```
# Gael Sola, FAO, 15/03/2017


# QA/QC CHECKS
# -9- lf indicators match nlcl


######################################################################
# Erase all objects
rm(list=ls())

# Check working directory
getwd()

# remove plots
dev.off()

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


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

# Define working directory
setwd("/media/gs/LACIE SHARE/FAO-2017/COUNTRIES/Bangladesh/BFI_Sund_field/17-02-19_data/data")

### Prepare the tables

# Read tables tree, lf, locations
tree <- read.csv("tree_2.csv",header=T,stringsAsFactors = FALSE)
lf <- read.csv("lf.csv", header=T, stringsAsFactors = FALSE)
loc <-read.csv("locations.csv", header=T, stringsAsFactor = FALSE)
plot_info <- read.csv("plot.csv", header=T, stringsAsFactors = FALSE)
lf_object <- read.csv("lf_object.csv", header=T, stringsAsFactors = FALSE)
sapling <- read.csv("sapling_2.csv", header=T, stringsAsFactors = FALSE)

# set as 100 the veg type for abiotic land feature
lf_object[which(lf_object$object>1),]$vegetation_veg_type <- 100

# Create a unique id for each lf
lf$plot_lf <- paste(lf$plot_plot_id,lf$lf_id, sep="_")
tree$plot_lf <- paste(tree$plot_plot_id, tree$tree_lfid, sep="_")
lf_object$plot_lf <- paste(lf_object$plot_plot_id, lf_object$lf_lf_id, sep="_")

# merge: nlcl from lf
# test <- merge(tree,lf[,c("plot_lf","nlcl")], by="plot_lf", all.x=TRUE)
#summary(as.factor(tree$nlcl))

summary(as.factor(lf$nlcl))
summary(as.factor(lf$nlcl_label))
dim(lf[is.na(lf$nlcl),])[1]

# Convert nlcl to factors to order it
lf$nlcl_2 <- factor(lf$nlcl, levels=c("FEh","BF","FMh","FDp", "S", "H",      # Natural vegetation
                                      "FPr","FP", "OT","OS", "PCs","PCm","SC", # Cultivated tree, shru
                                      "SF","NMF", "FSp","FMp","RP",            # veg flooded
                                      "RS",                                    # Mixed classes
                                      "MF","B","BS","RB",                      # Natural surfaces
                                      "BNl","AP","DS","BR","SP",               # Artificial surfaces (
                                      "R","Ba","BH", "L","FWa","BWa","Po",     # Water (FWa to be adde
                                      "IA","FPd"))                             # Missing

# Check
summary(lf$nlcl_2)
lf[which(is.na(lf$nlcl_2)),]$nlcl
unique(lf[,c("nlcl","nlcl_label")])
#!!! Warning there is a lf with empty nlcl (many empty fields and no objects actually) -> 948_2

# Cleaning lf table
```

```
lf <- lf[which(lf$nlcl != ""),]

#-----------------------------------------------------------------#
# Check: Matching number of land features between lf and lf_object  #
#-----------------------------------------------------------------#

# Number of unique land feature for different tables
length(unique(lf$plot_lf))
length(unique(lf_object$plot_lf))
length(unique(tree$plot_lf))

# Identification of the land feautres in table lf that don't have objects
vec <- unique(lf_object$plot_lf)
missing_obj <- lf[-which(lf$plot_lf %in% vec),]
missing_obj$plot_lf

# Identification of trees with nlcl == NA
tree[which(is.na(tree$nlcl)),]$tree_id
dim(tree[which(tree$tree_id == 0),])
tree[which(tree$tree_id == 0),]$plot_plot_id

### Results:
#     - five land features missing objects: "318_2"  "948_2"  "296_2"  "1764_2" "1546_2"
#     - 2 trees have 0 as tree_id in plot 225

# Number of object per lf
lf_object$count <- 1
lf_nbobj <- aggregate(count~plot_lf,data=lf_object, FUN=sum)
names(lf_nbobj)[2] <- "nb_obj"
table(lf_nbobj$nb_obj)


#-----------------------------------------------------------------#
#-----------------------------------------------------------------#
#
# Assigning nlcl based on land feature objects, tree and  plot info
#
#-----------------------------------------------------------------#
#-----------------------------------------------------------------#


#-----------------------------------------------------------------#
#            Level 1. vegetated or not
#-----------------------------------------------------------------#

# Aggregate objects with min. If min of object is > 2 then not vegetated
lf1 <- aggregate(object~plot_lf, data = lf_object, FUN=min)
table(lf1$object)

# Create the lvl1 class
lf1$lvl1 <- NA
lf1[which(lf1$object == 1),]$lvl1 <- "V"
lf1[which(lf1$object == 2),]$lvl1 <- "A"
lf1[which(lf1$object == 3),]$lvl1 <- "A"

# Merge with lf table
lf1 <- merge(lf, lf1[,c(1,3)], by="plot_lf")
# !!! NOTE: this process removes the 5 lf with no objects.

# Re-order and assign levels
lf1$lvl1 <- factor(lf1$lvl1, levels=c("V", "A"))
summary(lf1$lvl1)



#-----------------------------------------------------------------#
#            Level 2. Terrestrial vs aquatic
#-----------------------------------------------------------------#
```

```
#  Aggregate the objects with max if max of object is 3 then the lf contains water
lf_water <- aggregate(object~plot_lf, data = lf_object, FUN=max)
table(lf_water$object)
lf_water$water <- ifelse(lf_water$object == 3, "water", "no water")

# merge with lf1
lf2 <- merge(lf1, lf_water[,c(1,3)], by="plot_lf")
summary(as.factor(lf2$water))

# Create the lvl 2 class
lf2$lvl2 <- NA
lf2[which(lf2$lvl1 == "V" & lf2$water == "water"),]$lvl2 <- "VAq"
lf2[which(lf2$lvl1 == "V" & lf2$water == "no water"),]$lvl2 <- "VT"
lf2[which(lf2$lvl1 == "A" & lf2$water == "water"),]$lvl2 <- "W"
lf2[which(lf2$lvl1 == "A" & lf2$water == "no water"),]$lvl2 <- "AT"

# Re-order and assign levels
lf2$lvl2 <- factor(lf2$lvl2, levels=c("VT", "VAq", "AT", "W"))
summary(lf2$lvl2)

# # Method 2: Add aqua form GIS file
# aqua <- read.csv("QAQC/BFI_Plot_Terr_Aqua.csv", header=T)
# str(aqua)
# names(aqua)[1] <- "plot_plot_id"
# lf_aqua <- merge(lf, aqua, by="plot_plot_id")
# summary(lf_aqua$Terr.Aqua)


#-------------------------------------------------------------------#
#   Level 3. natural vs cultivated (semi natural goes to natural)
#-------------------------------------------------------------------#

#------#
# FOR VEGETATION  ONLY (VT and Vaq)
#------#

###
### Control than no lf have both natural and cultivated at the same time
###

# Select only the objects vegetation and tree
test <- lf_object[which(lf_object$object == 1 & lf_object$vegetation_veg_type == 1),]
test$count <-1

# count the number of object trees per lf
test2 <- aggregate(count~plot_lf, data=test, FUN=sum)
names(test2)[2] <- "flag"
table(test2$flag)

# Retrieve the land feature id if flag > 1
test3 <-merge(test, test2, by="plot_lf")
unique(test3[which(test3$flag > 1),]$plot_lf)

#!!! results: 2 lf with issues:
#          - 130_1: 2 object tree, one semi natural and one plantation. no cover perc
#          - 247_1: 2 object tree,  both labeled plantation but one cat semi natural, the other is n
# RULE => if one natural or semi natural object described the whole lf is categorised semi nat (use
# RULE 2 => if tree is nat and shrub is cultiv, the overall class is nat, tree>shrub>grass

#!!! Aditional check: for plot_lf 1769_2 and 481_1, water with veg type tree entered, happened sever

###
### Assigning lvl3
###

# Select only vegetation
lf_veg <- lf_object[which(lf_object$object == 1),]
length(unique(lf_veg$plot_lf))
```

```
# Create an indicator combining veg_type and artif
lf_veg$combine <- lf_veg$vegetation_veg_type*10+lf_veg$vegetation_artif

# Combine veg_type (priority tree>shrub>grass and nat>culti)
lf_artif <- aggregate(combine~plot_lf, data=lf_veg, FUN=min)
summary(as.factor(lf_artif$combine))

# Retrieve artif for all veg_type
lf_artif$artif <- lf_artif$combine - round(lf_artif$combine/10,0)*10
summary(as.factor(lf_artif$artif))

# Control
#lf_artif[which(lf_artif$plot_lf == "130_1"),]$artif

# Create lvl3 for vegetation
lf3 <- merge(lf2, lf_artif, by="plot_lf", all.x=TRUE)

# Assign lvl2 to lvl3 for the non veg
lf3$lvl3 <- as.character(lf3$lvl2)

# Assign lvl3 for veg
lf3[which(lf3$lvl3=="VT" & lf3$artif == 1),]$lvl3 <- "NV"
lf3[which(lf3$lvl3=="VT" & lf3$artif == 2),]$lvl3 <- "CV"
lf3[which(lf3$lvl3=="VAq" & lf3$artif == 1),]$lvl3 <- "NVF"
lf3[which(lf3$lvl3=="VAq" & lf3$artif == 2),]$lvl3 <- "CVF"

# Re-order and assign levels
lf3$lvl3 <- factor(lf3$lvl3, levels=c("NV", "CV", "NVF", "CVF", "AT", "W"))
summary(lf3$lvl3)

#------#
# FOR NON VEGETATED AREA (AT or W)
#------#

# TO BE DONE

#---------------------------------------------------------------------#
#   Level 4. tree, shrub or herb dominated area
#---------------------------------------------------------------------#

#------#
# FOR VEGETATION ONLY (NV, CV, NVF, CVF)
#------#

# The variable combine contains the info about tree (1) shrub (2) or herb (3)
table(lf3$combine)

# Copy lf3 to lf4
lf4 <- lf3

# Assign lvl4
lf4$lvl4 <- as.character(lf4$lvl3)
lf4[which(lf4$lvl3 == "NV" & lf4$combine == 11),]$lvl4 <- "F"
lf4[which(lf4$lvl3 == "NV" & lf4$combine == 21),]$lvl4 <- "S"
lf4[which(lf4$lvl3 == "NV" & lf4$combine == 31),]$lvl4 <- "H"
lf4[which(lf4$lvl3 == "CV" & lf4$combine == 12),]$lvl4 <- "CT"
lf4[which(lf4$lvl3 == "CV" & lf4$combine == 22),]$lvl4 <- "CS"
lf4[which(lf4$lvl3 == "CV" & lf4$combine == 32),]$lvl4 <- "CH"
lf4[which(lf4$lvl3 == "NVF" & lf4$combine == 11),]$lvl4 <- "NFT"
lf4[which(lf4$lvl3 == "NVF" & lf4$combine == 21),]$lvl4 <- "NFS"
lf4[which(lf4$lvl3 == "NVF" & lf4$combine == 31),]$lvl4 <- "NFH"
lf4[which(lf4$lvl3 == "CVF" & lf4$combine == 12),]$lvl4 <- "FTc"
lf4[which(lf4$lvl3 == "CVF" & lf4$combine == 22),]$lvl4 <- "Fc"
# lf4[which(lf4$lvl3 == "CVF" & lf4$combine == 32),]$lvl4 <- "PR"

# Re-order and assign levels
lf4$lvl4 <- factor(lf4$lvl4, levels=c("F", "S", "H", "CT", "CS", "CH",
                                      "NFT", "NFS", "NFH", "FTc", "Fc", #"PR",
                                      "AT", "W"))
```

```
summary(lf4$lvl4)

# Check decision tree
summary(lf4$lvl1)
summary(lf4$lvl2)
summary(lf4$lvl3)
summary(lf4$lvl4)

#------#
# For non vegetated areas
#------#

# TO BE DONE

#---------------------------------------------------------------------#
#         Level 5. Forest type
#---------------------------------------------------------------------#

#------#
# FOR VEGETATION ONLY
#------#

###
### For natural trees (F), need to go back to tree measurements and retrieve dipterocarps
###

# Select a subset of the table tree for all trees in lf "F"
vec <- lf4[which(lf4$lvl4 == "F"),]$plot_lf
tree_sub <- tree[which(tree$plot_lf %in% vec),]

# create counting indicator for all trees and for Dipterocarp
tree_sub$count <- 1
tree_sub$dip <- ifelse(tree_sub$family == "DIP",1,0)
sum(tree_sub$count)
sum(tree_sub$dip)

# aggregate numbers of trees per land feature
lf_count <- aggregate(count~plot_lf, data=tree_sub, FUN=sum)
lf_dip <- aggregate(dip~plot_lf, data=tree_sub, FUN=sum)
lf_ftype <- merge(lf_count,lf_dip, by="plot_lf")

# create ftype classes
lf_ftype$ftype <- ifelse(lf_ftype$dip == 0, "evergreen",
                         ifelse(lf_ftype$count-lf_ftype$dip == 0, "dip","mixed")
                         )

# Merge with lf4
lf5 <- merge(lf4,lf_ftype, by="plot_lf", all.x=T)
table(lf5$ftype)

# Identify the land features in lvl4 "F" but no trees measured
lf5[which(lf5$lvl4 == "F" & is.na(lf5$ftype)),]$plot_lf

# Assign lvl5
lf5$lvl5 <- as.character(lf5$lvl4)
lf5[which(lf5$lvl5 == "F" & lf5$ftype == "evergreen"),]$lvl5 <- "FE"
lf5[which(lf5$lvl5 == "F" & lf5$ftype == "mixed"),]$lvl5 <- "FM"
lf5[which(lf5$lvl5 == "F" & lf5$ftype == "dip"),]$lvl5 <- "FD"

# Re-order lvl5 and assign levels
lf5$lvl5 <- factor(lf5$lvl5, levels=c("FE","FM","FD","F", "S", "H", "CT", "CS", "CH",
                                      "NFT", "NFS", "NFH", "FTc", "FSc", "PR",
                                      "AT", "W"))

summary(lf5$lvl5)

# Check if there is info in sapling
# Select sapling data for F land features
vec <- lf5[which(lf5$lvl4 == "F" & is.na(lf5$ftype)),]$plot_plot_id
```

```
sapling_sub <- sapling[which(sapling$plot_plot_id %in% vec),]
table(sapling_sub$plot_plot_id)

# Count the number of trees and dipterocarp per plot
sapling_sub$count <- 1
sapling_sub$dip <- ifelse(sapling_sub$family == "DIP", 1, 0)
sap_count <- aggregate(count~plot_plot_id, data=sapling_sub, FUN=sum)
sap_dip <- aggregate(dip~plot_plot_id, data=sapling_sub, FUN=sum)

# Associate forest class
sap_ftype <- merge(sap_count,sap_dip, by="plot_plot_id")
sap_ftype$ftype_sap <- ifelse(sap_ftype$dip == 0, "evergreen",
                              ifelse(sap_ftype$count-sap_ftype$dip == 0, "dip","mixed")
)
table(sap_ftype$ftype_sap)

# Merge lf5 with sap_ftype tables
lf5b <- merge(lf5, sap_ftype, by="plot_plot_id", all.x=T)

# Assign lvl5 when sapling info helps
lf5b[which(lf5b$lvl5 == "F" & lf5b$ftype_sap == "evergreen"),]$lvl5 <- "FE"
lf5b[which(lf5b$lvl5 == "F" & lf5b$ftype_sap == "mixed"),]$lvl5 <- "FM"
lf5b[which(lf5b$lvl5 == "F" & lf5b$ftype_sap == "dip"),]$lvl5 <- "FD"

summary(lf5b$lvl5)

# lvl5 not available:
lf5b[which(lf5b$lvl5 == "F"),]$plot_lf

# NOTE:
# -> 231 innacessible (LCCS only) cannot go further than natural forest F
# -> 475_2 sampled completely, might be lf containing trees but trees not in the plot.
#    Need to check the pictures.

###
### For cultivated trees (CT), depends on orchards.
###

# extract lf_object with cultivated trees
vec <- lf5b[which(lf5b$lvl5 == "CT"),]$plot_lf
lf_ct <- lf_object[which(lf_object$plot_lf %in% vec),c("plot_lf","vegetation_management")]

# assign orchard or other plantation
lf_ct$orch <- ifelse(lf_ct$vegetation_management == 9, 1, 0)
lf_ct <- aggregate(orch~plot_lf, data=lf_ct, FUN=min)
table(lf_ct$orch)
lf_ct$orch <- ifelse(lf_ct$orch==0, "other","orchard")

# merge with land features
lf5c <- merge(lf5b, lf_ct, by="plot_lf", all.x=TRUE)

# Assign lvl5 for CT (subdivide into FP and OT)
# Convert lvl5 to character
lf5c$lvl5 <- as.character(lf5c$lvl5)

# Assign FP and OT
lf5c[which(lf5c$lvl5 == "CT" & lf5c$orch == "orchard"),]$lvl5 <- "OT"
lf5c[which(lf5c$lvl5 == "CT" & lf5c$orch == "other"),]$lvl5 <- "FP"

# Convert back to factor
lf5c$lvl5 <- factor(lf5c$lvl5, levels=c("FE","FM","FD","F", "S", "H", "FP", "OT","CS", "CH",
                                       "NFT", "NFS", "NFH", "FTc", "FSc", "PR",
                                       "AT", "W"))

# Check
summary(lf5b$lvl5)
summary(lf5c$lvl5)

###
```

```
### For cultivated shrubs (CS) direct conversion from CS to OS
###
lf5d <- lf5c
levels(lf5d$lvl5)[9] <- "OS"
summary(lf5d$lvl5)


###
### For herbaceous crops (CH)
###

# TO BE DONE (WHERE TO GET TEMP VS PERMANENT???)


###
### for tree, shrub and herb dominated areas (aquatic) (NFT, NFS, NFH)
### And tree and shrub crop (FTc, FSc)  -> salinity
###

# select all objects for the land features targeted
vec <- c(unique(lf5d[which(lf5d$lvl3 == "NVF"),]$plot_lf),unique(lf5d[which(lf5d$lvl4 == "FTc"),]$pl
salin <- lf_object[which(lf_object$plot_lf %in% vec),]

# Checks
length(unique(salin$plot_lf))
table(salin$water_details_water_sal)

# aggregate objects to land feature with max.
salin <- aggregate(water_details_water_sal~plot_lf, data=salin, FUN=max)

# Checks
table(salin$water_details_water_sal)

# Merge salin with lf5d
lf5e <- merge(lf5d, salin, by="plot_lf", all.x=TRUE)

# Assign lvl5
table(lf5e$lvl5)
lf5e$lvl5 <- as.character(lf5e$lvl5)

lf5e[which(lf5e$lvl5 == "NFT" & lf5e$water_details_water_sal == 1),]$lvl5 <- "SF"
lf5e[which(lf5e$lvl5 == "NFT" & lf5e$water_details_water_sal == 2),]$lvl5 <- "NMF"
lf5e[which(lf5e$lvl5 == "NFS" & lf5e$water_details_water_sal == 1),]$lvl5 <- "SWr"
lf5e[which(lf5e$lvl5 == "NFS" & lf5e$water_details_water_sal == 2),]$lvl5 <- "MS"
lf5e[which(lf5e$lvl5 == "NFH" & lf5e$water_details_water_sal == 1),]$lvl5 <- "MA"
lf5e[which(lf5e$lvl5 == "NFH" & lf5e$water_details_water_sal == 2),]$lvl5 <- "SM"
lf5e[which(lf5e$lvl5 == "FTc" & lf5e$water_details_water_sal == 1),]$lvl5 <- "FSp"
lf5e[which(lf5e$lvl5 == "FTc" & lf5e$water_details_water_sal == 2),]$lvl5 <- "FMp"

# Convert back to factor
lf5e$lvl5 <- factor(lf5e$lvl5, levels=c("FE","FM","FD","F", "S", "H",      # Natural vegetation
                                        "FP", "OT","OS", "CH",             # Cultivated vegetation
                                        "SF","NMF", "SWr", "MS","MA", "SM", # Nat veg flooded
                                        "FSp","FMp","FSc", "PR",           # cultiv veg flooded
                                        "AT", "W"))                        # non forest

# Check
summary(lf5d$lvl5)
summary(lf5e$lvl5)

###
### For flooded shrub crop (FSc) direct conversion to reed plantation (RP)
###
lf5f <- lf5e
levels(lf5f$lvl5)[19] <- "RP"
summary(lf5f$lvl5)


###
```

```
### For paddy rice (PR) aggregate columns crop details
###

# Retrieve all land feature fro PR
vec <- unique(lf5f[which(lf5f$lvl5 == "PR"),]$plot_lf)
rice <- lf_object[which(lf_object$plot_lf %in% vec),]

# Check
length(unique(rice$plot_lf))

# Check if there is rice everywhere
rice$rice1 <- ifelse(rice$vegetation_crop_details_crop_label.1. == "Rice",1,0)
rice$rice2 <- ifelse(rice$vegetation_crop_details_crop_label.2. == "Rice",1,0)
rice$rice3 <- ifelse(rice$vegetation_crop_details_crop_label.3. == "Rice",1,0)
rice$rice <- rice$rice1 + rice$rice2 + rice$rice3
rice_check <- aggregate(rice~plot_lf, data=rice, FUN=max)
table(rice_check$rice)

# Count the number of crops in the same object
rice$crop1 <- ifelse(rice$vegetation_crop_details_crop_label.1. != "",1,0)
rice$crop2 <- ifelse(rice$vegetation_crop_details_crop_label.2. != "",1,0)
rice$crop3 <- ifelse(rice$vegetation_crop_details_crop_label.3. != "",1,0)
rice$crop <- rice$crop1 + rice$crop2 + rice$crop3
crop_check <- aggregate(crop~plot_lf, data=rice, FUN=max)
table(crop_check$crop)

# Merge with lf5f
lf5g <- merge(lf5f,crop_check, by="plot_lf", all.x=T)

# Assign the lvl5 for paddy rice PR: PRs and PRm
lf5g$lvl5 <- as.character(lf5g$lvl5)

lf5g[which(lf5g$lvl5 == "PR" & lf5g$crop == 1),]$lvl5 <- "PRs"
lf5g[which(lf5g$lvl5 == "PR" & lf5g$crop > 1),]$lvl5 <- "PRm"

# Convert back to factor
lf5g$lvl5 <- factor(lf5g$lvl5, levels=c("FE","FM","FD","F", "S", "H",      # Natural vegetation
                                        "FP", "OT","OS", "CH",             # Cultivated vegetation
                                        "SF","NMF", "SWr", "MS","MA", "SM", # Nat veg flooded
                                        "FSp","FMp","RP", "PRs","PRm",    # cultiv veg flooded
                                        "AT", "W"))                        # non forest

# Check
summary(lf5f$lvl5)
summary(lf5g$lvl5)

#------#
# FOR NON VEGETATED AREAS
#------#

# TO BE DONE


#---------------------------------------------------------------------#
#                 Level 6. various
#---------------------------------------------------------------------#

#------#
# FOR NATURAL FOREST
#------#

# Level change without option: FE => FEh, FM => FMh, FD => FDp
lf6 <- lf5g
lf6$lvl6 <- lf6$lvl5
levels(lf6$lvl6)[1] <- "FEh"
levels(lf6$lvl6)[2] <- "FMh"
levels(lf6$lvl6)[3] <- "FDp"

summary(lf6$lvl6)
```

```
table(as.character(lf6$lvl6))
table(lf6$nlcl)

#------#
# For Forest plantation FP
#------#

# Plantation based on pheno : FP -> FPe, FPm, FPd. Using tree data to detect dipterocarp

# Extract tree data for FP
vec <- unique(lf6[which(lf6$lvl6 == "FP"),]$plot_lf)
tree_sub <- tree[which(tree$plot_lf %in% vec),]

# create counting indicator for all trees and for Dipterocarp
tree_sub$count <- 1
tree_sub$dip <- ifelse(tree_sub$family == "DIP",1,0)
sum(tree_sub$count)
sum(tree_sub$dip)

# aggregate numbers of trees per land feature
lf_count <- aggregate(count~plot_lf, data=tree_sub, FUN=sum)
lf_dip <- aggregate(dip~plot_lf, data=tree_sub, FUN=sum)
lf_ftype <- merge(lf_count,lf_dip, by="plot_lf")

# create ftype classes
lf_ftype$ftype_p <- ifelse(lf_ftype$dip == 0, "evergreen",
                           ifelse(lf_ftype$count-lf_ftype$dip == 0, "dip","mixed")
)

# Merge with lf6
lf6b <- merge(lf6,lf_ftype[,c("plot_lf","ftype_p")], by="plot_lf", all.x=T)
table(lf6b$ftype_p)

# Identify the land features in lvl5 "FP" but no trees measured
lf6b[which(lf6b$lvl5 == "FP" & is.na(lf6b$ftype_p)),]$plot_lf

# Check if there is info in sapling
# Select sapling data for F land features
vec <- lf6b[which(lf6b$lvl5 == "FP" & is.na(lf6b$ftype_p)),]$plot_plot_id
sapling_sub <- sapling[which(sapling$plot_plot_id %in% vec),]
table(sapling_sub$plot_plot_id)

# Count the number of trees and dipterocarp per plot
sapling_sub$count <- 1
sapling_sub$dip <- ifelse(sapling_sub$family == "DIP", 1, 0)
sap_count <- aggregate(count~plot_plot_id, data=sapling_sub, FUN=sum)
sap_dip <- aggregate(dip~plot_plot_id, data=sapling_sub, FUN=sum)

# Associate forest class
sap_ftype <- merge(sap_count,sap_dip, by="plot_plot_id")
sap_ftype$ftype_sap_p <- ifelse(sap_ftype$dip == 0, "evergreen",
                                ifelse(sap_ftype$count-sap_ftype$dip == 0, "dip","mixed")
)
table(sap_ftype$ftype_sap_p)

# Merge with lf6
lf6c <- merge(lf6b, sap_ftype, by="plot_plot_id", all.x=TRUE)
table(lf6c$ftype_sap_p)

# Assign lvl6
lf6c$lvl6 <- as.character(lf6c$lvl6)
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_p == "evergreen"),]$lvl6 <- "FPe"
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_p == "mixed"),]$lvl6 <- "FPm"
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_p == "dip"),]$lvl6 <- "FPd"

# Assign lvl6 when sapling info helps
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_sap_p == "evergreen"),]$lvl6 <- "FPe"
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_sap_p == "mixed"),]$lvl6 <- "FPm"
lf6c[which(lf6c$lvl6 == "FP" & lf6c$ftype_sap_p == "dip"),]$lvl6 <- "FPd"
```

```
# Re-order lvl5 and assign levels
lf6c$lvl6 <- factor(lf6c$lvl6, levels=c("FEh","FMh","FDp","F", "S", "H",      # Natural vegetation
                                        "FPe","FPm","FPd","FP", "OT","OS", "CH",             # Culti
                                        "SF","NMF", "SWr", "MS","MA", "SM", # Nat veg flooded
                                        "FSp","FMp","RP", "PRs","PRm",    # cultiv veg flooded
                                        "AT", "W"))                        # non forest

# Check
summary(lf6c$lvl5)
summary(lf6c$lvl6)


# lvl6 not available:
lf6c[which(lf6c$lvl6 == "FP"),]$plot_lf

# plot 366 was only partially sampled
# plot 403 sampled completely, need to be check why falls in forest plantation but no tree measured
# plot 1021 sampled completely, same as above.

#------#
# FOR CROPS: PC and SC
#------#

# TO BE DONE (needs PC and SC first)


#-----------------------------------------------------------------#
#     Level 7. Rotation
#-----------------------------------------------------------------#

# To be done

#-----------------------------------------------------------------#
#     Level 8. Multiple strata: AgroForestry and Rural Settlement
#-----------------------------------------------------------------#

# To be done

#-----------------------------------------------------------------#
#                              RESULTS
#-----------------------------------------------------------------#


table(lf6c$nlcl_2,lf6c$lvl6)

write.csv(lf6c,file="QAQC/nlcl_reclassification.csv", row.names = FALSE)
write.csv(table(lf6c$nlcl_2, lf6c$lvl6),file="QAQC/nlcl_compa.csv")




#########################################################
#########################################################
#########################################################
#########################################################
```