```
##############################################################################
## R-script for Data corrections of Bangladesh Forest Inventory
## NOTE VERY IMPORTANT, BFI_QAQC script should have
## been run before launching this manuscript
## L. Saint-André INRA/CIRAD, Md. Akhter Hossain and BFI-HQ group 06/04/2017
##############################################################################

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

#function to recall the plot_id with necessary other variables where error is detected
#       to checking the error
recall_plot_lf <- function(number_of_the_plot){
  plot_id_lf<-lf[which(lf$plot_plot_id==number_of_the_plot), c("lf_id", "lf_details_lf_status_label"
  return(plot_id_lf)
}

#
recall_plot_lf_item <- function(file_to_find,code_number){
  test<-variables[which(variables$File==file_to_find),]
  test<-test[which(test$item_code==code_number), "item_label_en"]
  return(test)
}

##############################################################################################
#Read object code
#setwd("C:/Users/Akhter/Google Drive/5_Training Materials/R_Dhaka/Nacessary Materials")
setwd("C:/LSA2017/LSA/Projets/BiomassFormation/Bengladesh/Nacessary Materials")
variables<- read.csv("./BFI-DB/variables.csv", header = T, stringsAsFactors = FALSE)

#-------------------------------------------------------------------------------------------#
# CHECK 3: Subplot and lf
#-------------------------------------------------------------------------------------------#

#-------------------------------------------------------------------------------------------#
#- CHECK 3.1: number of plot matching between plot and lf
#-------------------------------------------------------------------------------------------#

# Correction procedure:
# Step 1: Be inform about the non-sample plots and Keep the plots
#         in the database,
# Step 2: Take a new random plot in the correspondig hexagonal patch as new plot ID
#         and measured them

#-------------------------------------------------------------------------------------------#
# CHECK 3.2: Consistency between lf and lf_Object
#-------------------------------------------------------------------------------------------#
# Correction procedure:
# Step 1: plots should be checked at BFI HQ by checking land feature photgraph and the sketch of the
#         If successful see Step 3
# If not successful Step2: by cold check by revisiting the plot location
# Step 3: manual correction directly into OPENFORIS (too many variables to enter, list needed)
# NOTE: The tables lf, lf_object (from the QAQC procedure) are required

# Example Correction for plot 1142 (NOTE ARBITRARY DATA FILLING, should be perfomed by Falgoonee)

# just a recall of LFs of the plot (information)
recall_plot_lf(1142)

# code to be filled for all LF of the plot
# LF number 1
lf_object_new<-data.frame(plot_plot_id=1142,
                          lf_lf_id=1,
                          object_id=1,
                          object=1,
                          vegetation_veg_type=1,
                          vegetation_artif=2,
                          vegetation_cover_perc=8,
                          vegetation_maturity=6,
```

```
                          vegetation_management=7,
                          vegetation_treatment.1.=0,
                          vegetation_rotation=3,
                          vegetation_crop_details_crop.1.=NA,
                          vegetation_crop_details_cultivation=NA,
                          vegetation_crop_details_crop_watersup.1.=NA,
                          non_veg_details_non_veg=NA,
                          non_veg_details_non_veg_perc=NA,
                          water_details_water=NA,
                          water_details_water_sal=NA,
                          water_details_water_perc=NA)

#To add columns for labels of each variables
if(!is.na(lf_object_new$object)){lf_object_new$object_label<-recall_plot_lf_item("object",lf_object_
if(!is.na(lf_object_new$vegetation_veg_type)){lf_object_new$vegetation_veg_type_label<-recall_plot_l
if(!is.na(lf_object_new$vegetation_artif)){lf_object_new$vegetation_artif_label<-recall_plot_lf_iten
if(!is.na(lf_object_new$vegetation_cover_perc)){lf_object_new$vegetation_cover_perc_label<-recall_pl
if(!is.na(lf_object_new$vegetation_maturity)){lf_object_new$vegetation_maturity_label<-recall_plot_l
if(!is.na(lf_object_new$vegetation_management)){lf_object_new$vegetation_management_label<-recall_pl
if(!is.na(lf_object_new$vegetation_treatment.1.)){lf_object_new$vegetation_treatment_label.1.<-recal
if(!is.na(lf_object_new$vegetation_rotation)){lf_object_new$vegetation_rotation_label<-recall_plot_l
if(!is.na(lf_object_new$vegetation_crop_details_crop.1.)){lf_object_new$vegetation_crop_details_crop
if(!is.na(lf_object_new$vegetation_crop_details_cultivation)){lf_object_new$vegetation_crop_details_
if(!is.na(lf_object_new$vegetation_crop_details_crop_watersup.1.)){lf_object_new$vegetation_crop_det
if(!is.na(lf_object_new$non_veg_details_non_veg)){lf_object_new$non_veg_details_non_veg_label<-recal
if(!is.na(lf_object_new$non_veg_details_non_veg_perc)){lf_object_new$non_veg_details_non_veg_perc_la
if(!is.na(lf_object_new$water_details_water)){lf_object_new$water_details_water_label<-recall_plot_l
if(!is.na(lf_object_new$water_details_water_sal)){lf_object_new$water_details_water_sal_label<-recal
if(!is.na(lf_object_new$water_details_water_perc)){lf_object_new$water_details_water_perc_label<-rec

# To combine the row (lf_object_new) with lf_object
lf_object <- smartbind(lf_object,lf_object_new)


#-------------------------------------------------------------------------------#
#- CHECK 3.3: Matching between lf (non vegetated) and tree presence
#             (data: lf.csv and tree.csv)
#-------------------------------------------------------------------------------#
#  Step 1: plots should be checked in BFI HQ by checking the non-vegetated object details
#          wether it is roads, settlements etc. If non-vegetated object details are consistent
#          then no more correction is needed. If not successfull see step 2
#  Step 2: Check the lf photographs in BFI HQ to identify the lf object details. If successfull
#          then do correction by following r-script. If not successful see step 3.
#  Step 3: by cold check by revisiting the plot location


#-------------------------------------------------------------------------------#
#- C3.4: Matching between vegetation type (vegetated) and tree presence
#        (data: lf-object.csv and tree_sub.csv)
#-------------------------------------------------------------------------------#
# Step 1: Check the lf photograph in the BFI HQ to see if the vegetation type is
#         selected properly or not. If it is selected properly then it is ok but if
#         it is different then the selected vegetation type then make correction using
#         following r-script. If not successful see the step 2.
# Step 2: Do cold check by revisiting the plot location
#

#-------------------------------------------------------------------------------------------#
#- C3.5: Compare leaf coverage with extimated crown cover (data: subplot_info.csv and lf.csv)
#        [leaf coverage by land feature is estimated by using the data from lf_subp_prop]
#-------------------------------------------------------------------------------------------#
# Step 1: Do cold check for the plots having "crown cover max is 100 but leaf cover is 0"
#         as well as "crown cover is 0 but leaf cover is not 0" by revisiting
#         the plot location

#-------------------------------------------------------------------------#
#- C3.6: Absence and not recording reference point for all plots
#        (data: plot_info.csv)
#-------------------------------------------------------------------------#
```

```
#Step 1: Check the land feature photographs in the BFI HQ, if no reference point
#         is observed then it is ok. But if any reference point is observed then
#         see step 2
#Step 3:  Cold check by revisiting the plot
#Step 2: Manual correction directly into OPENFORIS (too many possible variables to enter, list neede

#-------------------------------------------------------------------------------------#
#- C3.7: To see consistency of distance between reference point and plot centre
#        (data: plot_info.csv)
#-------------------------------------------------------------------------------------#
# Step 1: If the distance between RP and plot centre is more than 200 then do cold check
#         by revisiting the plots and change the

#-------------------------------------------------------------------------------------#
#- C4.1 and C4.2: no DBH or no Height in tree and spaling datasets
#-------------------------------------------------------------------------------------#
#remove stump and dead trees
tree_live <- tree[which(tree$tree_status == 1),]

#remove saplings with no DBH and plots with no sapling
# NOTE that saplings with no-dbh should be verified
# NOTE that saplings with no height can be estimated in DAAN procedures
saplings_live <- saplings[which((!is.na(saplings$sap_dia)) & (saplings$sap_notes != "No sapling foun

#-------------------------------------------------------------------------------------#
#- C4.6: Outliers in H_D relationship
#-------------------------------------------------------------------------------------#
# Step 1: Height of the outliers is set to "NA". It will be estimated from the H-D models
#         in the Analysis script

# plot dbh - h with outliers
p_hd <- ggplot(tree_live)
p_hd <- p_hd + geom_point(aes(x=tree_dia, y=tree_total_lgt))
p_hd <- p_hd + xlim(x_lim[1],x_lim[2])
p_hd <- p_hd + ylim(y_lim[1],y_lim[2])
p_hd <- p_hd + xlab("tree diameter at breast height (cm)") + ylab("tree total height (m)")
p_hd

# NOTE: The tables tree_Outlierstore, tree_live (from the QAQC procedure) are required
tree_live <- tree_live[order(tree_live$plot_plot_id, tree_live$subplot_subp_id,tree_live$tree_id),]
for(i in 1:dim(tree_outlierstore)[1]){
  plot_id_of_outlier = tree_outlierstore$plot_id[i]
  subplot_id_of_outlier = tree_outlierstore$subplot_subp_id[i]
  tree_id_of_outlier = tree_outlierstore$tree_id[i]
  for (j in 1:dim(tree_live)[1]){
    if (tree_live$plot_plot_id[j]==plot_id_of_outlier){break}
  }
  for (k in j:dim(tree_live)[1]){
    if (tree_live$subplot_subp_id[k]==subplot_id_of_outlier){break}
  }
  for (l in k:dim(tree_live)[1]){
    if (tree_live$tree_id[l]==tree_id_of_outlier){break}
  }
  tree_live$tree_total_lgt[l]=NA
}

# Verification that outliers have been removed
# plot dbh - h without outliers
p_hd <- ggplot(tree_live)
p_hd <- p_hd + geom_point(aes(x=tree_dia, y=tree_total_lgt))
p_hd <- p_hd + xlim(x_lim[1],x_lim[2])
p_hd <- p_hd + ylim(y_lim[1],y_lim[2])
p_hd <- p_hd + xlab("tree diameter at breast height (cm)") + ylab("tree total height (m)")
p_hd
```