

Organic vs. conventional crop field summary statistics

The purpose of this Rmarkdown file is to do a preliminary comparison of fields in Kern County that grow organic versus conventional crops, as determined by CDFA APN records – to see complete methodology see Rmarkdown files 1-5. The comparison is done at the crop level so that fields with organically grown carrots are compared to fields with conventionally grown carrots, etc. The comparisons include:

- * Number of fields with crop_i
- * Total hectares with crop_i
- * Average crop_i field size
- * Average soil quality of fields with crop_i

```
library(rgdal)
library(sf)
library(tidyverse)
```

```
options(scipen = 9999) #
```

1. Prepare Kern Ag crop data for comparison

```
# Values for for() loop
buf_width = c(50) # c(0,50,100)
years = 2017 # 2013:2017

#### This forloop reads in KernAg_CDFA_join shapefile, the shapefile that joined Kern Ag with CDFA parcels
for(i in years){
  for(j in buf_width){

    tmp = readOGR(paste0("../R_output/spatial/KernAg_CDFA_pest/",i,"/B",j,"/KernAg_CDFA_Pest",i,"_B",j),
                  st_as_sf()) # Convert to spatial dataframe

    # This following chunk removes the word "ORGANIC" from the commodity column. This project uses CDFA data

    output = tmp %>%
      separate(col = "COMM",
               into = c("COMM_x","COMM_y"),
               sep = "-",
               remove = FALSE)

    # Make all COMM columns characters for if() statement matching
    output$COMM=as.character(output$COMM)
    output$COMM_x=as.character(output$COMM_x)
    output$COMM_y=as.character(output$COMM_y)

    # This if() statement eliminates the word "ORGANIC" from commodity names, but keeps the full original name
    output$COMM_new <- ifelse(output$COMM_y == "ORGANIC"|is.na(output$COMM_y),
                              output$COMM_x,
                              output$COMM)

    # Remove intermediate COMM columns
    output = output %>%
```

```

    dplyr::select(-COMM_x,-COMM_y)

    # Create output name based on year and buffer width
    assign(paste0("all_",i,"_B",j),output)
  }
}

## OGR data source with driver: ESRI Shapefile
## Source: "/Users/clairepowers/Desktop/Organics_Final/Working/R_files/R_output/spatial/KernAg_CDFA_pes
## with 8531 features
## It has 33 fields

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 8330
## rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
## 20, ...].

# The warning message notes which COMM observations do not have a hyphenated name, i.e. CARROTS-ORGANIC

```

2. Evaluate fields with organically grown crops, as determined by CDFA APN records. See Rmarkdown files 1-5 for detailed methods.

```

years = 2017 # Set years to evaluate
buf_width = c(50) # Set buffer widths to evaluate

for(i in years){
  for(j in buf_width){

    tmp = eval(as.name(paste0("all_",i,"_B",j))) %>% # Call dataframe that corresponds to the correct
    as.data.frame() %>% # Remove the spatial characteristic of the dataframe
    dplyr::select(-geometry) %>% # Remove the geometry column
    filter(CDFA == "1") %>% # Filter for CDFA Organic fields
    group_by(COMM_new) %>% # Group by the commodity
    summarise(total_fields = n(), # Count total number of fields for each commodity
              total_hectares = round(sum(AREA_HE),1), # Count hectares for each commodity
              average_field_size = round(mean(AREA_HE),1), # Find average field size for each commod
              average_SoilQ = mean(STORIE_, na.rm = T), # Find average soil quality for each commodi
              total_pest_app = sum(KgPstPr), # Total pesticide applied for this crop
              kg_per_hectare = sum(KgPstPr)/sum(acrstrt*0.0405)) %>% # Pesticide per hectare
    arrange(-total_fields) %>% # Arrange dataframe with total field column in descending order
    head(5) # Select the five crops with the most fields

    assign(paste0("ORG_",i,"_B",j,"_SumStats"),tmp)
  }
}

crops_B50 = ORG_2017_B50_SumStats$COMM_new # Create character vector of the 5 organic crops with the mo

```

3. Evaluate fields with conventionally grown crops

```

# buf_width = 100
#
# for(i in years){

```

```

#   for(j in buf_width){
#
#       tmp = eval(as.name(paste0("all_",i,"_B",j))) %>%
#       as.data.frame() %>%
#       select(-geometry) %>%
#       filter(is.na(CDFA) & COMM_new %in% crops_B100) %>% # Filter for nonCDFA (conventional) fields
#       group_by(COMM_new) %>%
#       summarise(total_fields = n(),
#                 total_hectares = round(sum(AREA_HE),1),
#                 average_field_size = round(mean(AREA_HE),1),
#                 average_SoilQ = mean(STORIE_, na.rm = T),
#                 total_pest_app = sum(KgPstPr),
#                 kg_per_hectare = sum(KgPstPr)/sum(acrstrt*0.0405)) %>%
#       arrange(total_fields = crops_B100) # Arrange order of the 'crops' character vector
#
#       assign(paste0("CONV_",i,"_B",j,"_SumStats"),tmp)
#   }
# }

buf_width = 50

for(i in years){
  for(j in buf_width){

    tmp = eval(as.name(paste0("all_",i,"_B",j))) %>%
    as.data.frame() %>%
    dplyr::select(-geometry) %>%
    filter(is.na(CDFA) & COMM_new %in% crops_B50) %>% # Filter for nonCDFA (conventional) fields and
    group_by(COMM_new) %>%
    summarise(total_fields = n(),
              total_hectares = round(sum(AREA_HE),1),
              average_field_size = round(mean(AREA_HE),1),
              average_SoilQ = mean(STORIE_, na.rm = T),
              total_pest_app = sum(KgPstPr),
              kg_per_hectare = sum(KgPstPr)/sum(acrstrt*0.0405)) %>%
    arrange(total_fields = crops_B50) # Arrange order of the 'crops' character vector

    assign(paste0("CONV_",i,"_B",j,"_SumStats"),tmp)

  }
}

```

4. Evaluate all agriculture fields in Kern County

```

# buf_width = 100
#
# for(i in years){
#   for(j in buf_width){
#
#       tmp = eval(as.name(paste0("all_",i,"_B",j))) %>%
#       as.data.frame() %>%
#       dplyr::select(-geometry) %>%

```

```

#       filter(COMM_new %in% crops_B100) %>% # Filter for nonCDFA (conventional) fields and the five m
#       group_by(COMM_new) %>%
#       summarise(total_fields = n(),
#                 total_hectares = round(sum(AREA_HE),1),
#                 average_field_size = round(mean(AREA_HE),1),
#                 average_SoilQ = mean(STORIE_, na.rm = T),
#                 total_pest_app = sum(KgPstPr),
#                 kg_per_hectare = sum(KgPstPr)/sum(acrstrt*0.0405)) %>%
#       arrange(total_fields = crops_B100) # Arrange order of the 'crops' character vector
#
#       assign(paste0("ALL_",i,"_B",j,"_SumStats"),tmp)
#
#   }
# }

buf_width = 50

for(i in years){
  for(j in buf_width){

    tmp = eval(as.name(paste0("all_",i,"_B",j))) %>%
    as.data.frame() %>%
    dplyr::select(-geometry) %>%
    filter(COMM_new %in% crops_B50) %>% # Filter for nonCDFA (conventional) fields and the five most
    group_by(COMM_new) %>%
    summarise(total_fields = n(),
              total_hectares = round(sum(AREA_HE),1),
              average_field_size = round(mean(AREA_HE),1),
              average_SoilQ = mean(STORIE_, na.rm = T),
              total_pest_app = sum(KgPstPr),
              kg_per_hectare = sum(KgPstPr)/sum(acrstrt*0.0405)) %>%
    arrange(total_fields = crops_B50) # Arrange order of the 'crops' character vector

    assign(paste0("ALL_",i,"_B",j,"_SumStats"),tmp)

  }
}

```

5. Combine data into a table and export

```

combined = cbind(ALL_2017_B50_SumStats, CONV_2017_B50_SumStats, ORG_2017_B50_SumStats) # Combined three o

colnames(combined) = c("crop", "total_fields", "total_hectares", "avg_field_H", "avg_soilQ", "total_pest_use",
                      "crop2", "total_C_fields", "total_C_hectares", "avg_C_field_ar", "avg_C_soilQ", "total_pest_app",
                      "crop3", "total_O_fields", "total_O_hectares", "avg_O_field_ar", "avg_O_soilQ", "total_pest_app")

combined = dplyr::select(combined, c(-crop2, -crop3)) # Remove redundant crop columns
combined = combined[, c(1,2,8,14,3,9,15,4,10,16,5,11,17,6,12,18,7,13,19)] # Reorder columns to have summ
combined

```

##	crop	total_fields	total_C_fields	total_O_fields	total_hectares
## 1	GRAPE	971	915	56	26301.8
## 2	CARROT	342	299	43	9624.9
## 3	LETTUCE LEAF	55	30	25	1163.7

## 4	ORANGE	604	581	23	13125.6
## 5	SWISS CHARD	43	25	18	974.2
##	total_C_hectares	total_O_hectares	avg_field_H	avg_C_field_ar	
## 1	24726.3	1575.6	27.1	27.0	
## 2	8345.2	1279.6	28.1	27.9	
## 3	721.3	442.4	21.2	24.0	
## 4	12597.3	528.3	21.7	21.7	
## 5	648.3	325.9	22.7	25.9	
##	avg_O_field_ar	avg_soilQ	avg_C_soilQ	avg_O_soilQ	total_pest_use
## 1	28.1	1.963132	1.987640	1.554667	2826400.899
## 2	29.8	1.746617	1.770667	1.614634	2686849.832
## 3	17.7	1.479423	1.591481	1.358400	15855.149
## 4	23.0	1.814571	1.807692	1.943125	1422019.110
## 5	18.1	1.472619	1.509167	1.423889	3955.189
##	total_C_pest_use	total_O_pest_use	pest_kgperH	pest_C_kgperH	
## 1	2514201.00	312199.9032	33.52574	31.60488	
## 2	2586366.55	100483.2824	363.63832	385.99326	
## 3	10829.32	5025.8338	32.15785	35.05427	
## 4	1391272.99	30746.1156	58.48045	58.30739	
## 5	3131.27	823.9195	22.37027	22.27503	
##	pest_O_kgperH				
## 1	65.66578				
## 2	145.99846				
## 3	27.29779				
## 4	67.55324				
## 5	22.73978				

```
write_csv(combined,"../R_output/CSV/Comparison_SummaryStats_B50.csv") # Write output CSV
```

```
#
# combined = cbind(ALL_2017_B100_SumStats,CONV_2017_B100_SumStats,ORG_2017_B100_SumStats) # Combined th
#
# colnames(combined) =c("crop","total_fields","total_hectares","avg_field_H","avg_soilQ","total_pest_us
# "crop2","total_C_fields","total_C_hectares","avg_C_field_ar","avg_C_soilQ","tot
# "crop3","total_O_fields","total_O_hectares","avg_O_field_ar","avg_O_soilQ","tot
#
# combined = dplyr::select(combined,c(-crop2,-crop3)) # Remove redundant crop columns
# combined = combined[,c(1,2,8,14,3,9,15,4,10,16,5,11,17,6,12,18,7,13,19)] # Reorder columns to have su
# combined
#
# write_csv(combined,"../R_output/CSV/Comparison_SummaryStats_B100.csv") # Write output CSV
```