Carbon Inventory Code - Prepared in R Studio

Carbon Counters

3/18/2021

This code was used to prepare the full landscape carbon inventory, as well as the working lands inventories used in the baseline reference scenario code (see supplemental materials). For all data citations and detailed explanation of methods, please refer to the Methodology section of the report.

## Code setup

# Attach packages  
library(tidyverse)  
library(tidyr)  
library(here)  
library(janitor)  
library(plotly)  
library(kableExtra)  
library(effsize)  
library(broom)  
library(formattable)  
library(purrr)  
  
# read in data. file descriptions above each file.  
  
# carbon value table from CARB  
carbon\_vals <- read.csv(here::here("files", "luts", "lut\_lf\_carb.csv"), encoding = "UTF-8")  
  
# existing vegetation cover (LANDFIRE) look up table  
evc\_lut <- read.csv(here::here("files", "luts", "lut\_evc.csv"), encoding = "UTF-8") %>%  
 clean\_names() %>%  
 rename(classnames\_evc = x\_u\_feff\_classnames)  
  
# existing vegetation height (LANDFIRE) look up table  
evh\_lut <- read.csv(here::here("files", "luts", "lut\_evh.csv"), encoding = "UTF-8") %>%   
 clean\_names() %>%   
 rename(classnames\_evh = x\_u\_feff\_classnames)  
  
# existing vegetation type (LANDFIRE) look up table  
evt\_lut <- read.csv(here::here("files", "luts", "lut\_evt.csv"), encoding = "UTF-8") %>%   
 clean\_names() %>%   
 rename(classnames\_evt = x\_u\_feff\_evt\_name)  
  
# nitrogen look up table (2007 SB County GHG inventory)  
lut\_n <- read.csv(here::here("files", "luts", "lut\_n.csv"), encoding = "UTF-8") %>%   
 clean\_names() %>%   
 rename(nitrogen\_cat = x\_u\_feff\_nitrogen)  
  
# 2016 evc/evh/evt for 2016, assigned to points for each 30x30m pixel in the county (our pre-processed data)  
lf\_evc\_16 <- read\_csv(here::here("files", "natlands", "LF\_2016\_EVC.csv"))  
  
lf\_evh\_16 <- read\_csv(here::here("files", "natlands", "LF\_2016\_EVH.csv"))  
  
lf\_evt\_16 <- read\_csv(here::here("files", "natlands", "LF\_2016\_EVT.csv"))  
  
# read in nitrogen categories to apply to LANDFIRE data where appropriate  
lf\_reclass\_n <- read.csv(here::here("files", "luts", "lf\_reclass\_nitrogen.csv"), encoding = "UTF-8") %>%   
 clean\_names() %>%   
 rename(classnames\_evt = x\_u\_feff\_evt) %>%   
 rename(lf\_n\_category = n\_category)  
  
# read in soil data - unit = gC / m^2  
  
soil <- read\_csv(here::here("files", "soil", "ssurgo.csv")) %>%   
 dplyr::select(pointid, soc0\_30)

## 2016 full landscape inventory

###############################################################  
# Natural lands  
###############################################################  
  
# Data cleaning and merging  
  
carbon\_vals <- carbon\_vals %>%   
 clean\_names("snake") %>%   
 filter(!is.na(w\_total\_mt\_cha)) %>%   
 rename(total\_mt = w\_total\_mt\_cha) %>%   
 rename(grouped = x\_u\_feff\_lf\_key)  
  
# combine all three (EVC, EVH, EVT) landfire dataframes for 2016, and clean  
precombin\_df <- merge(lf\_evc\_16, lf\_evh\_16, by = "OBJECTID")  
  
# combine all together, and create "grouped" variable to later match with carbon values  
combined\_lf\_df <- merge(precombin\_df, lf\_evt\_16, by = "OBJECTID") %>%   
 dplyr::select(OBJECTID, pointid, CLASSNAMES.x, CLASSNAMES.y, EVT\_NAME, Reclass\_16) %>%   
 clean\_names("snake") %>%   
 rename(classnames\_evc = classnames\_x) %>%   
 rename(classnames\_evh = classnames\_y) %>%  
 rename(classnames\_evt = evt\_name) %>%   
 left\_join(evh\_lut, by = "classnames\_evh") %>%   
 left\_join(evc\_lut, by = "classnames\_evc") %>%   
 left\_join(evt\_lut, by = "classnames\_evt") %>%   
 mutate(grouped = paste(evt\_group, evh\_group, evc\_group, sep = "")) %>%   
 left\_join(lf\_reclass\_n, by = "classnames\_evt") %>%   
 dplyr::select(-reclass\_16) %>%   
 rename(reclass\_16 = reclass\_category)  
  
###############################################################  
# Add agricultural lands  
###############################################################  
  
# read in ag data  
  
ag\_2016\_raw <- read.csv(here::here("files", "ag", "ag\_2016.csv"), encoding = "UTF-8", na.strings=c(""," ", "NoData", "NA"))  
  
# clean 2016 ag data and make appropriate changes to nitrogen values and classifications (some of these may be outdated/irrelevant, but this does not affect the results)  
# designate which points will rely on CalAg data, and which on LANDFIRE. This effectively overlays the CalAg data onto the LANDFIRE data  
  
ag\_2016 <- ag\_2016\_raw %>%   
 dplyr::select(!c(organic, crop\_list)) %>%   
 rename(nitrogen = nitrogren\_) %>%   
 clean\_names("snake") %>%   
 rename(pointid = objectid) %>%   
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(nitrogen = as.character(nitrogen)) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Barren / Fallow" | ag\_class == "Greenhouse", 0, nitrogen)) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Irrigated Pasture", "Fodder", as.character(ag\_class))) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Barren / Fallow", "Fallow", as.character(ag\_class))) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Pastureland", "Field Crops", nitrogen)) %>%   
 mutate(source = ifelse(is.na(ag\_class), "landfire", "calag"))  
  
# merge natural lands with ag, replace grouped name with ag classification where appropriate, clean data  
  
combined\_ag\_natland <- merge(combined\_lf\_df, ag\_2016, by = "pointid") %>%  
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(lf\_n\_category = as.character(lf\_n\_category)) %>%   
 mutate(grouped = ifelse(is.na(ag\_class), grouped, ag\_class)) %>%   
 mutate(reclass\_cat = ifelse(is.na(ag\_class), as.character(reclass\_16), as.character(ag\_class))) %>%  
 mutate(reclass\_cat = as.character(reclass\_cat)) %>%   
 mutate(reclass\_cat = ifelse(reclass\_cat == "Barren / Fallow", "Fallow", as.character(reclass\_cat))) %>%   
 dplyr::select(evt\_group, pointid, reclass\_cat, grouped, nitrogen, lf\_n\_category, source) %>%   
 rename(nitrogen\_cat = nitrogen) %>%   
 mutate(reclass\_cat = ifelse(reclass\_cat == "Wetland", "Riparian/Wetland", as.character(reclass\_cat))) %>%  
 mutate(reclass\_cat = ifelse(reclass\_cat == "Irrigated Pasture", "Fodder", as.character(reclass\_cat))) %>%  
 mutate(nitrogen\_cat = ifelse(source == "calag", as.character(nitrogen\_cat), as.character(lf\_n\_category)))  
  
# create simplified file to use in GIS mapping  
reclass\_map\_file <- combined\_ag\_natland %>%   
 dplyr::select(pointid, reclass\_cat)  
  
# calculate stored carbon and nitrous oxide emissions for each pixel (900 sq m)  
  
ag\_natland\_carbon\_n\_16 <- combined\_ag\_natland %>%   
 left\_join(carbon\_vals, by = "grouped") %>%  
 mutate(mt\_900 = (total\_mt\*.09)) %>% # MT carbon per hectare multiplied by .09 to get metric tons of carbon per pixel (900 sq m)  
 left\_join(lut\_n, by = "nitrogen\_cat") %>%   
 mutate(lbs\_n\_pixel = (n\_rate\_lbs\_acre\*.222395)) %>% # nitrogen application rate (pounds per acre) multiplied by .222395 to get pounds of N applied per per pixel  
 mutate(emit\_n\_lbs\_pix = (lbs\_n\_pixel \* .0175)) %>% # 1.75% of nitrogen escapes at N2O emissions  
 dplyr::select(!c(n\_rate\_lbs\_acre, lbs\_n\_pixel)) %>%   
 mutate(stock\_abvgc\_mtco2e\_pixel = (mt\_900\*1)) %>% # multiply metric tons of carbon by 3.67 to get MT of CO2 equivalent # Decided to report as MT instead, replace 3.67 value w 1 to not break rest of code  
 mutate(emit\_no\_mtco2e\_pix = emit\_n\_lbs\_pix\*298\*0.000453592) # multiply pounds to N2O emissions by 298 to convert to pounds CO2e, then by 0.000453592 to get metric tonnes  
  
# combine aboveground carbon and nitrogen data with soil data  
   
all\_c\_n\_soil <- merge(ag\_natland\_carbon\_n\_16, soil, by = "pointid") %>%   
 mutate(soil900 = (soc0\_30\*900)) %>% #per m^2 to per 900 m^2  
 mutate(soilMT = (soil900/1000000)) %>% # grams to metric tons of organic carbon  
 mutate(stock\_soilc\_mtco2e\_pix = soilMT\*1) # convert to CO2e # Decided to report as MT instead, replace 3.67 value w 1 to not break rest of code  
   
# make into a table, does not yet include urban forestry  
  
all\_clean\_16\_no\_tree <- all\_c\_n\_soil %>%   
 dplyr::select(pointid, reclass\_cat, stock\_abvgc\_mtco2e\_pixel, stock\_soilc\_mtco2e\_pix, emit\_no\_mtco2e\_pix, source) %>%   
 mutate(emit\_no\_mtco2e\_pix = replace\_na(emit\_no\_mtco2e\_pix, 0))  
  
# calculate acreages per landclass category  
  
all\_acreages\_16 <- all\_clean\_16\_no\_tree %>%   
 group\_by(reclass\_cat) %>%   
 summarize(pixels = n()) %>%   
 mutate(sqmeter = pixels\*900) %>%   
 mutate(acreage = sqmeter/4047) %>%   
 dplyr::select(! c(pixels, sqmeter)) %>%   
 adorn\_totals()  
  
#create a summary (preliminary inventory without urban forestry)  
ci\_summary\_cat\_16 <- all\_clean\_16\_no\_tree %>%  
 dplyr::select(!source) %>%   
 group\_by(reclass\_cat) %>%  
 summarise\_all(.funs = c(sum="sum"), na.rm = TRUE) %>%  
 mutate(net = (stock\_soilc\_mtco2e\_pix\_sum + stock\_abvgc\_mtco2e\_pixel\_sum)) %>%  
 merge(all\_acreages\_16, by = "reclass\_cat") %>%   
 dplyr::select(!pointid\_sum)  
  
###############################################################  
# Add in urban forestry  
###############################################################  
  
# CO2e stored/urban tree canopy (metric tons/acre) converted to MT C  
  
tree\_num <- 114.8730627/3.67  
  
# Assign # of acres of developed land as object for calculations  
# If making changes, double check this cell reference is correct (should reference "developed")  
  
urban\_acres <- all\_acreages\_16[2, 2]  
  
# create a row that matches ci\_summary\_cat\_16 (preliminary inventory) to add data with rbind  
   
tree\_row <-data.frame("Urban Forestry (Aboveground Only)", tree\_num\*urban\_acres, 0, 0, tree\_num\*urban\_acres, 0)  
names(tree\_row)<-c("reclass\_cat", "stock\_abvgc\_mtco2e\_pixel\_sum", "stock\_soilc\_mtco2e\_pix\_sum", "emit\_no\_mtco2e\_pix\_sum", "net", "acreage")  
  
# add urban forestry row to inventory  
  
ci\_summary\_cat\_16 <- rbind(ci\_summary\_cat\_16, tree\_row) %>%   
 mutate(reclass\_cat = ifelse(reclass\_cat == "Urban Forestry (Aboveground Only)", "Developed", reclass\_cat)) %>%   
 group\_by(reclass\_cat) %>%   
 summarise\_all(sum) %>%   
 adorn\_totals()  
  
# rename columns  
  
colnames(ci\_summary\_cat\_16) = c("Landcover Classification", "Total Aboveground Carbon (MT C)", "Total Soil Carbon (MT C)", "Total NO Emissions (MT CO2e)", "Total Stocks (MT C)", "Acres")  
  
# final landscape inventory  
ci\_summary\_cat\_16

## Landcover Classification Total Aboveground Carbon (MT C)  
## Barren 0.000  
## Developed 3372234.090  
## Fallow 2293.470  
## Fodder 12624.849  
## Forest 11932600.874  
## Grassland 676689.881  
## Greenhouse 0.000  
## Orchard 136676.009  
## Pastureland 3115.975  
## Riparian/Wetland 42546.252  
## Row Crop 62001.362  
## Shrubland 18591734.128  
## Vineyard 34147.413  
## Water 0.000  
## Total 34866664.304  
## Total Soil Carbon (MT C) Total NO Emissions (MT CO2e) Total Stocks (MT C)  
## 90077.412 0.0000 90077.412  
## 1356058.392 0.0000 4728292.483  
## 94663.749 0.0000 96957.219  
## 107472.848 396.2391 120097.697  
## 2747504.041 0.0000 14680104.915  
## 3543562.388 0.0000 4220252.268  
## 19398.360 0.0000 19398.360  
## 269416.907 3600.7312 406092.916  
## 48735.973 276.5252 51851.948  
## 111282.989 0.0000 153829.241  
## 1113708.673 32871.0189 1175710.036  
## 6228164.288 0.0000 24819898.416  
## 508851.928 1815.7271 542999.341  
## 8469.227 0.0000 8469.227  
## 16247367.175 38960.2416 51114031.479  
## Acres  
## 22207.784  
## 105836.175  
## 8027.502  
## 6844.403  
## 292859.155  
## 239987.991  
## 1937.213  
## 16878.280  
## 2541.216  
## 14932.394  
## 80597.480  
## 804518.162  
## 30613.121  
## 4381.468  
## 1632162.342

The next section essentially repeats the process above, using a loop, but excludes LANDFIRE data to create inventories from the CalAg data for 2012, 2016, and 2019. These will be used to create the baseline reference scenario

###############################################################  
# Process dataframes to use in baseline projections  
###############################################################  
  
# read in ag 2012 and 2019,and re-read in 2016 data  
  
ag\_2019\_raw <- read.csv(here::here("files", "ag", "ag\_2019.csv"), encoding = "UTF-8", na.strings=c(""," ", "NoData", "NA"))  
  
ag\_2019 <- ag\_2019\_raw %>%   
 dplyr::select(!c(organic, crop\_list)) %>%   
 rename(nitrogen = nitrogren\_) %>%   
 clean\_names("snake") %>%   
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(nitrogen = as.character(nitrogen)) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Barren / Fallow" | ag\_class == "Greenhouse", 0, nitrogen)) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Irrigated Pasture", "Fodder", as.character(ag\_class))) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Barren / Fallow", "Fallow", as.character(ag\_class))) %>%   
 rename(pointid = objectid) %>%   
 mutate(source = ifelse(is.na(ag\_class), "landfire", "calag"))  
  
ag\_2012\_raw <- read.csv(here::here("files", "ag", "ag\_2012.csv"), encoding = "UTF-8", na.strings=c(""," ", "NoData", "NA"))  
  
ag\_2012 <- ag\_2012\_raw %>%   
 dplyr::select(!c(organic, crop\_list)) %>%   
 rename(nitrogen = nitrogren\_) %>%   
 clean\_names("snake") %>%   
 rename(pointid = objectid) %>%   
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(nitrogen = as.character(nitrogen)) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Barren / Fallow" | ag\_class == "Greenhouse", 0, nitrogen)) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Irrigated Pasture", "Fodder", as.character(ag\_class))) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Barren / Fallow", "Fallow", as.character(ag\_class))) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Pastureland", "Field Crops", nitrogen)) %>%   
 mutate(source = ifelse(is.na(ag\_class), "landfire", "calag"))  
  
ag\_2016\_update <- ag\_2016\_raw %>%   
 dplyr::select(!c(organic, crop\_list)) %>%   
 rename(nitrogen = nitrogren\_) %>%   
 clean\_names("snake") %>%   
 rename(pointid = objectid) %>%   
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(nitrogen = as.character(nitrogen)) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Barren / Fallow" | ag\_class == "Greenhouse", 0, nitrogen)) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Irrigated Pasture", "Fodder", as.character(ag\_class))) %>%   
 mutate(ag\_class = ifelse(ag\_class == "Barren / Fallow", "Fallow", as.character(ag\_class))) %>%   
 mutate(nitrogen = ifelse(ag\_class == "Pastureland", "Field Crops", nitrogen)) %>%   
 mutate(source = ifelse(is.na(ag\_class), "landfire", "calag"))  
  
# put files into a list  
  
ag\_files\_list <- list(ag\_2012, ag\_2016\_update, ag\_2019)  
  
# first, merge with LANDFIRE data so as to not break any of the above code  
  
fx\_merge <- function(ag) {  
   
 merge(combined\_lf\_df, ag, by = "pointid") %>%  
 mutate(ag\_class = as.character(ag\_class)) %>%   
 mutate(grouped = ifelse(is.na(ag\_class), grouped, ag\_class)) %>%   
 mutate(reclass\_cat = ifelse(is.na(ag\_class), reclass\_16, ag\_class)) %>%   
 dplyr::select(evt\_group, pointid, reclass\_cat, grouped, nitrogen, source) %>%   
 rename(nitrogen\_cat = nitrogen) %>%   
 filter(source == "calag")  
   
}  
  
results <- lapply(ag\_files\_list, fx\_merge) %>%   
 setNames(c(2012, 2016, 2019))  
  
# make into data frames  
  
fx\_df <- function(result){  
 df\_name <- data.frame(result)  
}  
  
dfs <- lapply(results, fx\_df) %>%   
 setNames(c(2012, 2016, 2019))  
  
# calculate stored carbon and nitrous oxide emissions for each pixel (900 sq m)  
  
fx\_ghg\_calc <- function(dfs) {  
  
dfs %>%   
 left\_join(carbon\_vals, by = "grouped") %>%  
 mutate(mt\_900 = (total\_mt\*.09)) %>% # MT carbon per hectare multiplied by .09 to get metric tons of carbon per pixel (900 sq m)  
 left\_join(lut\_n, by = "nitrogen\_cat") %>%   
 mutate(lbs\_n\_pixel = (n\_rate\_lbs\_acre\*.222395)) %>% # nitrogen application rate (pounds per acre) multiplied by .222395 to get pounds of N applied per per pixel  
 mutate(emit\_n\_lbs\_pix = (lbs\_n\_pixel \* .0175)) %>% # 1% of nitrogen escapes at NO emissions  
 dplyr::select(!c(n\_rate\_lbs\_acre, lbs\_n\_pixel)) %>%   
 mutate(stock\_abvgc\_mtco2e\_pixel = (mt\_900\*1)) %>% # multiply metric tons of carbon by 3.67 to get MT of CO2 equivalent # Decided to report as MT instead, replace 3.67 value w 1 to not break rest of code  
 mutate(emit\_no\_mtco2e\_pix = emit\_n\_lbs\_pix\*298\*0.000453592) # multiply pounds to NO emissions by 298 to convert to pounds CO2e, then by 0.000453592 to get metric tonnes   
  
}  
  
ghgs <- lapply(dfs, fx\_ghg\_calc) %>%   
 setNames(c(2012, 2016, 2019))  
  
ghg\_dfs <- lapply(ghgs, fx\_df) %>%   
 setNames(c(2012, 2016, 2019))  
  
# add soil  
  
fx\_soil <- function(ghg\_dfs) {  
   
ghg\_dfs %>%   
 merge(soil, by = "pointid") %>%   
 mutate(soil900 = (soc0\_30\*900)) %>% #per m^2 to per 900 m^2  
 mutate(soilMT = (soil900/1000000)) %>% # grams to metric tons of organic carbon  
 mutate(stock\_soilc\_mtco2e\_pix = soilMT\*1) # convert to CO2e #Decided to report as MT instead, replace 3.67 value w 1 to not break rest of code  
}  
  
soil\_results <- lapply(ghg\_dfs, fx\_soil) %>%   
 setNames(c(2012, 2016, 2019))  
  
soil\_results\_dfs <- lapply(soil\_results, fx\_df) %>%   
 setNames(c(2012, 2016, 2019))  
  
# name these just to preview  
combined\_2012\_df <- soil\_results\_dfs[[1]]  
combined\_2016\_df <- soil\_results\_dfs[[2]]  
combined\_2019\_df <- soil\_results\_dfs[[3]]  
  
# clean up results and summarize values  
fx\_all\_clean <- function(df) {  
df %>%   
 dplyr::select(reclass\_cat, stock\_abvgc\_mtco2e\_pixel, stock\_soilc\_mtco2e\_pix, emit\_no\_mtco2e\_pix) %>%   
 group\_by(reclass\_cat) %>%   
 summarise\_all(.funs = c(sum="sum"), na.rm = TRUE) %>%   
 adorn\_totals()  
}  
  
summaries <- lapply(soil\_results\_dfs, fx\_all\_clean) %>%   
 setNames(c(2012, 2016, 2019))  
  
summaries\_dfs <- lapply(summaries, fx\_df) %>%   
 setNames(c(2012, 2016, 2019))  
  
names <- names(summaries\_dfs)  
  
res\_list <- vector("list", length = length(names)) %>%   
 setNames(names)  
  
# lastly, make some tables  
  
for(i in names){  
  
loop\_total\_table <- soil\_results\_dfs[[i]] %>%   
 group\_by(reclass\_cat) %>%   
 summarize(pixels = n()) %>%   
 mutate(sqmeter = pixels\*900) %>%   
 mutate(acreage = sqmeter/4047) %>%   
 merge(summaries\_dfs[[i]], by = "reclass\_cat") %>%  
 filter(reclass\_cat %in% c("Fallow", "Fodder", "Orchard", "Row Crop", "Pastureland", "Vineyard", "Greenhouse")) %>%   
 mutate(net = (stock\_soilc\_mtco2e\_pix\_sum + stock\_abvgc\_mtco2e\_pixel\_sum)) %>%   
 adorn\_totals() %>%   
 mutate(year = i)  
   
res\_list[[i]] <- loop\_total\_table  
   
 }  
  
total\_tables <- lapply(res\_list, fx\_df) %>%   
 setNames(c(2012, 2016, 2019))  
  
# and name them as objects  
  
ag\_acreage\_12 <- total\_tables[[1]]  
ag\_acreage\_16 <- total\_tables[[2]]  
ag\_acreage\_19 <- total\_tables[[3]]