Baseline Reference Scenario - Prepared in R Studio

Carbon Counters

3/18/2021

## Code Setup

# Attach packages  
library(tidyverse)  
library(tidyr)  
library(here)  
library(janitor)  
library(plotly)  
library(kableExtra)  
library(effsize)  
library(stargazer)  
library(broom)  
library(plotly)  
  
# Read in data files and clean up, created in carbon inventory script. These files contain # of acres of each ag class according to calag (excludes landfire).   
  
ag\_12 <- read\_csv(here::here("results", "ag\_final\_12.csv")) %>%   
 dplyr::select(! c(pixels, sqmeter))  
ag\_16 <- read\_csv(here::here("results", "ag\_final\_16.csv")) %>%   
 dplyr::select(! c(pixels, sqmeter))  
ag\_19 <- read\_csv(here::here("results", "ag\_final\_19.csv")) %>%   
 dplyr::select(! c(pixels, sqmeter))  
  
# change column names  
colnames <- c("class", "acres","abvgc", "soilc", "noemit", "net", "year")  
colnames(ag\_12) = colnames  
colnames(ag\_16) = colnames  
colnames(ag\_19) = colnames  
  
#Add in rangeland values from cal ag  
  
# First, use carbon inventory to get average carbon values per acre  
inventory <- read\_csv(here("results", "inventory\_16.csv")) %>%   
 clean\_names  
  
# crop report #s https://countyofsb.org/agcomm/cropReportArchive.sbc  
  
range\_acre\_19 <- 573678  
range\_acre\_12 <- 584125  
range\_acre\_16 <- 586047  
  
# assign total # of grassland acres from carbon inventory, then subtract from rangeland acreage to get shrubland weight  
range\_grass <- 239987.991  
range\_shrub <- range\_acre\_16 - range\_grass  
range\_weights <- c(range\_grass, range\_shrub)  
  
# get average aboveground and soil carbon storage per acre of grassland and shrubland  
shrub\_grass <- inventory %>%   
 filter(landcover\_classification %in% c("Grassland","Shrubland")) %>%   
 mutate(soil\_avg = total\_soil\_carbon\_mt\_c/acres) %>%   
 mutate(abvg\_avg = total\_aboveground\_carbon\_mt\_c/acres)  
  
# calculate average aboveground and soil carbon for rangeland  
avg\_soil\_rangel <- weighted.mean(shrub\_grass$soil\_avg, range\_weights)  
avg\_abvg\_rangel <- weighted.mean(shrub\_grass$abvg\_avg, range\_weights)  
  
# prepare rows to add to ag inventories  
range\_data <- data.frame(class = "Rangeland", "acres" = c(range\_acre\_12, range\_acre\_16, range\_acre\_19), "abvgc" = 0, "soilc" = 0, "noemit" = 0, "net" = 0, "year" = c(2012, 2016, 2019)) %>%   
 mutate(abvgc = avg\_abvg\_rangel\*acres) %>%   
 mutate(soilc = avg\_soil\_rangel\*acres) %>%   
 mutate(net = abvgc+soilc)  
  
# make combined dataframe to make calculations easier  
  
all\_ag\_df <- rbind(ag\_12, ag\_16) %>%   
 rbind(ag\_19) %>%   
 rbind(range\_data) %>%   
 filter(class != "Total")   
  
# calculate per acre storage and nitrous oxide emissions  
best\_at <- all\_ag\_df %>%   
 filter(year == 2016) %>%  
 mutate(best\_abvg = (abvgc/acres)) %>%  
 mutate(best\_soil = (soilc/acres)) %>%  
 mutate(best\_overall = ((abvgc+soilc)/acres)) %>%  
 mutate(worst\_n2o = (noemit/acres))  
  
# wide format dataframe  
all\_ag\_df\_wide <- all\_ag\_df %>%   
 pivot\_wider(names\_from = class,  
 values\_from = c(acres, abvgc, soilc, noemit, net)) %>%   
 clean\_names()

## Run linear regressions

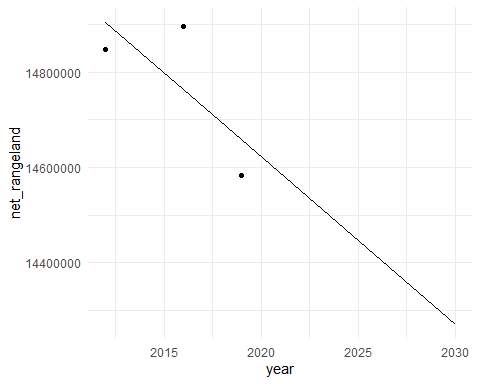
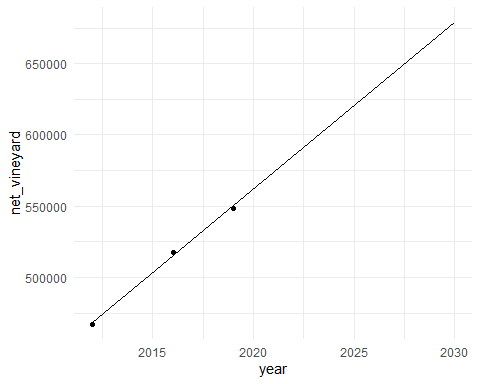
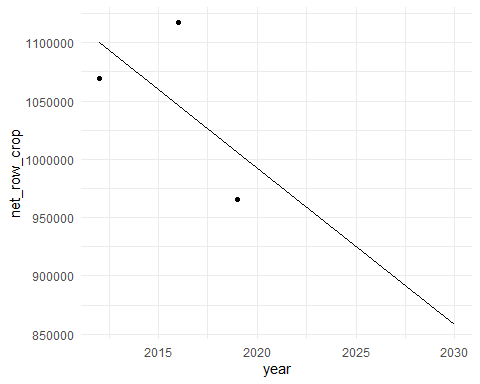
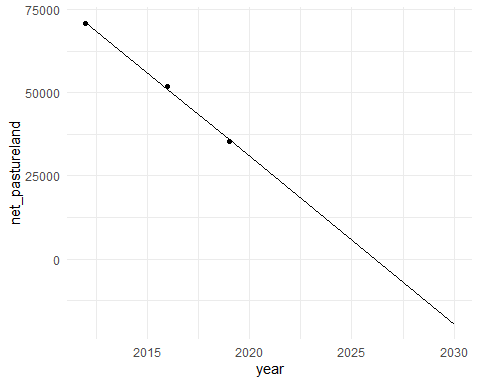
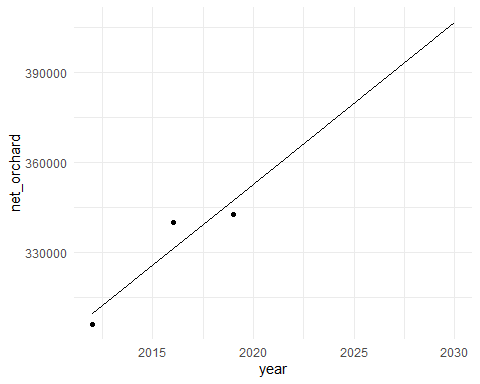
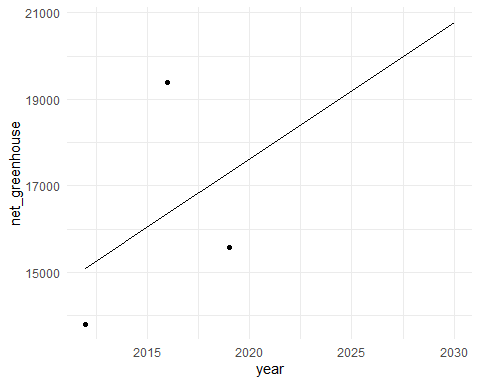
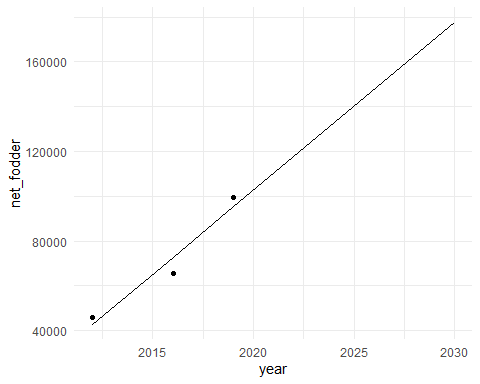
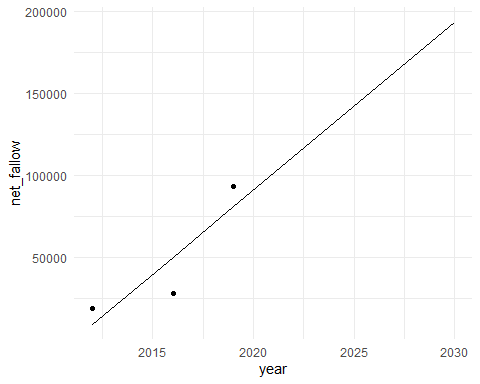
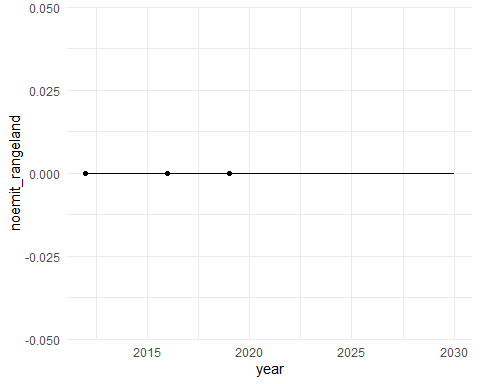
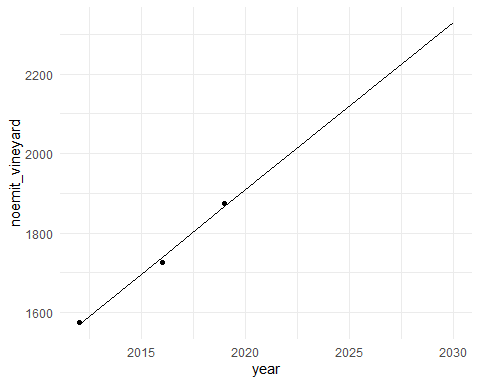
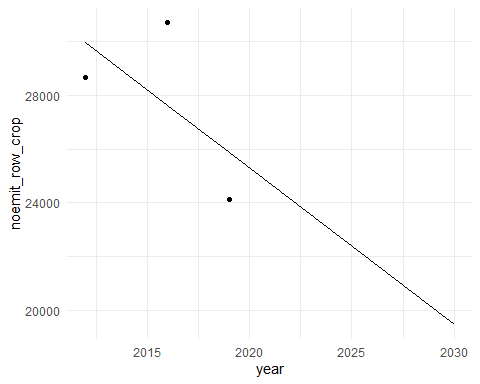
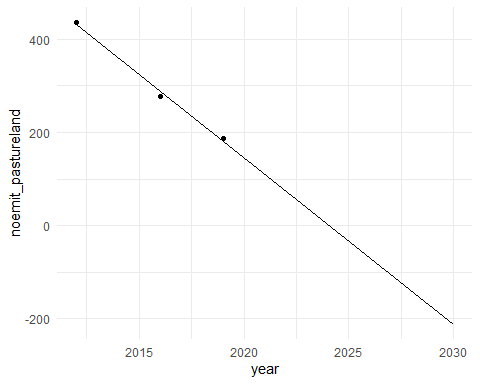
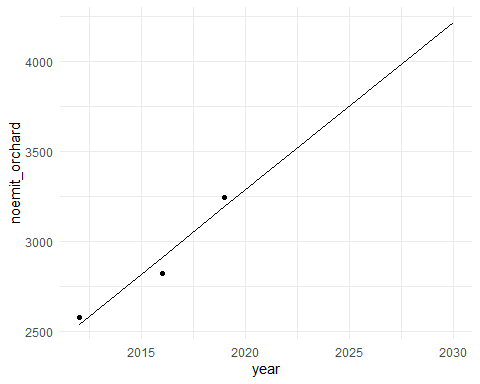
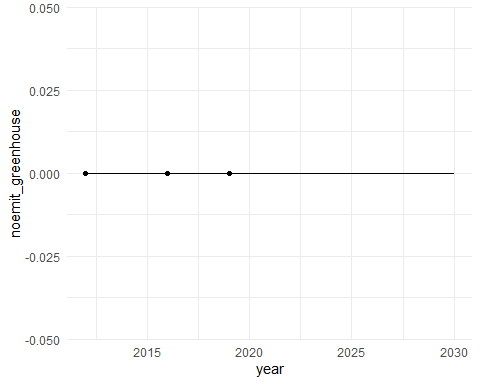
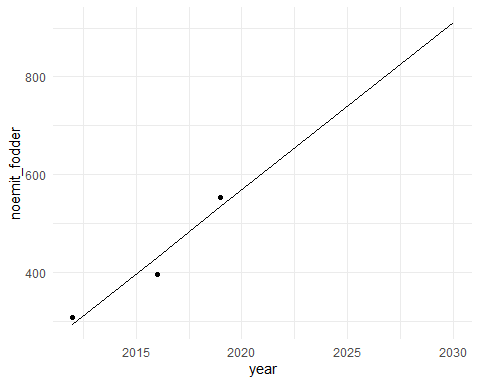
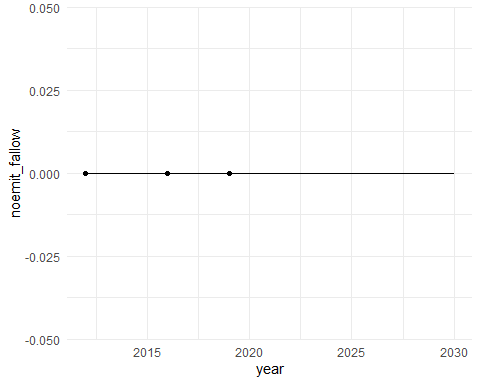
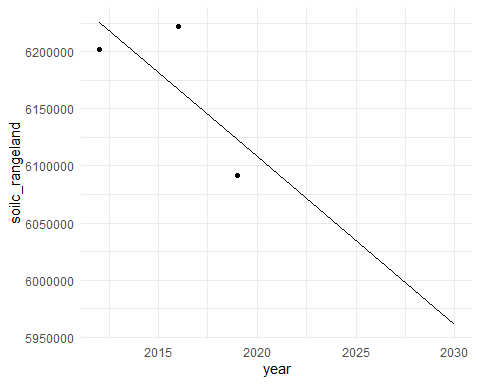
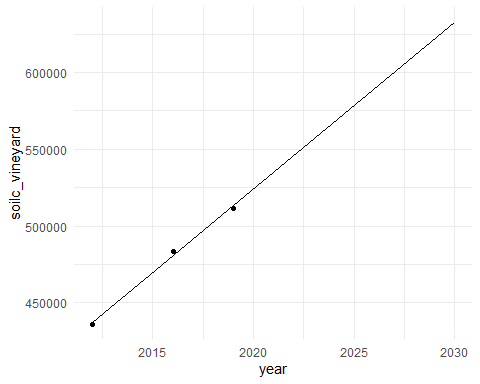
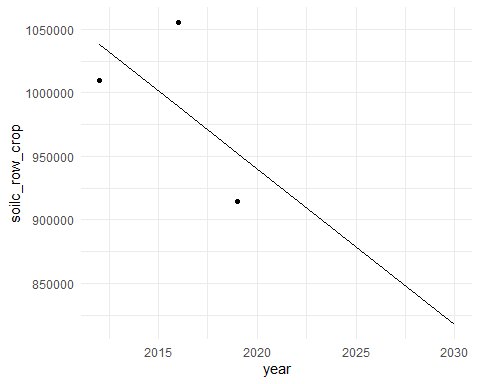
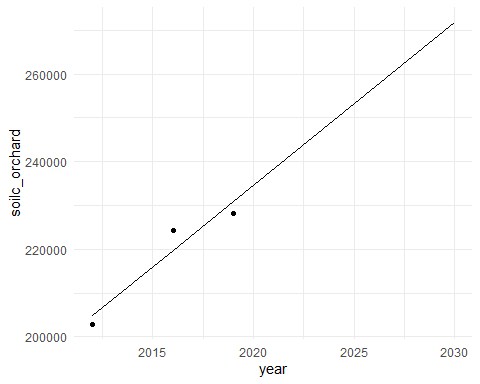
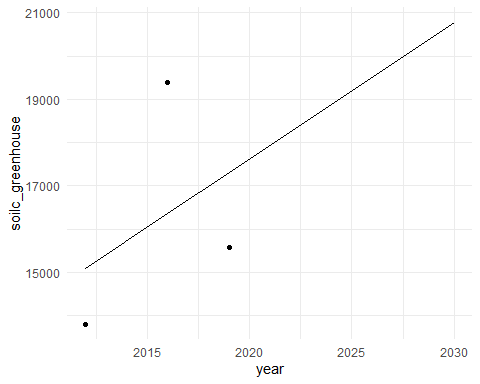
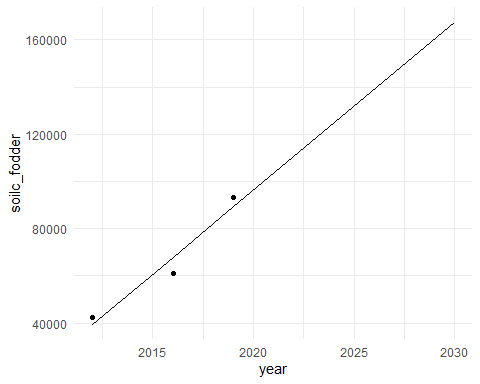
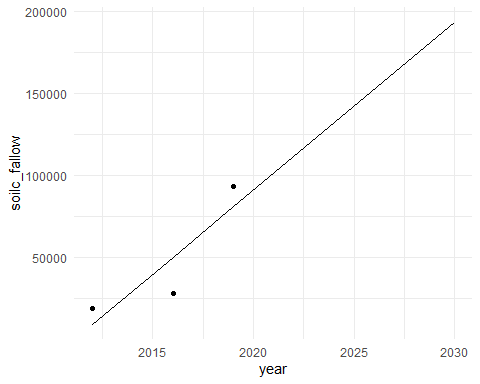
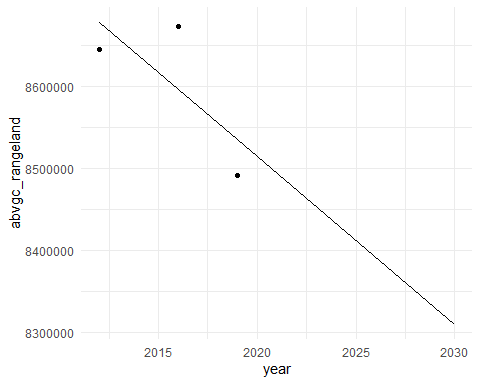
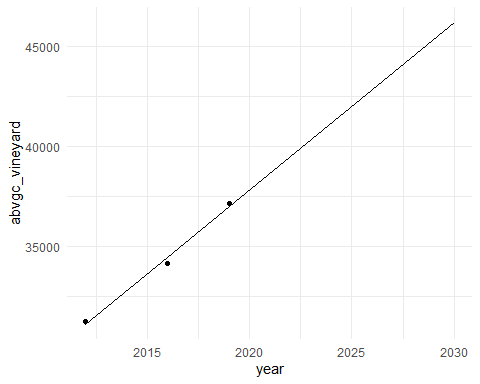
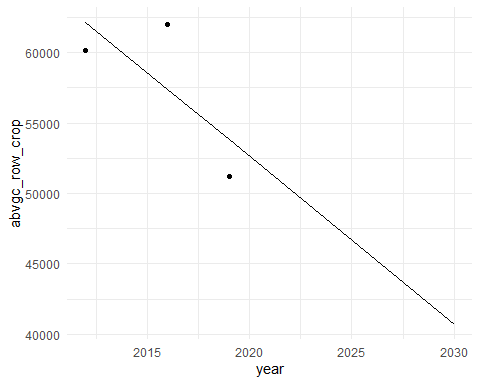
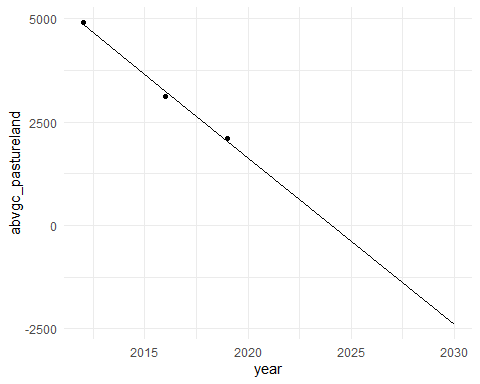
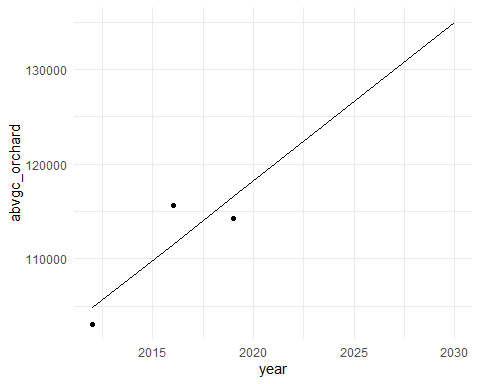
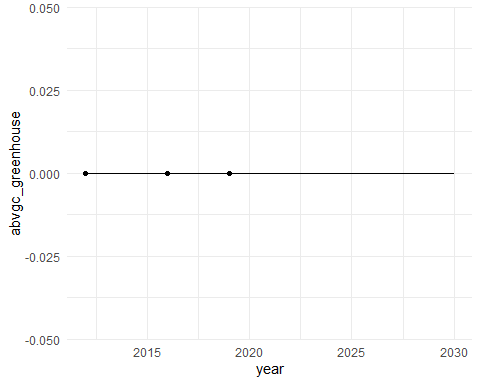
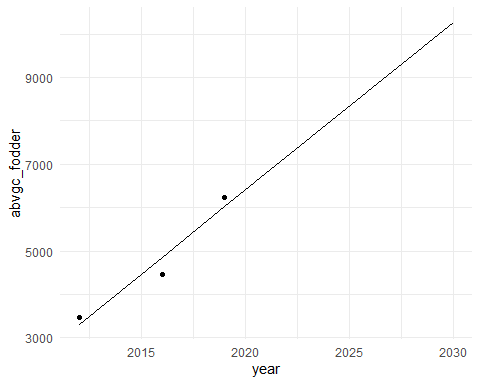
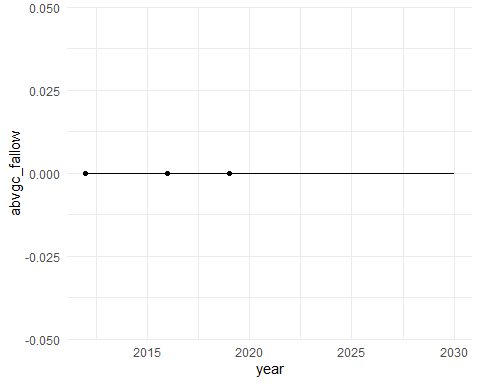
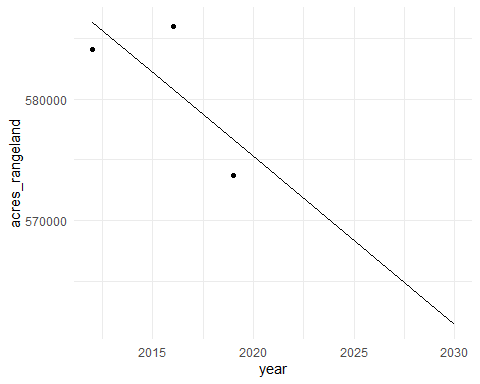
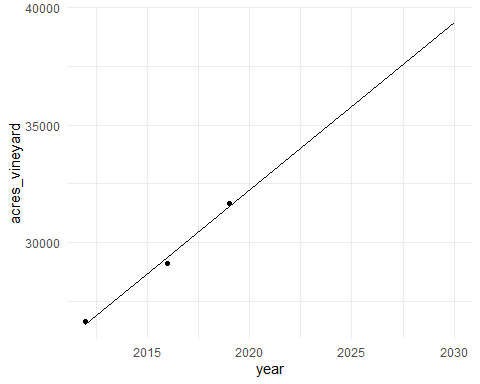
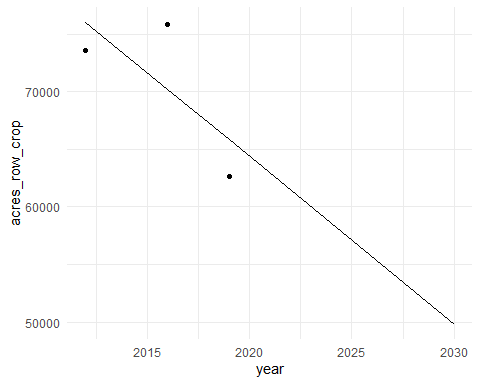
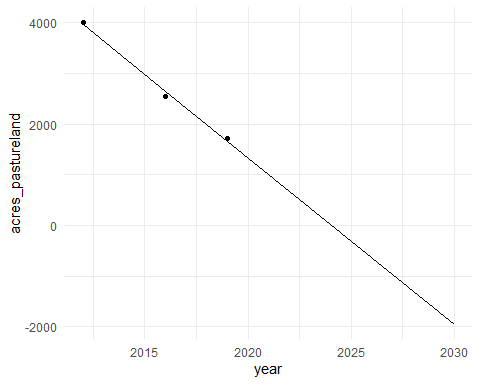
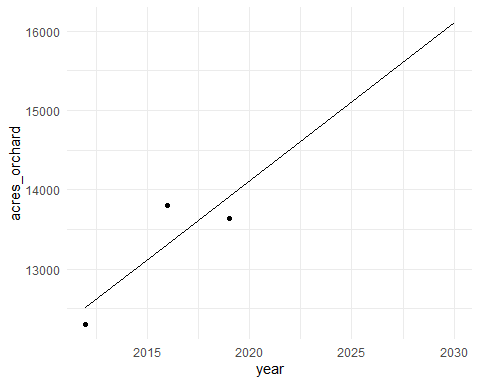
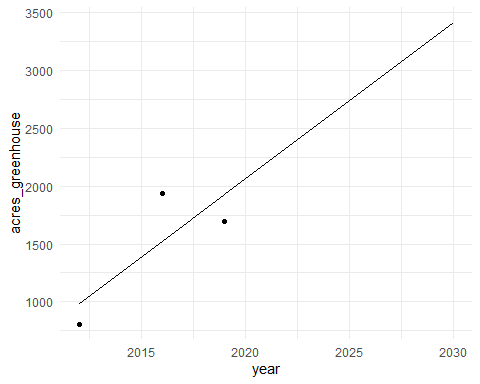
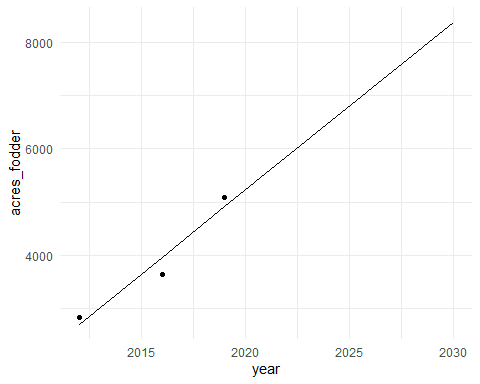
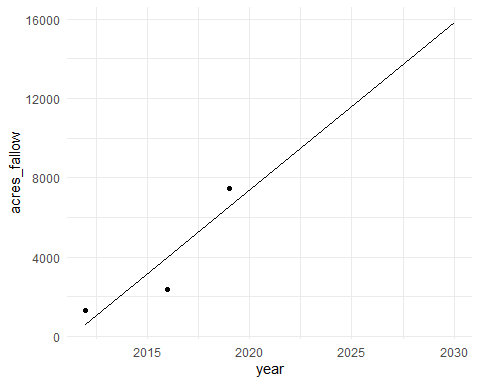
## First, we'll make a data frame of years out to 2030  
   
predict\_df\_30\_acres <- data.frame(year = c(2012, 2016, 2019, 2030))  
  
# Next we will build models to fit a linear regression to each crop category along with total ag acreage  
  
select\_ag <- all\_ag\_df\_wide %>%   
 dplyr::select(2:41)  
  
ag\_names <- colnames(select\_ag)  
  
res\_list\_plots <- vector("list", length = length(ag\_names)) %>%   
 setNames(ag\_names)  
  
# first, linear regressions  
  
fx\_lm <- function(name) {  
  
 lm\_loop <- lm(all\_ag\_df\_wide[[name]] ~ year, data = all\_ag\_df\_wide)  
   
}  
  
all\_lms <- lapply(ag\_names, fx\_lm) %>%   
 setNames(ag\_names)  
  
all\_lms[[1]] # cool

##   
## Call:  
## lm(formula = all\_ag\_df\_wide[[name]] ~ year, data = all\_ag\_df\_wide)  
##   
## Coefficients:  
## (Intercept) year   
## -1700970.1 845.7

# Now we will use these models to predict acreage (total and per crop type) in 2030  
  
fx\_predict <- function(lm) {  
   
 predict\_run\_loop <- predict(lm, newdata = predict\_df\_30\_acres)  
 predict\_df\_loop <- data.frame(predict\_df\_30\_acres, predict\_run\_loop)  
 predict\_df\_loop <- predict\_df\_loop  
   
}  
  
looped\_predict\_dfs <- lapply(all\_lms, fx\_predict) %>%   
 setNames(ag\_names)  
  
looped\_predict\_dfs[[1]] # great

## year predict\_run\_loop  
## 1 2012 611.5671  
## 2 2016 3994.4333  
## 3 2019 6531.5830  
## 4 2030 15834.4650

# These plots compare real data to the regression  
  
for (name in ag\_names) {  
 plot <- print(ggplot() +   
 geom\_point(data = all\_ag\_df\_wide, aes(x = year, y = all\_ag\_df\_wide[[name]])) +  
 geom\_line(data = looped\_predict\_dfs[[name]], aes(x = year, y = predict\_run\_loop)) +  
 theme\_minimal() +   
 labs(x = "year",  
 y = paste(name)))  
   
}



## Clean and prepare results for use

#create big dataframe of all predicted values (same units), easier for doing visualizations  
  
#first, change column names  
for(name in ag\_names){  
   
 colnames(looped\_predict\_dfs[[name]]) <- c("year", name)  
   
}  
  
# merge dataframes, common column = year  
all\_predict\_df<- looped\_predict\_dfs %>%   
 reduce(full\_join, by = "year")  
  
# make this dataframe tidy  
tidy\_predict\_df <- all\_predict\_df %>%   
 pivot\_longer(!year,  
 names\_to = c("variable", "land\_class"),  
 names\_sep = "\_",  
 values\_to = "value")  
  
# add totals  
totals\_predict <- tidy\_predict\_df %>%   
 group\_by(variable, year) %>%   
 summarise(value = sum(value)) %>%   
 mutate(land\_class = "total\_allclasses") %>%   
 relocate(year, variable, land\_class, value)  
  
# put it together  
final\_predict\_df <- bind\_rows(tidy\_predict\_df, totals\_predict)  
  
# do the same for observed values  
  
tidy\_observed <- all\_ag\_df %>%   
 pivot\_longer(cols = 2:6,  
 names\_to = "variable",  
 values\_to = "value") %>%   
 rename(land\_class = class) %>%   
 relocate(year, variable, land\_class, value)  
  
totals\_observed <- tidy\_observed %>%   
 group\_by(variable, year) %>%   
 summarise(value = sum(value)) %>%   
 mutate(land\_class = "total\_allclasses") %>%   
 relocate(year, variable, land\_class, value)  
  
final\_observed\_df <- bind\_rows(tidy\_observed, totals\_observed)