity$Best_guess_binomial))
```

```
## [1] 32149
```

1.2. Filter tropics

I am going to filter all the coordinates that are between the Tropic of Cancer and Tropic of Capricorn. Additionally, I am going to select the records that are located in forests biomes (Tropical & Subtropical Dry Broadleaf Forests and Tropical & Subtropical Moist Broadleaf Forests). I excluded Tropical & Subtropical Coniferous Forests because Gymnosperms don't produce fruits.

```
TropicsDiversity <- diversity %>%

  # Select the coordinates that fall within the tropics
  filter(between(Latitude, -23.43, 23.43)) %>%

  # Select records located inside forest ecosystems
  subset(Biome == "Tropical & Subtropical Moist Broadleaf Forests" | Biome == "Tropical & Subtropical D
  # drop to remove levels with zero records
  droplevels()

# Number of records per region
table(TropicsDiversity$Region)
```

```
##
##          Africa            Asia       Australia       Caribbean
##          258495          663822             192             918
## Central America   North America         Oceania         Pacific
##           92815           80405            2316           18673
##   South America
##          169251
```

```
#Number of records per biome
table(TropicsDiversity$Biome)
```

```
##
##   Tropical & Subtropical Dry Broadleaf Forests
##                                          40404
## Tropical & Subtropical Moist Broadleaf Forests
##                                        1246483
```

```
# Total number of records
length(TropicsDiversity$Source_ID)
```

```
## [1] 1286887
```

```
# Total number of species
length(unique(TropicsDiversity$Best_guess_binomial))
```

```
## [1] 14346
```

### 2. Selecting bird and mammal frugivores in the PREDICTS dataset

I use Terborgh's (1986) definition of frugivore: animal whose diet is made up of at least 50% fleshy fruits.

2.1. Identify species of frugivorous mammals:

I use the *Eltontraits* dataset for mammals to select frugivore species. The dataset uses diet information mostly taken from Walker's Mammals of the World. For species that did not have information on their diet in this source - or in other books or primary literature - the diet was infered based on congeneric species.

```
Mammal_Diet <- read.csv("./Data/Eltontraits/MamFuncDat.csv", sep = ";")
```

```
Frugivore_mammals <- Mammal_Diet %>%

  # Filter mammals that have a diet composed of at least 50% fruit
  filter(Diet.Fruit >= 50) %>%

  # Select the columns of scientific name, % of diet composed of fruits, and diet certainty level
  select(Scientific, Diet.Fruit, Diet.Certainty)

# Number of frugivore mammal species
length(unique(Frugivore_mammals$Scientific))
```

```
## [1] 605
```

2.2. Identify species of frugivorous birds:

Again, I use the *Eltontraits* dataset but for birds.

```
Birds_Diet <- read.csv("./Data/Eltontraits/BirdFuncDat.csv", sep = ";")
```

With the following code, I filter the frugivorous birds and merge the resulting species with the mammal species identified in the previous step

```
Frugivores <- Birds_Diet %>%

  # Filter birds that have a diet composed of at least 50% fruits
  filter(Diet.Fruit >= 50) %>%

   # Select the columns of scientific name, % of diet composed of fruits, and diet certainty level
  select(Scientific, Diet.Fruit, Diet.Certainty)  %>%

  # Bind bird and mammal data
  rbind(Frugivore_mammals) %>%

  # Rename the scientific name column to match the PREDICTS dataset
  rename(Best_guess_binomial = Scientific)

# Total number of frugivorous birds and mammals

length(unique(Frugivores$Best_guess_binomial))
```

```
## [1] 1953
```

The level of uncertainty in the Eltontraits dataset provides information on the source from which the species diet was taken and whether it was inferred from other species in the same genus or from a different genus.

ABC: The source is reliable (A: high, B: reasonably, C: quality unclear or estimated from a congeneric species) D1: Value is typical for the genus D2: Value is typical for the family

```
# Number of records per level of uncertainty

table(Frugivores$Diet.Certainty)
```

```
## 
##         A    B    C   D1   D2  ABC 
##    0 1020  160   56  205   26  486
```

### 2.3. Check the species names

The PREDICTS database presents binomial species names using the Catalog of Life nomenclature (Hudson et al., 2014). For this reason, I verified that the scientific names used in the EltonTraits dataset matched the corresponding species name presented in the Catalog of Life.

```
# Get a character vector with the species names
Frugivore_Names <- as.character(Frugivores$Best_guess_binomial)

# Get the submitted species names and the corresponding name used in the Catalogue of Life (id = 1)
Frugivore_Names<- gnr_resolve(names = Frugivore_Names, data_source_ids = 1) %>%

  # Identify records that have a different name in the Catalogue of Life (if the name is the same
  # the score will be 0.988)
  subset(score < 0.98)

# Number of species that have different nomenclatures in both tables

length(unique(Frugivore_Names$matched_name))
```

```
## [1] 0
```

Since all of the species names matched the ones in the Catalogue of Life, I will merge the Frugivore species data with the PREDICTS data.

### 2.4. Get the records that belong to frugivore birds and mammals in the PREDICTS dataset

```
# Merge both datasets and retain only the species of PREDICTS that match the species in the Frugivores
PREDICTS_frugivores <- merge(x=TropicsDiversity, y=Frugivores, by = "Best_guess_binomial", all.x = FALS
  droplevels()

# Total number of records
length(PREDICTS_frugivores$Source_ID)
```

```
## [1] 29057
```

```
# Total number of species
length(unique(PREDICTS_frugivores$Best_guess_binomial))
```

```
## [1] 456
```

```
#  Number of records belonging to Birds and mammals
table(PREDICTS_frugivores$Class)
```

```
##
##     Aves Mammalia
##    26247     2810
```

```
# Total number of sourceIDs
length(unique(PREDICTS_frugivores$Source_ID))
```

```
## [1] 64
```

```r
# Total number of studies
length(unique(PREDICTS_frugivores$SS))
```

```
## [1] 74
```

```r
# Total number of sites
length(unique(PREDICTS_frugivores$SSBS))
```

```
## [1] 2472
```

The description of the habitat type is classified at the site level, however, each site has multiple records representing different species sampled. Therefore, it is necessary to get a unique record for each site to properly count how many sites there are for the predominant type and use intensity categories.

```r
# Function that returns the first record for each SSBS (SSBS = Concatenation of Source_ID, Study_number

Number_sites_landuses <- function(dataframe){
# Get a vector with the SSBS names
Sites <- dataframe %>% pull(SSBS) %>% as.character()
# Find the index of the first match for each SSBS and eliminate the duplicates
Indeces <- match(Sites, dataframe$SSBS) %>% unique()
# Get a data frame with only the first matches of each SSBS
justFirsts <- dataframe[Indeces, ]
return(justFirsts)
}
```

```r
# Get the first record for each SSBS (SSBS = Concatenation of Source_ID, Study_number, Block and Site_n
justFirstsFrugivores <- Number_sites_landuses(PREDICTS_frugivores)

# Number of sites per land-use type and land-use intensity
addmargins(table(justFirstsFrugivores$Predominant_habitat,justFirstsFrugivores$Use_intensity), 2)
```

```
## 
##                                             Minimal use Light use
##   Primary forest                                    468       267
##   Primary non-forest                                  6         4
##   Young secondary vegetation                         65        11
##   Intermediate secondary vegetation                  99        70
##   Mature secondary vegetation                        18       137
##   Secondary vegetation (indeterminate age)           43        21
##   Plantation forest                                  34       360
##   Pasture                                             0         8
##   Cropland                                           78        21
##   Urban                                               0        40
##   Cannot decide                                       0         0
## 
##                                             Intense use Cannot decide Sum
##   Primary forest                                     59            47 841
##   Primary non-forest                                  9             0  19
##   Young secondary vegetation                          0            82 158
##   Intermediate secondary vegetation                   2            39 210
##   Mature secondary vegetation                       0           24 179
##   Secondary vegetation (indeterminate age)          0           27  91
##   Plantation forest                                19           14 427
##   Pasture                                           9          172 189
##   Cropland                                         66           95 260
##   Urban                                             1            0  41
##   Cannot decide                                     0           57  57
```

### 3. Selecting plants dispersed by endozoochory in the PREDICTS dataset

3.1. Check species nomenclature

I am going to use the TRY dataset which has information on the traits of Dispersal syndrome and Fruit type. Since the nomenclature varies a lot with respect to the PREDICTS dataset, the first thing that I'm going to do is to check the nomeclature and get all of the species names in binomial format (i.e. raise all of the subspecies and varieties to species level).

First I will identify the number of words in each species names to know if they are classified as genre, species, subspecies or variety level.

```r
Plantspecies <- TRYdata %>%

  # get only the records that have information for the trait
  subset(TraitID != "NA") %>%

  # select only the columns with important information
  select(Dataset, SpeciesName, AccSpeciesName, TraitID, TraitName,
         DataName, OriglName, OrigValueStr, Reference, Comment) %>%

  # drop levels
  droplevels() %>%

  # create a column to store the information of level of classification
  mutate(Nwords = NA)


# Count the words of the species names to determine the level of classification
for (i in 1:nrow(Plantspecies)){

  # print the index to see the progress of the loop
  print(i)

  # count the words in the consolidated species name
  Plantspecies$Nwords[i] <- wordcount(Plantspecies$AccSpeciesName[i])
}

# Select the species that are identified at the species or subspecies level

Plantspecies <- Plantspecies  %>%

  # Filter the rows that are classified at species, subspecies or variety level
  filter(Nwords == 2 | Nwords == 4) %>%

  #drop levels
  droplevels()
```

Table 1: Fragment of the resulting table

| | AccSpeciesName | OrigValueStr | Nwords |
|---|---|---|---|
| 1 | Carnegiea gigantea | Animal | 2 |
| 141 | Ericameria nauseosa var. nauseosa | Wind | 4 |
| 730 | Papaver dubium subsp. laevigatum | Wind | 4 |

After filtering only the names that are at the species, variety and subspecies level, I will check the nomenclature comparing it with the Catalogue of Life.

```
# Make a vector with the species names
PlantNames <- Plantspecies %>%

  # Get unique names
  distinct(AccSpeciesName) %>%

  # Make a character vector with the names of the species
  pull()


# Divide the vector because the gnr_resolve function does not work with large character vectors
a <- PlantNames[1:5500]
b <- PlantNames[5501:11000]
c <- PlantNames[11001:15983]

# Get the data source id for Catalogue of life
sources <- gnr_datasources()

# Get the matched name in the Catalogue of life
aUpdated <- gnr_resolve(names= a, data_source_ids=1)
bUpdated <- gnr_resolve(names= b, data_source_ids=1)
cUpdated <- gnr_resolve(names= c, data_source_ids=1)

# Export results
Updated_plant_names <- rbind(aUpdated, bUpdated, cUpdated)
```

Table 2: Fragment of the resulting table showing names that are already updated (score > 0.988), and a species that has a different nomeclature in the Catalague of life (score = 0.75) including a synonym

| | user_supplied_name | matched_name | data_source_title | score |
|---|---|---|---|---|
| 1 | Carnegiea gigantea | Carnegiea gigantea (Engelm.) Britton & Rose | Catalogue of Life | 0.988 |
| 401 | Barringtonia acutangula subsp. spicata | Barringtonia acutangula subsp. spicata | Catalogue of Life | 0.999 |
| 5076 | Hypochoeris glabra | Hypochaeris glabra L. | Catalogue of Life | 0.750 |
| 5077 | Hypochoeris glabra | Hypochaeris glabrata Phil. | Catalogue of Life | 0.750 |

```
# Raise subspecies and varieties to species level
Updated_plant_names <- Updated_plant_names %>%
  # Get the updated names and raise the subspecies and varieties to species by selecting the
  # first two words of the string (this will include synonyms)

  mutate(Updatedname = word(matched_name, 1, 2)) %>%

  # Select the original name and the updated name in the Catalogue of life
  select(user_supplied_name, Updatedname) %>%

  # Rename the column to match the original TRY dataset
  rename("AccSpeciesName" = user_supplied_name)
```

| | AccSpeciesName | Updatedname |
|---|---|---|
| 1 | Carnegiea gigantea | Carnegiea gigantea |
| 401 | Barringtonia acutangula subsp. spicata | Barringtonia acutangula |
| 5076 | Hypochoeris glabra | Hypochaeris glabra |
| 5077 | Hypochoeris glabra | Hypochaeris glabrata |

In order to find out what synonyms are present in the PREDICTS dataset, I am going to get the species names that are shared between the PREDICTS dataset and the Updated names of the TRY dataset

```
Updated_plant_names <- Updated_plant_names %>%

  # Subset the species that are present in both datasets
  subset(Updatedname %in% TropicsDiversity$Best_guess_binomial) %>%

  # drop levels
  droplevels()

  # How many species are present in both datasets
  length(unique(Updated_plant_names$Updatedname))
```

```
## [1] 1151
```

I am going to create a table with the TRY records that belong to species, subspecies and varieties that are present (as species) in the PREDICTS database

```
Plantspecies <- merge(x=Plantspecies, y=Updated_plant_names, by = "AccSpeciesName", all.x = FALSE)
```

Table 4: Fragment of the resulting table

| | AccSpeciesName | TraitID | TraitName | OrigValueStr | Updatedname |
|---|---|---|---|---|---|
| 1 | Abarema jupunba | 28 | Dispersal syndrome | animal | Abarema jupunba |
| 154 | Acridocarpus longifolius | 99 | Fruit type | nut | Acridocarpus longifolius |

### 3.2. Identify plants with endozoochorous seed dispersal

First, I will divide the dataset into one table with the trait of Dispersal syndrome and another one with the trait of Fruit type.

```r
# Get the table for dispersal syndrome
Dispersal_syndrome <- Plantspecies %>%
  subset(TraitID == 28) %>%
  droplevels()

# Number of species in the dataset
length(unique(Dispersal_syndrome$Updatedname))
```

```
## [1] 738
```

```r
# Get the table for fruit type
Fruit_type <- Plantspecies %>%
  subset(TraitID == 99) %>%
  droplevels()

# Number of species in the dataset
length(unique(Fruit_type$Updatedname))
```

```
## [1] 548
```

I will start selecting plants with endozoochorous seed dispersal from the Dispersal syndrome table. Therefore, I am going to eliminate all of the records that explicitly state a different seed dispersal syndrome. Additionally, I will also eliminate the records where no mammal or bird were included as consumers

```r
# See the factors of dispersal syndrome type
DS_levels <- as.data.frame(levels(Dispersal_syndrome$OrigValueStr))

# Example of the mentioned values
DS_levels$`levels(Dispersal_syndrome$OrigValueStr)`[c(4, 8, 14, 19)]
```

```
## [1] a-wind
## [2] agouti, horse, cattle, deer, tapir, peccary, paca, some rodents
## [3] anemochory
## [4] animal
## 233 Levels: 0 ... zoochor
```

This is the list of values I will not include in my analysis

```r
Not_endozoo <- c("a-wind", "Anemo", "anemochory", "Anemochory", "ant", "ant-dispersed",
                 "Ants", "autochor", "Autochory", "ballochor", "boleochor", "Cichlids", "Pacu",
                 "wind+water", "commerce", "wind", "dysochor", "epizoochor", "ethelochor",
                 "exozoochory", "explosive", "Explosive", "Fish", "Floating", "generative",
                 "germinule", "machinery", "hemerochor", "man", "meteorochor",
                 "nautochor", "ombrochor", "seed contamination", "ground", "    speirochor", "snail",
                 "turtles", "Wind", "water", "Water", "agochor", "Anchorage", "terrapin",
                 "Pheidole", "car", "carried", "clothes", "environmental", "erosion", "Geochelone",
                 "rainwash", "reptiles", "roe", "speirochor", "unassisted", "Unassisted",
                 "unspecialised")


# Filter out all of the records with values in the vector Not_endozoo
```

```r
Dispersal_syndrome <- Dispersal_syndrome %>%

  filter(!str_detect(OrigValueStr, paste(Not_endozoo, collapse = "|"))) %>%

  # drop levels
  droplevels()


# Number of species in the dataset
length(unique(Dispersal_syndrome$Updatedname))
```

```
## [1] 719
```

Now I am going to create a vector with the species names that are explicitly classified with endozoochory.**However**, since each dataset that makes up the TRY dataset uses different terms, I will filter each one separately.

```r
# The Miombo dataset uses a 1 to specify species that have endozoochory
Endoo_PlantSpecies <- Dispersal_syndrome %>%

  # Select specific dataset
  subset(Dataset == "Miombo tree species - SLA, leaf and seed size") %>%

  # Filter values with condition TRUE for endozoochory
  filter(OriglName == "dispersal mode: endozoochory" & OrigValueStr == 1) %>%

  # Make a character vector
  droplevels() %>% pull(Updatedname) %>% as.character()
```

I created a function to make things faster:

```r
# Function that updates the vector of species names that have endozoochorus dispersal syndrome by
# taking a specific dataset and searching for the TRUE condition for endozoochory

Filter_endozoochory_terms <- function(DatasetName, FilterValue ) {
  c(Endoo_PlantSpecies, as.character(Dispersal_syndrome %>% # Update vector

                                       # Select specific dataset
                                       subset(Dataset == DatasetName) %>%

                                       # Filter out all of the species that have already been identifie
                                       # as having endozoochory
                                       subset(Updatedname %nin% Endoo_PlantSpecies) %>%

                                       # Filter values with condition TRUE for endozoochory
                                       filter(str_detect(OrigValueStr, FilterValue)) %>%

                                       # Get unique species names
                                       distinct(Updatedname) %>%

                                       # Make a vector
                                       droplevels() %>% pull(Updatedname)))
}
```

Filter all datasets

```
# Apply function Filter_endozoochory_terms with each of the databases

Endoo_PlantSpecies <- Filter_endozoochory_terms(DatasetName = "The LEDA Traitbase", FilterValue = "endoz
Endoo_PlantSpecies <- Filter_endozoochory_terms(DatasetName = "BASECO: a floristic and ecological databa
                              FilterValue = "Endo-zoochory")
Endoo_PlantSpecies <- Filter_endozoochory_terms(DatasetName = "Global Seed Mass, Plant Height Database"
Endoo_PlantSpecies <- Filter_endozoochory_terms(DatasetName = "KEW Seed Information Database (SID)", Fil

# Number of endozoochorus species until now
length(unique(Endoo_PlantSpecies))
```

```
## [1] 326
```

Since a lot of species have an animal as seed disperser but don't have information whether it is endozoochory or epizoochory, I will use the dataset of fruit type to clarify the dispersal syndrome.

```
# Get the list of species that need revision
Revise_species <-   Dispersal_syndrome %>%

  # Subset all the species that are not present in the Endozoochory list
  subset(Updatedname %nin% Endoo_PlantSpecies) %>%

  # Get unique species names
  distinct(Updatedname, .keep_all = TRUE) %>%

  # drop levels
  droplevels()

# Number of species that need revision
length(unique(Revise_species$Updatedname))
```

```
## [1] 393
```

```
# Identify the species in the list of revision that have fleshy fruits and therefore may be dispersed b


# Create a character vector with the factors of fleshy fruits
Fleshy <- c("berry", "Berry", "drupe", "Drupe")

# Identify the species that have fleshy fruits
Fruit_type <- Fruit_type %>%

  # Filter all of the records with fleshy fruits
  filter(str_detect(OrigValueStr, paste(Fleshy, collapse = "|"))) %>%

  # Select only the name of the species and the fruit type
  select(Updatedname, OrigValueStr) %>%

  # Get the unique species names
  distinct(Updatedname, .keep_all = TRUE) %>%
  # drop levels
  droplevels()

  # Number of species with fleshy fruits
  table(Fruit_type$OrigValueStr)
```

```
##
## berry Berry drupe Drupe
##   119     5   106     9
```

Now I will add to the list of species with endozoochory, those species that:

- Needed revision (are dispersed by animals) and have fleshy fruits
- Species with fleshy fruits but without a seed dispersal syndrome value. I will do this because fruit morphology usually indicates the means of dispersal (Howe & Smallwood, 1982), and most studies equate fleshy fruits with endozoochory (Chen, Cornwell, Zhang, & Moles, 2017).

```
Endoo_PlantSpecies <- c(Endoo_PlantSpecies, as.character(merge(x = Revise_species, y = Fruit_type, by=
                                                           # datasets of plants dispersed by animals, a

                                                           # Filter out all of the species that don't h
                                                           # have fleshy fruits
                                                           subset(OrigValueStr.y != "NA") %>%

                                                           # drop levels
                                                           droplevels() %>%

                                                           # Get unique species names
                                                           distinct(Updatedname) %>%

                                                           # Make a vector
                                                           droplevels() %>% pull(Updatedname)))

# Eliminate repeated records between the Fruit type and Dispersal syndrome datasets
Endoo_PlantSpecies <- unique(Endoo_PlantSpecies)

# Updated number of species with endozoochorous seed dispersal
length(unique(Endoo_PlantSpecies))
```

```
## [1] 534
```

The rest of the species were revised manually using the Royal Botanic Gardens Kew Seed Information Database (SID), 2020

```
# Update list to revise
Revise_species <-   Revise_species %>%

  # Subset all the species that are not present in the Endozoochory list
  subset(Updatedname %nin% Endoo_PlantSpecies) %>%

  # drop levels
  droplevels()

# Number of species to revise
length(unique(Revise_species$Updatedname))
```

```
## [1] 372
```

```
# Load species checked manually
Revised <- read.csv("./output/intermediate_files/01_Filter_data_Plant_species_revised.csv", sep = ";")
```

Get the final list of endozoochorous plant species

```
Endoo_PlantSpecies <- c(Endoo_PlantSpecies, as.character(Revised  %>%
                                                              pull(Updatedname)))

# Number of endozoochorous plant species
length(unique(Endoo_PlantSpecies))
```

```
## [1] 632
```

3.3. Identify plants with endozoochorous seed dispersal in the PREDICTS database

```
Endoo_PlantSpecies <- read.csv("./output/intermediate_files/01_Filter_data_List_endozoochory_species.csv

  # rename column to match the PREDICTS dataset
  rename(Best_guess_binomial = x) %>%

  # get unique species names
  distinct(Best_guess_binomial)


PREDICTSendooPlants <- merge(x=TropicsDiversity, y= Endoo_PlantSpecies, by= "Best_guess_binomial", all.

# Total number of records
length(PREDICTSendooPlants$Source_ID)
```

```
## [1] 70560
```

```
# Total number of species
length(unique(PREDICTSendooPlants$Best_guess_binomial))
```

```
## [1] 632
```

```
# Total number of sourceIDs
length(unique(PREDICTSendooPlants$Source_ID))
```

```
## [1] 39
```

```r
# Total number of studies
length(unique(PREDICTSendooPlants$SS))
```

```
## [1] 51
```

```r
# Total number of sites
length(unique(PREDICTSendooPlants$SSBS))
```

```
## [1] 2519
```

The description of the habitat type is classified at the site level, however, each site has multiple records representing different species sampled. Therefore, it is necessary to get a unique record for each site to properly count how many sites there are for the predominant type and use intensity categories.

```r
# Get the first record for each SSBS (SSBS = Concatenation of Source_ID, Study_number, Block and Site_n
justFirstsEndoPlants <- Number_sites_landuses(PREDICTSendooPlants)

# Number of sites per land-use type and land-use intensity
addmargins(table(justFirstsEndoPlants$Predominant_habitat,justFirstsEndoPlants$Use_intensity), 2)
```

```
## 
##                                             Minimal use Light use
##   Primary forest                                    474       121
##   Primary non-forest                                  6         2
##   Young secondary vegetation                        108        37
##   Intermediate secondary vegetation                  75        86
##   Mature secondary vegetation                        66        10
##   Secondary vegetation (indeterminate age)           14        30
##   Plantation forest                                 232       346
##   Pasture                                           261       150
##   Cropland                                           34         0
##   Urban                                               0         0
##   Cannot decide                                       0         0
## 
##                                             Intense use Cannot decide Sum
##   Primary forest                                    102            68 765
##   Primary non-forest                                  0             0   8
##   Young secondary vegetation                          0             4 149
##   Intermediate secondary vegetation                   2            26 189
##   Mature secondary vegetation                         6             0  82
##   Secondary vegetation (indeterminate age)            0           124 168
##   Plantation forest                                  43            12 633
##   Pasture                                             4             0 415
##   Cropland                                           60            16 110
##   Urban                                               0             0   0
##   Cannot decide                                       0             0   0
```

### 4. Merge the filtered data of frugivores and endozoochorous plants

```r
# Eliminate columns that don't match between the two data frames
PREDICTS_frugivores <- PREDICTS_frugivores[,c(-90, -91)]

# Bind the rows of frugivores to the PREDICTS endozoochorous plants data frame
PREDICTS_frugivores_and_plants <- rbind(PREDICTSendooPlants, PREDICTS_frugivores)

# Total number of sourceIDs
length(unique(PREDICTS_frugivores_and_plants$Source_ID))
```

```
## [1] 99
```

```r
# Total number of studies
length(unique(PREDICTS_frugivores_and_plants$SS))
```

```
## [1] 125
```

```r
# Total number of sites
length(unique(PREDICTS_frugivores_and_plants$SSBS))
```

```
## [1] 4991
```

```r
# Recalculate number of sites

# Get the first record for each SSBS (SSBS = Concatenation of Source_ID, Study_number, Block and Site_n
justFirsts_Frugivores_and_plants <- Number_sites_landuses(PREDICTS_frugivores_and_plants)

# Number of sites per land-use type and land-use intensity
addmargins(table(justFirsts_Frugivores_and_plants$Predominant_habitat,justFirsts_Frugivores_and_plants$
```

```
##
##                                             Minimal use Light use
##   Primary forest                                    942       388
##   Primary non-forest                                 12         6
##   Young secondary vegetation                        173        48
##   Intermediate secondary vegetation                 174       156
##   Mature secondary vegetation                        84       147
##   Secondary vegetation (indeterminate age)           57        51
##   Plantation forest                                 266       706
##   Pasture                                           261       158
##   Cropland                                          112        21
##   Urban                                               0        40
##   Cannot decide                                       0         0
##
##                                             Intense use Cannot decide  Sum
##   Primary forest                                    161           115 1606
##   Primary non-forest                                  9             0   27
##   Young secondary vegetation                          0            86  307
##   Intermediate secondary vegetation                   4            65  399
##   Mature secondary vegetation                         6            24  261
##   Secondary vegetation (indeterminate age)            0           151  259
##   Plantation forest                                  62            26 1060
##   Pasture                                       13            172  604
##   Cropland                                     126            111  370
##   Urban                                          1              0   41
##   Cannot decide                                  0             57   57
```