

```

#####-----#####
## R-script for Quality Assurance and Quality Checking of Bangladesh Forest Inventory ##
##                               soil and litter data                               ##
#####-----#####
# Md. Akhter Hossain, Forest Inventory Data Analysis, FAO of the UN, December 2017

#list of checks:
#CHECK 01: Check the compatibility of bulk density data by forest zone
#           considering the soil condition of Bangladesh in order to identify
#           the outliers
#           [input: soil bulk density data; output: list of plots, subplot
#           with strange value (value > 2.0 g/cc) of bulk density data]
#CHECK 02: Check if the number of bulk density data for different soil
#           depths/layer in different forest zones is appropriate or not as
#           per BFI manual
#           [input: bulk density data; output: the number of soil depth/layer
#           not appropriate as per BFI manual]
#CHECK 03: To check, in case of completely accessible or surveyed plots,
#           whether the number of bulk density data reported from 3 subplots
#           or not.
#           [input: bulk density data; output: number of completely sampled
#           plots with < 3 subplots from which soil bulk density was reported
#           and list of those plots]
#CHECK 04: To see the soil organic carbon values, if they are within normal
#           range or not
#           [input: Soil organic carbon data; output: outliers in organic
#           carbon data and number of soil layers in which outliers (strange)
#           data was found]
#CHECK 05: To identify i) plots having only soil carbon data but no BD data, and
#           ii) plots with mismatched land feature in bulk density and soil carbon
#           data
#           [input: Soil BD and organic carbon data, output: list of plots
#           with only organic carbon data but no BD data, plots with nonmatching
#           land feature number]
#CHECK 06: To see the relationship between bulk density and soil carbon and
#           evaluate visually if there is any abnormal relationship
#           [input: Soil organic carbon data, output: graphs indicating the
#           relationship between soil organic carbon and bulk density]
#CHECK 07: To see if there is any outlier in soil carbon (t/ha) at
#           different forest zones
#           [Input: soil organic carbon data; output: soil organic carbon
#           in t/ha by soil depths and forest zones]
#CHECK 08: To identify the plots in different zone from which soil carbon
#           data is reported other than the specified soil layers as per
#           BFI manual
#           [Input: soil organic carbon, output: list of plots where soil
#           organic carbon is reported from soil layers other than specified
#           by BFI manual]
#CHECK 09: To identify the plots and subplots in which the soil
#           sample is collected from layers other than specified by BFI manual
#           in different forest zone [2 defined layers in hill, village and
#           Sal forests are 0-15cm and 15-30 cm; 3 specified layers in
#           Sundarbans and coastal forests are 0-15 cm, 15-30cm and 30-100 cm]
#           [Input: Soil texture data; output: graph presenting soil texture
#           sample collection outside the specified soil depths and number of
#           subplots and plots where sample was collected outside the
#           specified soil layers/depths as per BFI manual]
#CHECK 10: To identify the layers if it is not matched with BFI soil survey
#           design or if any unknown layers is written in the data sheet
#           [Input: Soil texture data; Output: list of plots in which soil is
#           collected from undefined soil layers]
#CHECK 11: To identify the outliers in percentages of clay, silt and sandy
#           particles in soil [i) if the sum of sand, silt and clay percentage
#           is 100 or not; ii) if there are any outliers/negative values.]
#           [Input: soil texture data; output: List of plots and layers where
#           the percentages of clay, silt and sand particles of same sample
#           is not 100]
#CHECK 12: To see if there is any outlier in the litter carbon (%)

```

```

# [Input: Litter carbon data; output: graph (boxplot) presenting
# outliers in the litter carbon data]
#CHECK 13: To see the relationship in order to check the consistency between
# canopy coverage and leaf oven dry weight
# [Input: Litter dry weight data; output: Box plot with outliers in
# litter dry weight data based on land class legends and list of
# plots with strange values in litter weight)
#CHECK 14: To see if there is any subplots having more than one litter dry wt
# data
# [Input: Litter dry weight data;
# output: list of plots and subplots having more than one litter
# dry wt data]
#CHECK 15: Identify the plots with mismatched land feature id in litter carbon and
# litter dry weight data
# [Input: lit_car.csv and lit_dry_wt.csv, lf.csv
# Output: list of plots with mismatches with lf.csv by land feature id]

# Check working directory
getwd()

# remove plots
dev.off()

#Erase memory
rm(list=ls())

#install.packages("dplyr")

library(ggplot2)

# Define working directory
setwd("C:/Users/Md. Akhter Hossain/Google Drive/5_Training Materials/R_Dhaka/Necessary Materials")

#####

# Read tables tree, lf, plot, species, lf_object
lf <- read.csv("./BFI-DB/lf.csv", header=T, stringsAsFactors = FALSE)
plot_info <- read.csv("./BFI-DB/plot.csv", header=T, stringsAsFactors = FALSE)
subplot_info <- read.csv("./BFI-DB/subplot.csv", header=T, stringsAsFactors = F)
soil_bulk <- read.csv("./BFI_soil_data/Soil_bulk_density.csv", header=T, stringsAsFactors = F)
lit_car <- read.csv("./BFI_soil_data/Litter_carbon.csv", header = T, stringsAsFactors = F)
lit_dry_wt <- read.csv("./BFI_soil_data/Litter_dry_weight.csv", header =T, stringsAsFactors = F)
soil_car <- read.csv("./BFI_soil_data/Soil_carbon.csv", header=T, stringsAsFactors = F)
soil_tex <- read.csv("./BFI_soil_data/Soil_texture.csv", header = T, stringsAsFactors = F)
soil_qc <- read.csv("./BFI_soil_data/soil_qaqc_ku.csv", header=T, stringsAsFactors = F)

# rename team name column
names(plot_info)[33] <- "team"

# Create team data frame
team_name <- plot_info[,c("plot_id","team")]
names(team_name)[1] <- "plot_plot_id"

#create an output file to gather all erros in one file
output <- data.frame(step="Step",cat="Category",plot_id=9999,issue="Issue", "team" = 99, stringsAsFactors=F)

#####
#-----#
#CHECK : Soil bulk density
#CHECK 01: Check the compatibility of bulk density data by forest zone
# considering the soil condition of Bangladesh in order to identify
# the outliers
# [input: soil bulk density data; output: list of plots, subplot
# with strange value (value > 2.0 g/cc) of bulk density data]
#-----#

```

```
#####

#identify 5 plots having maximum bulk density
soil_bulk_work <- soil_bulk[which(soil_bulk$Plot_no!=791),] #data of plot_no 791 was deleted

#by the team by the soil data has been sent to the KU for analysis
soil_bulk_work[order(-soil_bulk$Bulk_density_g_cc),][1:15,]$Bulk_density_g_cc
names(soil_bulk_work)[1]<-"plot_plot_id"
soil_bulk_work<- merge(soil_bulk_work, plot_info [, c("plot_id", "location_zone")],
                      by.x ="plot_plot_id",
                      by.y= "plot_id",all.x=T)

#write the location zone of plot 891 as "villages" because it's data was not submitted yet
soil_bulk_work$location_zone <- ifelse(is.na(soil_bulk_work$location_zone), "Villages", soil_bulk_work$location_zone)

#plot soil bulk density by zone
ggplot(soil_bulk_work[which(!is.na(soil_bulk_work$location_zone)),], aes(x=location_zone, y=Bulk_density_g_cc)) +
  xlab("Forest zones") + ylab("Bulk density (g/cc)") +
  # geom_boxplot()+
  geom_bar(stat="summary", fun.y="mean", aes(fill="forest zones"))+
  ggtitle("Bulk density in different forest zones")+
  facet_wrap(~Depth)
#+geom_jitter()

#plot soil bulk density by zone highlighting the outliers
ggplot(soil_bulk_work[which(!is.na(soil_bulk_work$location_zone) & soil_bulk_work$Bulk_density_g_cc>2.65),], aes(x=location_zone, y=Bulk_density_g_cc)) +
  xlab("Forest zones") + ylab("Bulk density (g/cc)") +
  # geom_bar(stat="identity", aes(fill="forest zones"))+
  geom_boxplot()+
  ggtitle("Bulk density >2.65 g/cc in different forest zones")+
  facet_wrap(~Depth)
#+geom_jitter()

#plot soil bulk density by zone
ggplot(soil_bulk_work[which(!is.na(soil_bulk_work$location_zone)),], aes(x=location_zone, y=Bulk_density_g_cc)) +
  xlab("Forest zones") + ylab("Bulk density (g/cc)") +
  # geom_bar(stat="identity", aes(fill="forest zones"))+
  geom_boxplot()+
  ggtitle("Bulk density in different forest zones")+
  facet_wrap(~Depth)
#+geom_jitter()

#select plots having more than 2.65 g/cc bulk density
#As per soil manual soils have lower bulk density (0.1 - 0.6 g cm-3)
# than mineral (1.0 - 1.8 g cm-3); rocks upto (2.65 g/cc)
max_bulk_den <- soil_bulk_work[which(soil_bulk_work$Bulk_density_g_cc >=2.65),]
unique(max_bulk_den$plot_plot_id)

#select the plots with zero bulk density
min_bulk_den <- soil_bulk_work[which(soil_bulk_work$Bulk_density_g_cc==0),]

#combine the max and min bulk density by rbind
bulk_den_outlier <- rbind(max_bulk_den, min_bulk_den)
bulk_den_outlier <- merge(bulk_den_outlier, team_name, by ="plot_plot_id", all.x=TRUE)

#write.csv(bulk_den_outlier, file= "./QAQC_outputs/bulk_density_outliers_6.1.csv")

#include the plots with outlier in the output file
# Create table for output
bulk_den_outlier$issue <- paste("bulk density of", bulk_den_outlier$plot_plot_id,
                              bulk_den_outlier$Subplot_no, "and", bulk_den_outlier$Depth,
                              "is", bulk_den_outlier$Bulk_density_g_cc, sep=" ")
```

```

bulk_den_outlier$step <- "Step 1"
bulk_den_outlier$cat <- "bulk density outlier"
names(bulk_den_outlier)[1] <- "plot_id"

# Add issues to output
if(dim(bulk_den_outlier)[1] == 0){

  output1 <- rbind(output, data.frame(step="step 6.1",cat="bulk density",plot_id=9999,issue="Passed"

} else {

  output1 <- rbind(output, bulk_den_outlier[,c("step","cat","plot_id","issue", "team")])
}

#----- End of the CHECK 1 -----#

#####
#-----#
#CHECK 2: Check if the number of bulk density data for different soil
# depths/layer in different forest zones is appropriate or not as
# per BFI manual
# [input: bulk density data; output: the number of soil depth/layer
# not appropriate as per BFI manual]
#-----#
#####
soil_bulk_depths <- soil_bulk[which(soil_bulk$Plot_no!=791),] #plot 791 is excluded
#because data of this plot was not submitted but deleted by the field team
soil_bulk_depths$depth_count <- 1
soil_bulk_depths <- aggregate(depth_count~Plot_no+Subplot_no,
                             data=soil_bulk_depths, FUN=sum)

#merge the location zone
soil_bulk_depths<- merge(soil_bulk_depths, plot_info [, c("plot_id", "location_zone")],
                        by.x= "Plot_no",
                        by.y= "plot_id" ,all.x=T)

#write the location zone of plot 891 as "villages" because it's data was not submitted yet
soil_bulk_depths$location_zone <- ifelse(is.na(soil_bulk_depths$location_zone), "Villages", soil_bu

#plot/subplot in which bulk density data found for only one layer
a <- soil_bulk_depths[which(soil_bulk_depths$depth_count < 2),]
#a$issue <- paste("bulk density data in only 1 soil layer under subplot",
#               a$Subplot_no, "in", a$location_zone, "zone", sep=" ")

#plot/subplot in which bulk density data found for more than 3 depths
b <- soil_bulk_depths[which(soil_bulk_depths$depth_count > 3),]
#b$issue <- paste("bulk density data in only in 4 soil layers of subplot",
#               b$Subplot_no, "in", a$location_zone, "zone", sep=" ")

#if the bulk density data in less than 3 depths in plots/subplots of
# sundarbans and coastal zones
c <- soil_bulk_depths[which((soil_bulk_depths$location_zone== "Sundarbans" |
                             soil_bulk_depths$location_zone=="Coastal") &
                             soil_bulk_depths$depth_count < 3),]
#c$issue <- paste("bulk density data in 2 soil layers under subplot",
#               b$Subplot_no, "in", c$location_zone, "zone", sep=" ")

#if the bulk density data is in more than 2 depths in plots/subplots of
# forest zones other than sundarbans and coastal
d <- soil_bulk_depths[which(soil_bulk_depths$Plot_no!=1219 &
                             (soil_bulk_depths$location_zone!= "Sundarbans" &
                              soil_bulk_depths$location_zone!="Coastal") &
                             soil_bulk_depths$depth_count >2),] #since plot 1219 is already recorded

```

```

#d$issue <- paste("bulk density data in 3 soil layers of subplot",
#               b$Subplot_no, "in", d$location_zone, "zone", sep=" ")

#prepare a data frame for all soil layer inconsistency
soil_bulk_density_incon <- rbind(a,b,c,d)

#add the issue
soil_bulk_density_incon$issue <- ifelse((soil_bulk_density_incon$location_zone!= "Sundarbans" &
soil_bulk_density_incon$location_zone!="Coastal") &
soil_bulk_density_incon$depth_count >2,
paste("bulk density data in 3 soil layers of subplot",
soil_bulk_density_incon$Subplot_no,
"in",
soil_bulk_density_incon$location_zone,
"zone", sep=" "),
ifelse((soil_bulk_density_incon$location_zone== "Sundarbans" &
soil_bulk_density_incon$location_zone=="Coastal") &
soil_bulk_density_incon$depth_count < 3,
paste("bulk density data in 2 soil layers under subplot",
soil_bulk_density_incon$Subplot_no, "in",
soil_bulk_density_incon$location_zone, "zone", sep=" "),
ifelse(soil_bulk_density_incon$depth_count>3,
paste("bulk density data in only in 4 soil layers of subplot",
soil_bulk_density_incon$Subplot_no, "in",
soil_bulk_density_incon$location_zone, "zone", sep=" "),
paste("bulk density data in only 1 soil layer under subplot",
soil_bulk_density_incon$Subplot_no, "in", soil_bulk_density_incon$location_zone, "zone", sep=" ")))

#export the results
soil_bulk_density_incon <- merge(soil_bulk_density_incon, team_name,
by.x="Plot_no",
by.y="plot_plot_id",
all.x=TRUE)

soil_bulk_density_incon$depth_count_label <- paste(soil_bulk_density_incon$depth_count, "soil layers")

#write.csv(soil_bulk_density_incon, file= "./QAQC_outputs/bulk_density_inconsistency_layers6.2.csv")

#plot the data
soil_bulk_density_incon$count <- 1
ggplot(soil_bulk_density_incon, aes(x=location_zone, y=count))+
  geom_bar(stat="identity", aes(fill= location_zone))+
  xlab("Forest Zones")+ylab("Number of subplots")+
  facet_wrap(~depth_count_label)+
  ggtitle("Numebr of subplots with inconsistent soil layers based on zone")

#prepare the data to include in the output file
soil_bulk_density_incon$step <- "Step 2"
soil_bulk_density_incon$cat <- "bulk density sample depth number"
names(soil_bulk_density_incon)[1] <- "plot_id"

# Add issues to output
if(dim(soil_bulk_density_incon)[1] == 0){
  output2 <- rbind(output1, data.frame(step="step 2",cat="bulk density",plot_id=9999,issue="Passed"))
} else {
  output2 <- rbind(output1, soil_bulk_density_incon[,c("step","cat","plot_id","issue", "team")])
}
#-----End of the check 2 -----#

#####

```

```

#-----#
#CHECK 3: To check, in case of completely accessible or surveyed plots,
#         whether the number of bulk density data reported from 3 subplots
#         or not.
#         [input: bulk density data; output: number of completely sampled
#         plots with < 3 subplots from which soil bulk density was reported
#         and list of those plots]
#-----#
#####
soil_bulk_depths <- soil_bulk_depths[,c("Plot_no", "Subplot_no")]
soil_bulk_depths$subplot_count <- 1
soil_bulk_depths <- aggregate(subplot_count~Plot_no, data=soil_bulk_depths, FUN=sum)

#merge the locations
soil_bulk_depths<- merge(soil_bulk_depths, plot_info [, c("plot_id", "location_zone","plot_details_p
                    by.x ="Plot_no", by.y ="plot_id", all.x=T)

#if there is NA in location zone
soil_bulk_depths[which(is.na(soil_bulk_depths$location_zone)),]

#write the location zone of plot 891 as "villages" because it's data was not submitted yet
soil_bulk_depths$location_zone <- ifelse(is.na(soil_bulk_depths$location_zone), "Villages", soil_bu

#prepare the data.frame to plot the number of plots Vs number of subplot from which soil sample coll
soil_bulk_depths$plot_count <- 1

#plot the data
ggplot(soil_bulk_depths [which(soil_bulk_depths$plot_details_plot_status_label==
                            "Accessible - Sampled completely"),], aes(x=subplot_count, y=plot_c
  geom_bar(stat="identity", aes(fill=subplot_count))+
  xlab("Subplots identification number")+ ylab("Number of subplots")+
  ggtitle("Number of subplots from which soil sample collected for bulk density")

#select the plots in which soil sample was collected from less than
#two subplots
soil_bulk_subp_no <- soil_bulk_depths[which(soil_bulk_depths$subplot_count <3 &
                                           soil_bulk_depths$plot_details_plot_status_label=="Acc

#prepare the data to include in the output file
soil_bulk_subp_no$issue <- paste("plot sampled completely but soil bulk density data available for c
                            soil_bulk_subp_no$subplot_count, "subplot", sep=" ")
names(soil_bulk_subp_no) [1] <- "plot_id"
soil_bulk_subp_no$step <- "Step 3"
soil_bulk_subp_no$cat <- "bulk density subplot number"
soil_bulk_subp_no <- merge(soil_bulk_subp_no, team_name,
                          by.x ="plot_id", by.y="plot_plot_id", all.x=TRUE)
#write.csv(soil_bulk_subp_no, file = "./QAQC_outputs/bulk_density_data_from_2_subplots_but_plot_samp

# Add issues to output
if(dim(soil_bulk_subp_no)[1] == 0){
  output3 <- rbind(output2, data.frame(step="step 3",cat="bulk density",plot_id=9999,issue="Passed s
} else {
  output3 <- rbind(output2, soil_bulk_subp_no[,c("step","cat","plot_id","issue", "team")])
}
#-----End of the CHECK 3 -----#
#####
#CHECK 4: To see the soil organic carbon values, if they are within normal
#         range or not

```

```

# [input: Soil organic carbon data; output: outliers in organic
# carbon data and number of soil layers in which outliers (strange)
# data was found]
#-----#
#####
#soil carbon per plot and land feature
soil_car[order(-soil_car$Organic_carbon_perc),][1:20,]

#plot the carbon perc data by layer
ggplot(soil_car, aes(x=Soil_layer, y=Organic_carbon_perc))+
  geom_boxplot()+
  xlab("Soil layers (cm)")+ylab("Organic carbon (%)")

# Add a column with soil depth interval (cm)
soil_car$depth_interval_cm <- ifelse(soil_car$Soil_layer=="0-15", 15,
                                     ifelse(soil_car$Soil_layer=="15-30", 15, 70))

#summerize the data by interval depth
soil_car_sum <- aggregate(depth_interval_cm~plot_id+Land_feature_no, data=soil_car,
                          FUN=sum)

#prepare the result to incorporate in the output file
soil_car_sum$step <- "Step 4"
soil_car_sum$cat <- "strange soil layer"
soil_car_sum$issue <- "Strange number of soil layer"
soil_car_sum <- merge(soil_car_sum, team_name,
                      by.x="plot_id",
                      by.y = "plot_plot_id", all.x=TRUE)

#select the plots missing any layer
strnNumberLyer <- soil_car_sum[which(soil_car_sum$depth_interval_cm!=30 &
                                     soil_car_sum$depth_interval_cm!=100),]

#add soil layer and SOC %
strnNumberLyer <- merge(strnNumberLyer,
                       soil_car[,c("plot_id",
                                     "Soil_layer",
                                     "Organic_carbon_perc")],
                       by = "plot_id")

# add forest zones
strnNumberLyer <- merge(strnNumberLyer,
                       plot_info[,c("plot_id",
                                     "location_zone")],
                       by="plot_id", all.x=TRUE)

#export result
write.csv(strnNumberLyer, file = "../QAQC_outputs/C04strange_Number_soil_layer.csv")

#incorporate the results in the output

#seperate the plots with organic carbon greater than 5%
soil_car_prblm <- soil_car[soil_car$Organic_carbon_perc>=5.0,]
soil_car_prblm$layer_count <- 1
soil_car_prblm <- merge(soil_car_prblm, plot_info[,c("plot_id", "location_zone")],
                       by.x="plot_id", by.y="plot_id",
                       all.x=TRUE)

#show the problems in graph
ggplot(soil_car_prblm, aes(x=Soil_layer, y=layer_count))+
  geom_bar(stat="identity", aes(fill=Soil_layer))+
  xlab("Soil layers (cm)")+ylab("Numer of layers in which OC >= 5%")+
  ggtitle("Number of soil layers in which soil organic carbon (%) >= 5%")+
  facet_wrap(~location_zone)

```





```

all.x=TRUE)

soil_plot_lf_nmt[3:4]<- NULL
names(soil_plot_lf_nmt)[4] <- "lf_id_soil_bd"
soil_plot_lf_nmt <- unique(soil_plot_lf_nmt)
soil_plot_lf_nmt$cat <- "mismatch in soil C and BD"
soil_plot_lf_nmt$step <- "Step 5"
soil_plot_lf_nmt$issue <- paste("soil carbon data available for LF",
                                soil_plot_lf_nmt$lf_id_soil_car,
                                "but BD is available for LF",
                                soil_plot_lf_nmt$lf_id_soil_bd)

soil_plot_lf_nmt <- merge(soil_plot_lf_nmt, team_name,
                          by.x = "plot_id",
                          by.y= "plot_plot_id", all.x=TRUE)

#export the results
#write.csv(soil_plot_lf_nmt, file = "./QAQC_outputs/C05_mismatch_soil_BDC.csv")

#incldue the result in the output table
if(dim(soil_plot_lf_nmt)[1]==0){
  output5 <- rbind(output4, data.frame(step="Step 5", cat= "mismatch in soil C and BD", plot_id=999))
} else {
  output5 <- rbind(output4, soil_plot_lf_nmt[,c("step", "cat", "plot_id", "issue", "team")])
}

#####
#-----#
#CHECK 6: To see the relationship between bulk density and soil carbon and
#         evaluate visually if there is any abnormal relationship
#         [input: Soil organic carbon data, output: graphs indicating the
#         relationship between soil organic carbon and bulk density]
#-----#
#####

#merge forest zone with soil car data table
soil_car <- merge(soil_car, plot_info[,c("plot_id", "location_zone")],
                  by.x="plot_id",
                  by.y="plot_id", all.x=TRUE)

#merge team number
soil_car<- merge(soil_car, team_name, by.x="plot_id", by.y="plot_plot_id", all.x = TRUE)

soil_car$car_layer_class_label <- ifelse(soil_car$car_layer_class== 1, "0-15",
                                         ifelse(soil_car$car_layer_class==2, "15-30", "30-100"))

#plot organic carbon perc Vs bulk density
ggplot(soil_car)+
  aes(y=Bulk_density_g_cc, x=Organic_carbon_perc)+
  geom_point()+
  ylab("Bulk density g/cc")+xlab("Soil organic carbon (%)")+
  facet_wrap(~car_layer_class_label)+
  # geom_smooth(method=lm)+
  ggtitle("Bulk density Vs Soil organic carbon")
# facet_wrap(~team)

#plot organic carbon perc Vs bulk density by different layers
ggplot(soil_car[which(soil_car$Bulk_density_g_cc <=2 & soil_car$Organic_carbon_perc <= 5),])+
  aes(y=Bulk_density_g_cc, x=Organic_carbon_perc)+
  geom_point()+
  ylab("Bulk density g/cc")+xlab("Soil organic carbon (%)")+
  facet_wrap(~car_layer_class_label)+
  geom_smooth(method=lm)+
  ggtitle("Bulk density (</=2 g/cc) Vs Soil organic carbon (</=5 perc)")

```

```

# facet_wrap(~team)
#-----End of Check 06-----#

#####
#-----#
#CHECK 07: To see if there is any outlier in soil carbon (t/ha) at
#         different forest zones
#         [Input: soil organic carbon data; output: soil organic carbon
#         in t/ha by soil depths and forest zones]
#-----#
#####

#calculate soil carbon t/ha
#soil organic carbon (t/ha) = Bulk density (g/cm3) × OC (%) × Soil depth interval (cm) × 100
#Here, soil depth interval is 15 cm for 0-15 cm depth interval, 15 cm for
# soil depth 15-30 cm depth interval, and 70 cm for 30-100 cm interval. %OC
# is expressed as a decimal fraction (e.g., 5% is expressed as 0.05) and
# 100 is a conversion factor to convert the units to t/ha.
# The calculation method followed the protocol of Donato et al. (2009) .

#select the data exculding plot 791 data of which was deleted by the field team
soil_car_cal <- soil_car[which(soil_car$plot_id!=791),]

#Write the location_zone of plot 891; since the tree data was not submitted yet by
# the field team
soil_car_cal$location_zone <- ifelse(is.na(soil_car_cal$location_zone==891), "Villages",
                                   soil_car_cal$location_zone)

# Add a column with soil depth interval (cm)
soil_car_cal$soil_depth_interval_cm <- ifelse(soil_car_cal$car_layer_class==1, 15,
                                              ifelse(soil_car_cal$car_layer_class==2, 15, 70))

#convert the OC% to decimal
soil_car_cal$organic_car_decimal <- soil_car_cal$Organic_carbon_perc/100

#calculate carbon (t/ha)
soil_car_cal$soil_car_t_ha <- soil_car_cal$Bulk_density_g_cc*
  (soil_car_cal$Organic_carbon_perc/100)*
  soil_car_cal$soil_depth_interval_cm*100

#plot soil carbon Vs zone
ggplot(soil_car_cal)+
  aes(x=location_zone, y=soil_car_t_ha)+
  geom_boxplot()+
  xlab("Forest zones")+ylab("Soil organic carbon (t/ha)")+
  facet_wrap(~car_layer_class_label)+
  theme(axis.text.x = element_text(angle=90, hjust=1))+
  # coord_flip()+
  ggtitle("Soil organic carbon (t/ha) Vs Forest Zones")

#the very high value of soil carbon (t/ha) in hill forest was due to the
# strange value of bulk density
soil_car_cal[which(soil_car_cal$soil_car_t_ha >500),]

#plot soil carbon Vs zone excluding the outlier (OC more than 500 t/ha)
ggplot(soil_car_cal[which(soil_car_cal$soil_car_t_ha <500),])+
  aes(x=location_zone, y=soil_car_t_ha)+
  geom_boxplot()+
  xlab("Forest zones")+ylab("Soil organic carbon (t/ha)")+
  facet_wrap(~car_layer_class_label)+
  theme(axis.text.x = element_text(angle=90, hjust=1))+
  # coord_flip()+
  ggtitle("Soil organic carbon (t/ha) Vs Forest Zones")

```

```

#summerize the soil organic carbon (t/ha)
soil_car_cal <- aggregate(cbind(soil_car_t_ha,soil_depth_interval_cm)~
                        plot_id+Land_feature_no,
                        data=soil_car_cal, FUN=sum)

#prepare the table to identify the outliers and include that in the ouput table
soil_car_cal$issue <- paste("soil orgaic carbon is very high (", soil_car_cal$soil_car_t_ha, "t/ha)
soil_car_cal$step="Step 7"
soil_car_cal$cat="soil carbon outlier"
soil_car_cal <- merge(soil_car_cal, team_name,
                    by.x="plot_id",
                    by.y="plot_plot_id", all.x=TRUE)

soil_car_strange <- soil_car_cal[which(soil_car_cal$soil_car_t_ha >500),]

#incldue the result in the output table
if(dim(soil_car_strange)[1]==0){
  output7 <- rbind(output5, data.frame(step="Step 7", cat= "soil carbon outlier", plot_id=9999, team
} else {
  output7 <- rbind(output5, soil_car_strange[,c("step", "cat", "plot_id", "issue", "team")])
}

#-----End of Check 7-----#

#####
#-----#
#CHECK 8: To identify the plots in different zone from which soil carbon
# data is reported other than the specified soil layers as per
# BFI manual
# [Input: soil organic carbon, output: list of plots where soil
# organic carbon is reported from soil layers other than specified
# by BFI manual]
#-----#
#####

#the figure in the previous checks shows carbon in village from in 30-100 soil layer. But, as per BFI
# data was not supposed to collect from that layer of village forest zone.
# list the plot with details in which case soil organic carbon percentage is
# reported from 30-100 cm soil layer
soil_car_vill_prblm <- soil_car_cal[which(soil_car_cal$car_layer_class==3 & soil_car_cal$location_zone=="Village Forest Zone"),]

#write.csv(soil_car_vill_prblm, file = "../QAQC_outputs/soil_carbon_in_30-100_cm_layer_of_village_zone.csv")

#incorporate the error in output table
soil_car_vill_prblm$step <- "Step 8"
soil_car_vill_prblm$cat <- "organic carbon soil layer"
soil_car_vill_prblm$issue <- paste("Organic carbon is reported from", soil_car_vill_prblm$car_layer_class)

if(dim(soil_car_vill_prblm)[1]==0){
  output8 <- rbind(output7, data.frame(step="Step 8", cat="Organic Carbon", plot_id=9999, issue="Organic Carbon Reported from Village Forest Zone", team="BFI"))
} else {
  output8 <- rbind(output7, soil_car_vill_prblm[,c("step", "cat", "plot_id", "issue", "team")])
}

#-----End of Check 08-----#

#####
#-----#
#CHECK 9: To identify the plots and subplots in which the soil
# sample is collected from layers other than specified by BFI manual
# in different forest zone [2 defined layers in hill, village and
# Sal forests are 0-15cm and 15-30 cm; 3 specified layers in

```

```

# Sundarbans and coastal forests are 0-15 cm, 15-30cm and 30-100 cm]
# [Input: Soil texture data; output: graph presenting soil texture
# sample collection outside the specified soil depths and number of
# subplots and plots where sample was collected outside the
# specified soil layers/depths as per BFI manual]
#-----#
#####

head(soil_tex)
soil_tex_work <- soil_tex[which(soil_tex$plot_id!=791),]

soil_tex_work <- merge(soil_tex_work, plot_info[,c("plot_id", "location_zone")],
                      by.x="plot_id", by.y= "plot_id", all.x=TRUE)

#write the location of the 891 plot as "Villages"
soil_tex_work$location_zone <- ifelse(is.na(soil_tex_work$location_zone), "Villages", soil_tex_work$location_zone)

#plot sand percentag Vs soil depth
ggplot(soil_tex_work, aes(x=Layer, y=Sand_perc))+
  geom_boxplot()+
  xlab("Depth of soil layer")+ ylab("Sand particle percentage")+
  # facet_wrap(~team)
  facet_wrap(~location_zone)+
  ggtitle("Sandy particle percentage in different soil layers")

#plot silt percentage Vs soil depth
ggplot(soil_tex_work, aes(x=Layer, y=Clay_perc))+
  geom_boxplot()+
  xlab("Depth of soil layer")+ ylab("Silt particle percentage")+
  # facet_wrap(~team)
  facet_wrap(~location_zone)+
  ggtitle("Silty particle percentage in different soil layers")

#plot clay percentage Vs soil depth
ggplot(soil_tex_work, aes(x=Layer, y=Clay_perc))+
  geom_boxplot()+
  xlab("Depth of soil layer")+ ylab("Clay particle percentage")+
  # facet_wrap(~team)
  facet_wrap(~location_zone)+
  ggtitle("Clayey particle percentage in different soil layers")

#the figure showed that in village zone soil sample is collected from 30-100cm.
# But it is not suggested in the BFI manual. Let's identify the soil layers
# along with subplots and plots for which soil texture data is reported
tex_outlier <- soil_tex_work[which(soil_tex_work$Layer=="30-100" &
                                soil_tex_work$location_zone=="Villages"),]

#Export the list of plots, subplots for which soil texture data was reported
# from 30-100cm depth in Village zone

#write.csv(tex_outlier, file="./QAQC_outputs/texture_outlier_in_terms_of_soil_depths.csv")

#incorporate the error in output table
tex_outlier$step <- "Step 9"
tex_outlier$cat <- "Soil texture data and soil layer"
tex_outlier$issue <- paste("Soil texture data is reported from", tex_outlier$Layer,"cm layer of village zone")
tex_outlier <- merge(tex_outlier, team_name,
                    by.x= "plot_id",
                    by.y="plot_id",
                    all.x=TRUE)

if(dim(tex_outlier)[1]==0){
  output9 <- rbind(output8, data.frame(step="Step 9", cat="Soil texture layer", plot_id=9999, issue=""))
} else {
  output9 <- rbind(output8, tex_outlier[,c("step", "cat", "plot_id", "issue", "team")])
}

```

```
#-----End of Check 09-----#
```

```
#####
#-----#
#CHECK 10: To identify the layers if it is not matched with BFI soil survey
#          design or if any unknown layers is written in the data sheet
#          [Input: Soil texture data; Output: list 8of plots in which soil is
#          collected from undefined soil layers]
#-----#
#####
```

```
#the figure from check showed two more soil layers (15-13cm and 300-100 cm)
# may be these are written by mistake
```

```
#identify the plots with 15-13 cma and 300-100 layer [may be written by mistake]
```

```
#incorporate the error in output table
```

```
soil_tex_work$step <- "Step 10"
soil_tex_work$cat <- "organic carbon soil layer (unknown)"
soil_tex_work$issue <- paste("Organic carbon is reported from", soil_tex_work$Layer, "cm layer", sep = ", ")
soil_tex_work <- merge(soil_tex_work, team_name, by.x="plot_id",
                      by.y="plot_plot_id", all.x=TRUE)
```

```
soil_tex_unk_layer <- soil_tex_work[which(soil_tex_work$Layer == "15-13" |
                                         soil_tex_work$Layer == "300-100"),]
```

```
#write.csv(soil_tex_unk_layer, file = "./QAQC_outputs/soil_texture_unknown_layer.csv")
```

```
if(dim(soil_tex_unk_layer)[1]==0){
  output10 <- rbind(output9, data.frame(step="Step 10", cat="Organic Carbon", plot_id=9999, issue="Organic Carbon"))
} else {
  output10 <- rbind(output9, soil_tex_unk_layer[,c("step", "cat", "plot_id", "issue", "team")])
}
#-----End of Check 10 -----#
```

```
#####
#-----#
#CHECK 11: To identify the outliers in percentages of clay, silt and sandy
#          particles in soil [i) if the sum of sand, silt and clay percentage
#          is 100 or not; ii) if there are any outliers/negative values.]
#          [Input: soil texture data; output: List of plots and layers where
#          the percentages of clay, silt and sand particles of same sample
#          is not 100]
#-----#
#####
```

```
head(soil_tex)
soil_tex_work <- soil_tex[which(soil_tex$plot_id!=791),]
soil_tex_work$sand_clay_silt_perc <- round((soil_tex_work$Sand_perc +
                                           soil_tex_work$Clay_perc + soil_tex_work$Silt_perc, 1)
```

```
#see the order of values in sand, silt and clay
```

```
soil_tex_work[order(soil_tex_work$Sand_perc, decreasing=FALSE),][1:10,]
soil_tex_work[order(soil_tex_work$Silt_perc, decreasing=FALSE),][1:10,]
soil_tex_work[order(soil_tex_work$Clay_perc, decreasing=FALSE),][1:10,]
```

```
# Identify the negative and values and sum not being 100
```

```
#incorporate the error in output table
```

```
soil_tex_work$step <- "Step 11"
soil_tex_work$cat <- "soil textrue values"
soil_tex_work$issue <- paste("San_perc values is negative (", soil_tex_work$Sand_perc, ")", sep = ", ")
soil_tex_work <- merge(soil_tex_work, team_name,
                      by.x="plot_id", by.y = "plot_plot_id",
                      all.x=TRUE)
```

```

#identify the plots in which sum of sand, silt and cla percentage is not 100
soil_tex_sum_prblm <- soil_tex_work[which(soil_tex_work$sand_clay_silt_perc!=100),]

#identify the pltos in which the sand, silt and clay percentage is negative
soil_tex_value_prblm <- soil_tex_work[which(soil_tex_work$Sand_perc <0 |soil_tex_work$Silt_perc <0
      soil_tex_work$Clay_perc <0),]

#merge the problems (i.sum of sand, silt and clay percentage and ii. negative
# values in sand texture percentage
soil_tex_prblm <- rbind(soil_tex_value_prblm, soil_tex_sum_prblm)

#export the result
#write.csv(soil_tex_prblm, file = "./QAQC_outputs/Negative_values_in_soil_texture.csv")

if(dim(soil_tex_prblm)[1]==0){
  output11 <- rbind(output10, data.frame(step="Step 11", cat="soil texture", plot_id=9999, issue="Pa
} else {
  output11 <- rbind(output10, soil_tex_prblm[,c("step", "cat", "plot_id", "issue", "team")])
}
#-----End of the check 11-----#

#####
#-----#
#CHECK 12: To see if there is any outlier in the litter carbon (%)
# [Input: Litter carbon data; output: graph (boxplot) presenting
# outliers in the litter carbon data]
#-----#
#####
head(lit_car)
lit_car_work <- lit_car
lit_car_work <- merge(lit_car_work, team_name, by.x = "plot_id",
  by.y = "plot_plot_id", all.x = TRUE)
lit_car_work <- merge(lit_car_work, plot_info[,c("plot_id", "location_zone")],
  by.x="plot_id", by.y= "plot_id", all.x=TRUE)

#merge land nlcl
lit_car_work <- merge(lit_car_work, lf[,c("plot_plot_id", "lf_id", "nlcl")],
  by.x=c("plot_id", "Land_feature_no"),
  by.y=c("plot_plot_id", "lf_id"))

#plot litter carbon data
ggplot(lit_car_work, aes(y=Organic_carbon_in_litter_perc, x=nlcl))+
  geom_boxplot()+
  xlab("Land class legends")+ylab("Organic carbon in litter (%)")+
  ggtitle("Leaf carbon (%) in different land classes")

#identify the outliers in litter carbon content (carbon more than 55%)
#according to Burghouts (1992) average carbon content in litter is 46.90 %
#in primary forest of Danum Valley
#according to Pereira Júnior (2016) litter carbon is 42.76 ± 0.77%

lit_car_work[order(-lit_car_work$Organic_carbon_in_litter_perc),][1:5,]

lit_car_max <- lit_car_work[which(lit_car_work$Organic_carbon_in_litter_perc>55),]

write.csv(lit_car_max, file = "./QAQC_outputs/maximum_litter_carbon_values.csv")

```

```

#prepare the outliers table to include in the output table
lit_car_max$step <- "Step 12"
lit_car_max$cat <- "litter carbon outliers"
lit_car_max$issue <- paste("litter carbon is",
                          lit_car_max$Organic_carbon_in_litter_perc,
                          ">55%")

#incorporate the findings in the output file
if(dim(lit_car_max)[1]==0){
  output12 <- rbind(output11, data.frame(step="Step 12", cat="litter carbon outliers", plot_id=9999))
} else {
  output12 <- rbind(output11, lit_car_max[,c("step", "cat", "plot_id", "issue", "team")])
}

#####
#-----#
#CHECK 13: To see the relationship in order to check the consistency between
# canopy coverage and leaf oven dry weight
# [Input: Litter dry weight data; output: Box plot with outliers in
# litter dry weight data based on land class legends and list of
# plots with strange values in litter weight)
#-----#
#####

head(lit_dry_wt)
lit_dry_wt2 <- lit_dry_wt[which(lit_dry_wt$plot_id!=891),]
#lit_dry_wt2 <- lit_dry_wt[lit_dry_wt$subplot_id %in% c(1:5),]
unique(lit_dry_wt2$subplot_id)

#aggregate the litter oven dry wt by plot and land feature
lit_dry_wt2 <- aggregate(Oven_dry_litter_g~plot_id+Land_feature_no,
                        data=lit_dry_wt2, FUN = sum)

#add nlcl
lit_dry_wt2 <- merge(lit_dry_wt2,lf[,c("plot_plot_id", "lf_id", "nlcl")],
                    by.x=c("plot_id", "Land_feature_no"),
                    by.y=c("plot_plot_id", "lf_id"),
                    all.x=TRUE)

#plot litter carbon data
ggplot(lit_dry_wt2, aes(y=Oven_dry_litter_g, x=nlcl))+
  geom_boxplot()+
  xlab("Land class legends")+ylab("leaf oven dry weighth (gm)")+
  ggtitle("Leaf dry weight (gm) in different land classes")

#Add crown cover data to see if leaf carbon varies with canopy covers of
# individual plots
#Crown cover per plot and land feature (max is from lf_info)
# Leaf Cover per plot: Calculate count of subplots and multiply by leaf cover to get total leaf cover
subplot_count <- subplot_info[,c("plot_plot_id","subp_id", "subp_details_subp_leaf_pc")]
subplot_count <- unique(subplot_count)
subplot_count$count <- 1

subplot_count1 <- aggregate(count~plot_plot_id, data = subplot_count, FUN = sum)
subplot_count2 <- aggregate(subp_details_subp_leaf_pc~plot_plot_id, data = subplot_count, FUN = sum)
subplot_count <- merge(subplot_count1, subplot_count2,
                      by = "plot_plot_id", all.x=TRUE)
subplot_count$leaf_cover_plot <- (subplot_count$subp_details_subp_leaf_pc/subplot_count$count)

# Crown cover max per plot (max is from lf_info)
subplot_cc <- lf[,c("plot_plot_id", "lf_details_cover_min", "lf_details_cover_max"),]
subplot_cc_min <- aggregate(lf_details_cover_min~plot_plot_id, data = subplot_cc, FUN = max)

```

```

subplot_cc_max <- aggregate(lf_details_cover_max~plot_plot_id, data = subplot_cc, FUN = max)
subplot_cc <- merge(subplot_cc_min, subplot_cc_max, by = "plot_plot_id")

# Merge both tables together
cover <- merge(subplot_cc, subplot_count[,c("plot_plot_id", "leaf_cover_plot")], by = "plot_plot_id")
cover$cc_average <- (cover$lf_details_cover_min + cover$lf_details_cover_max)/2

#merge canopy cover and leaf cover
lit_dry_wt2 <- merge(lit_dry_wt2, cover[,c("plot_plot_id", "leaf_cover_plot", "cc_average")],
                    by.x= "plot_id", by.y = "plot_plot_id", all.x = TRUE)

#plot litter wt and canopy coverage
ggplot(lit_dry_wt2, aes(x=cc_average, y=Oven_dry_litter_g))+
  geom_point()+
  xlab("Canopy coverage")+ylab("Oven dry weight of litter (g)")+
  facet_wrap(~nlcl)

#plot litter wt and leaf coverage
ggplot(lit_dry_wt2, aes(x=leaf_cover_plot, y=Oven_dry_litter_g))+
  geom_point()+
  xlab("leaf cover plot")+ylab("Oven dry weight of litter (g)")+
  facet_wrap(~nlcl)

#order the woven dry wt
lit_dry_wt2[order(lit_dry_wt2$Oven_dry_litter_g, decreasing = TRUE),][1:10,]
str(lit_dry_wt2)

#Identify the plots with problematic leaf dry weight
#prepare the table for exporting the problematic plots into output table
#list the plot in output file
lit_dry_wt_prob <- lit_dry_wt2
lit_dry_wt_prob$step <- "Step 13"
lit_dry_wt_prob$cat <- "litter dry wt and canopy cover"
lit_dry_wt_prob$issue <- "litter dry wt is more than 400 g but leaf cover and canopy cover is <=50%"
lit_dry_wt_prob <- merge(lit_dry_wt_prob, team_name,
                        by.x = "plot_id",
                        by.y = "plot_plot_id",
                        all.x=TRUE)

#select plots in which litter oven dry wt is > 400 but
# leaf cover and canopy cover is less than 50%

lit_dry_wt_prob <- lit_dry_wt_prob[which((lit_dry_wt_prob$Oven_dry_litter_g >= 400 |
                                         lit_dry_wt_prob$Oven_dry_litter_g == 0) &
                                         lit_dry_wt_prob$leaf_cover_plot <= 0.50 &
                                         lit_dry_wt_prob$cc_average <= 50.0),]

#write.csv(lit_dry_wt_prob, file="./QAQC_outputs/outliers_in_litter_dry_wt.csv")

if(dim(lit_dry_wt_prob)[1]==0){
  output13 <- rbind(output12, data.frame(step="Step 13", cat="litter dry wt and canopy cover", plot.
} else {
  output13<- rbind(output12, lit_dry_wt_prob[,c("step", "cat", "plot_id", "issue", "team")])
}

#####
#-----#
#CHECK 14: To see if there is any subplots having more than one litter dry wt
# data
# [Input: Litter dry weight data;
# output: list of plots and subplots

```



```

#           having more than one littre dry wt data]
#-----#
#####

#summeris litter dry wt data by subplot
#if any subplot have more than one litter dry wt data
names(lit_dry_wt2)
lit_dry_wt2$subp_count <- 1
lit_wt_subp_no <- aggregate(subp_count~plot_id, data=lit_dry_wt2, FUN=sum)

lit_wt_subp_no[order(-lit_wt_subp_no$subp_count, decreasing=TRUE),][1:10,]

#list the plot in output file
lit_wt_subp_no_prob <- lit_wt_subp_no
lit_wt_subp_no_prob$step <- "Step 14"
lit_wt_subp_no_prob$cat <- "litter dry wt and canopy cover"
lit_wt_subp_no_prob$issue <- paste("Litter dry wt data is reported for",
                                   lit_wt_subp_no_prob$subp_count,
                                   "subplots of", lit_wt_subp_no_prob$plot_id,
                                   "plot", sep=" ")
lit_wt_subp_no_prob <- merge(lit_wt_subp_no_prob, team_name,
                             by.x="plot_id",
                             by.y="plot_plot_id",
                             all.x=TRUE)
lit_wt_subp_no_prob <-lit_wt_subp_no[which(lit_wt_subp_no$subp_count>3),]

#Add to ouput file
if(dim(lit_wt_subp_no_prob)[1]==0){
  output14 <- rbind(output13, data.frame(step="Step 14", cat="litter dry wt data from subplot number
} else {
  output14<- rbind(output13, lit_wt_subp_no_prob[,c("step", "cat", "plot_id", "issue", "team")])
}

#-----End of CHECK 13-----#

#####
#-----#
#CHECK 15: Identify the plots with mismatched land feature id in litter carbon and
#           litter dry weight data
#           [Input: lit_car.csv and lit_dry_wt.csv, lf.csv
#           Output: list of plots with mismatches with lf.csv by land feature id]
#-----#
#####
head(lit_car)
a1 <- lit_car[,c("plot_id", "Land_feature_no")]

#add nlcl
a1 <- merge(a1, lf[,c("plot_plot_id", "lf_id", "nlcl")],
            by.x=c("plot_id", "Land_feature_no"),
            by.y=c("plot_plot_id", "lf_id"), all.x=TRUE)

#select necessary columns from lit dry wt
a2 <- lit_dry_wt2[,c("plot_id", "Land_feature_no", "nlcl")]

#select plot, lf and nlcl columns from lf data base
lf_test <- lf[,c("plot_plot_id", "lf_id", "nlcl")]

#find mismatches in the lit_dry_wt2 and lit_car_work
#install.packages("sqldf")
require(sqldf)
a1NotIna2 <- sqldf('SELECT * FROM a1 EXCEPT SELECT * FROM a2')
a1NotInlf_test <- sqldf('SELECT * FROM a1 EXCEPT SELECT * FROM lf_test')
a2NotInlf_test <- sqldf('SELECT * FROM a2 EXCEPT SELECT * FROM lf_test')

```

#All lf\_id are present in lf.csv table