Strawberry Analysis - MA615

Jordan Stout

2024-10-10

In this document I will be explaining the process of cleaning open-source USDA data on Strawberry production in the US from 2018 to 2024. The cleaned data is exported as 4 separate CSV files: survey_data and census_data includes data on strawberry production that was collected through surveys and the census respectively, chemicals includes all the data regarding chemical use in strawberry production, and census_organic_data includes all data involving organic strawberry sales as per the census.

First we will load the data

```
strawberry <- read_csv("strawberries25_v3.csv", col_names = TRUE)
```

```
## Rows: 12669 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (15): Program, Period, Geo Level, State, State ANSI, Ag District, County...
## dbl  (2): Year, Ag District Code
## lgl  (4): Week Ending, Zip Code, Region, Watershed
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Now we check to make sure that every row is associated with a state.

```
state_all <- strawberry |> distinct(State)

state_all1 <- strawberry |> group_by(State) |> count()
```

Some columns of the data only have one value which is useless to us. To assist us in data cleaning we create a function 'drop_one_value_col()' which will remove such columns.

```
drop_one_value_col <- function(df){
  drop <- NULL
  for(i in 1:dim(df)[2]){
    if((df |> distinct(df[,i]) |> count()) == 1){
    drop = c(drop, i)
    }
  }
  if(is.null(drop)){
    return("none")}
  else{
     strawberry <- df[, -1*drop]
  }
```

```
}
strawberry <- drop_one_value_col(strawberry)
drop_one_value_col(strawberry)
```

```
## [1] "none"
```

Start by removing bad values from 'Value' and 'CV (%)' columns and make 'Value' column numeric type

```
strawberry$Value[strawberry$Value == "(D)"] <- NA
strawberry$Value[strawberry$Value == "(Z)"] <- NA

strawberry$`CV (%)`[strawberry$`CV (%)` == "(D)"] <- NA
strawberry$`CV (%)`[strawberry$`CV (%)` == "(L)"] <- NA
strawberry$`CV (%)`[strawberry$`CV (%)` == "(H)"] <- NA
```

We will only keep values of 'Geo Level' that include NATIONAL AND STATE becasue the rest is useless to us.

```
strawberry <- strawberry |>
  filter(`Geo Level`== "NATIONAL" | `Geo Level`== "STATE")
```

We will create two new data frames. 'census' will contain all the data where the program was the census, likewise 'survey' will include those where the program was survey.

```
census <- strawberry |>
  filter(Program=="CENSUS")

census <- census |>
  drop_one_value_col()

survey <- strawberry |>
  filter(Program == "SURVEY")

survey <- survey |>
  drop_one_value_col()

nrow(strawberry) == (nrow(survey) + nrow(census))
```

```
## [1] TRUE
```

Clean and isolate the chemicals used by creating two new columns: 'chem_name' and 'chem_num'

```
survey <- survey |>
  mutate(`Domain Category` = gsub(".*: \\(([^=]+) = ([0-9]+)\\)", "\\1,\\2", `Domain Category`)) |>
  separate(`Domain Category`, into = c("chem_name", "chem_num"), sep = ",") |>
  mutate(chem_name = trimws(chem_name), chem_num = as.numeric(trimws(chem_num)))
```

Extract 'use' from 'domain' column

```
survey <- survey |>
  separate(Domain, into = c("Domain", "use"), sep = ",", extra = "merge")
```

Use stringr + regexs to sort out the 'Data Item' column

```
survey <- survey |>
  mutate(measurement = str_extract(`Data Item`, "(?<=MEASURED\\s).*")) |>
  mutate(`Data Item` = str_remove(`Data Item`, "MEASURED.*")) |>
  separate(`Data Item`, into = c("Data Item", "category"), sep = "[,-]", extra = "merge", fill = "right
  mutate(measurement = gsub("\\bIN\\b", "", measurement)) |>
  mutate(measurement = trimws(measurement))
```

Remove 'Data Item' column because it has no more data that we need as well as remove unwanted commas

```
survey <- survey |>
  select(-`Data Item`)
survey$category <- gsub(",", "", survey$category)
```

Separate chemicals and total

```
total <- survey |>
  filter(Domain == "TOTAL")

chemicals <- survey |>
  filter(Domain == "CHEMICAL")

total <- drop_one_value_col(total)
chemicals <- drop_one_value_col(chemicals)
```

Organize 'Data Item' column in Census data

```
census <- census |>
  separate_wider_delim(cols = `Data Item`, delim = " - ", names = c("strawberries", "Category"), too_ma
```

Make a new dataframe out of the census specific data that includes only organic strawberry sales

```
census <- census |>
  separate_wider_delim(cols = strawberries, delim = ", ", names = c("strawberries", "ORGANIC", "organic
                       too_few = "align_start") |>
  drop_one_value_col()

census_organic <- census |>
  filter(ORGANIC == "ORGANIC")

census <- census[(is.na(census$ORGANIC)),]

census <- census |>
  drop_one_value_col()
```

Now we can split the category column in census into two columns. Measure and bearing to organize the bearing type.

```
census <- census |>
  separate_wider_delim(cols = `Category`, delim = " ", names = c("COL1", "COL2"), too_many = "merge", to

census$COL2 <- str_replace(census$COL2,"WITH ","")

census <- census |>
  rename(Measure = COL1, Bearing_type= COL2)
```

Split up 'Domain Category' column in the census dataframe

```
census <- census |>
  rename(size_bracket = `Domain Category`)

census$size_bracket <- str_replace(census$size_bracket, "NOT SPECIFIED", "TOTAL")

census$size_bracket <- str_replace(census$size_bracket, "AREA GROWN: ", "")
```

Now onto the organic census data. First lets drop all the one value columns

```
census_organic <- census_organic |>
  drop_one_value_col()
```

Now we can clean "Category" column similar to how we cleaned it with the survey data. By splitting the columns at the word measure.

```
census_organic <- census_organic |>
  separate(Category, into = c("Category", "measurement"), sep = " MEASURED ", extra = "merge", fill = "
```

Get rid of commas and 'IN' from the new column it is just units again.

```
census_organic$Category <- gsub(",", "", census_organic$Category)
census_organic$measurement <- gsub("IN", "", census_organic$measurement, ignore.case = TRUE)
```

Change values in 'Value' column to numeric type

```
census <- census |>
  mutate(Value = as.numeric(gsub(",", "", Value)))
```

This is a function that calculates ratios for each state by dividing total acres grown by total bearing acres, then applies those ratios to the total acres grown in the same size bracket where the value is marked as NA.

```
fix_bearing_na <- function(df) {
  df_ratio <- df
  states <- unique(df$State)
  for (state in states) {
    state_data <- df |>
      filter(State == state)

    grown_value <- state_data |>
      filter(Bearing_type == "GROWN", Domain == "TOTAL", size_bracket == "TOTAL") |>
```

```
      pull(Value)

    bearing_value <- state_data |>
      filter(Bearing_type == "BEARING", Domain == "TOTAL", size_bracket == "TOTAL") |>
      pull(Value)

    if (length(grown_value) > 0 && length(bearing_value) > 0 && grown_value > 0) {
      percentage <- bearing_value / grown_value}
    else {
      next
    }

    df_ratio <- df_ratio |>
      mutate(Value = ifelse(is.na(Value) & Bearing_type == "BEARING" & State == state, percentage * Valu
  }
  return(df_ratio)
}
```

Same function but for non-bearing acres

```
fix_nonbearing_na <- function(df) {
  df_ratio <- df

  states <- unique(df$State)

  for (state in states) {
    state_data <- df |>
      filter(State == state)

    grown_value <- state_data |>
      filter(Bearing_type == "GROWN", Domain == "TOTAL", size_bracket == "TOTAL") |>
      pull(Value)

    non_bearing_value <- state_data |>
      filter(Bearing_type == "NON-BEARING", Domain == "TOTAL", size_bracket == "TOTAL") |>
      pull(Value)

    if (length(grown_value) > 0 && length(non_bearing_value) > 0 && grown_value > 0) {
      percentage <- non_bearing_value / grown_value}
    else{
      next
    }

    df_ratio <- df_ratio |>
      mutate(Value = ifelse(is.na(Value) & Bearing_type == "NON-BEARING" & State == state, percentage *
  }
  return(df_ratio)
}
```

Apply both these functions

```
census <- fix_bearing_na(census)
census <- fix_nonbearing_na(census)
```

Write CSVs

```
# write.csv(survey, "survey_data.csv", row.names = FALSE)
# write.csv(census, "census_data.csv", row.names = FALSE)
# write.csv(chemicals, "chemicals.csv", row.names = FALSE)
# write.csv(census_organic, "census_organic_data.csv", row.names = FALSE)
```

Chemical Analysis

I like to believe that the EPA is not sentencing me to death by eating strawberries, so I will be using a dataset found through the USDA Pesticide Data Program which is a national pesticide residue monitoring program.

```
usda_chem_original<-read.csv("usda_chem_sample_data.csv")
library(tidyverse)
```

Do some initial cleaning. We have been working with Florda and California data, but we will include all samples whose origin is the US. We will subset the data to include the sample ID, pesticide names, the concentration found in the sample (ppm), and the established EPA tolerance level for that pesticide (ppm).

```
usda_chem<-as.data.frame(usda_chem_original)

usda_chem <- usda_chem |>
  filter(Country == "") |>
  select(Pesticide.Name, EPA.Tolerance..ppm., Concentration, Sample.ID) |>
  mutate(EPA.Tolerance..ppm. = as.numeric(EPA.Tolerance..ppm.))
usda_chem<-na.omit(usda_chem)
```

Now we will do some analysis to see if any samples exceed that EPA tolerance level.

```
usda_chem <- usda_chem |>
  mutate(uh_oh = Concentration > EPA.Tolerance..ppm.)

is_true<-usda_chem |>
  filter(uh_oh == TRUE)
is_true
```

```
##   Pesticide.Name EPA.Tolerance..ppm. Concentration          Sample.ID uh_oh
## 1   Pyriproxyfen                 0.3          0.37 OH1811260207SZWA1P  TRUE
```

We can see that there was one instance of a detected pesticide being about the EPA tolerance level: Pyriproxyfen. As per the EPAs Human Health Draft Risk Assessment, Pyriproxyfen has an RfD of 0.01mg/kg/day. This means that for every kilogram a persons weight, they are allowed to consume 0.01 mg/day. Let's see how many strawberries I would have to eat to experience the adverse effects of Pyriproxyfen using me as an example.

Safe Exposure = RfD x Body Weight = 0.01mg/kg/day x 65kg = 0.7mg/day

Kilograms of Strawberries = Safe Exposure / Concentration Found = 0.7mg/day / 0.37mg/kg = 1.89kg

Number of Strawberries = 1.89kg / 0.02kg = 95 Strawberries

If you find me eating 95 strawberries, please know I have bigger issues than Pyriproxyfen toxicity.

Let's talk about Chlorpyrifos, widely considered one of the most dangerous pesticides used in strawberry production. It has been linked to Reduced IQ, Attention disorders, Autism, Neurodevelopmental disorders, and Death. Chlorpyrifos was banned in the United States in March 2021 (6 years after the original proposition to ban the substance). It was later reinstated its use in August 2022 stating the EPA had failed to follow proper procedures.

We will now analyze the instances of Chlorpyrifos being detected in samples of US grown strawberries.

```
chlor <- usda_chem |>
  filter(Pesticide.Name == "Chlorpyrifos")
chlor
```

```
##   Pesticide.Name EPA.Tolerance..ppm. Concentration         Sample.ID uh_oh
## 1   Chlorpyrifos                 0.2         0.008 TX1906100105SZWA1 FALSE
```

As we can see, out of 2141 samples taken across the US there was one instance of Chlorpyrifos detection where is was recorded at 4% of the EPA defined tolerance level.

Malathion is, like Chlorpyrifos, considered an extremely dangerous Organophosphate used in strawberry production. Let's see how much gets into our food.

```
mala <- usda_chem |>
  filter(Pesticide.Name == "Malathion")
mala
```

```
##     Pesticide.Name EPA.Tolerance..ppm. Concentration          Sample.ID uh_oh
## 1        Malathion                   8        0.0120 CA1810150001SZWA1P FALSE
## 2        Malathion                   8        0.0084 CA1810150004SZWA1P FALSE
## 3        Malathion                   8        0.0052 CA1810150151SZWA1P FALSE
## 4        Malathion                   8        0.0038  CA1810150504SZWA1 FALSE
## 5        Malathion                   8        0.0050  CA1810150554SZWA1 FALSE
## 6        Malathion                   8        0.0120  CA1811140516SZWA1 FALSE
## 7        Malathion                   8        0.0140 CA1811140553SZWA1P FALSE
## 8        Malathion                   8        0.0020  CA1811140554SZWA1 FALSE
## 9        Malathion                   8        0.0250 CA1811140561SZWA1P FALSE
## 10       Malathion                   8        0.0045  CA1812170554SZWA1 FALSE
## 11       Malathion                   8        0.0065  CO1810220010SZWA1 FALSE
## 12       Malathion                   8        0.0230  CO1811260010SZWA1 FALSE
## 13       Malathion                   8        0.0047  FL1811260040SZWA1 FALSE
## 14       Malathion                   8        0.0046  FL1812030040SZWA1 FALSE
## 15       Malathion                   8        0.0260 MD1811280042SZWA1P FALSE
## 16       Malathion                   8        0.0047 MD1811280057SZWA1P FALSE
## 17       Malathion                   8        0.0380  MI1810230005SZWA1 FALSE
## 18       Malathion                   8        0.0190  MI1810230006SZWA1 FALSE
## 19       Malathion                   8        0.0240  MI1810230021SZWA1 FALSE
## 20       Malathion                   8        0.0062  MI1811270006SZWA1 FALSE
## 21       Malathion                   8        0.0170 MI1811270056SZWA1P FALSE
## 22       Malathion                   8        0.0034  MI1812040005SZWA1 FALSE
## 23       Malathion                   8        0.0110  MI1812040053SZWA1 FALSE
```

```
## 24       Malathion                 8        0.0180 NY1812170249SZWA1P FALSE
## 25       Malathion                 8        0.0370 NY1812170265SZWA1P FALSE
## 26       Malathion                 8        0.0420  OH1810220118SZWA1 FALSE
## 27       Malathion                 8        0.0230  OH1811260118SZWA1 FALSE
## 28       Malathion                 8        0.0140 OH1812260203SZWA1P FALSE
## 29       Malathion                 8        0.0360  TX1811120103SZWA1 FALSE
## 30       Malathion                 8        0.0022 TX1811120216SZWA1P FALSE
## 31       Malathion                 8        0.0036 TX1812170216SZWA1P FALSE
## 32       Malathion                 8        0.0120 CA1901140004SZWA1P FALSE
## 33       Malathion                 8        0.0081 CA1901140151SZWA1U FALSE
## 34       Malathion                 8        0.0150 CA1901140561SZWA1P FALSE
## 35       Malathion                 8        0.0140  CA1902110513SZWA1 FALSE
## 36       Malathion                 8        0.0056  CA1902110547SZWA1 FALSE
## 37       Malathion                 8        0.0041  CA1902110549SZWA1 FALSE
## 38       Malathion                 8        0.0130 CA1902110561SZWA1P FALSE
## 39       Malathion                 8        0.0150 CA1902110621SZWA1A FALSE
## 40       Malathion                 8        0.0210 CA1902110621SZWA1B FALSE
## 41       Malathion                 8        0.0230 CA1903110001SZWA1P FALSE
## 42       Malathion                 8        0.0130 CA1903110150SZWA1P FALSE
## 43       Malathion                 8        0.0096  CA1903110549SZWA1 FALSE
## 44       Malathion                 8        0.0300  CA1903110554SZWA1 FALSE
## 45       Malathion                 8        0.0200  CA1903110556SZWA1 FALSE
## 46       Malathion                 8        0.0150 CA1904080150SZWA1P FALSE
## 47       Malathion                 8        0.0037  CA1904080513SZWA1 FALSE
## 48       Malathion                 8        0.0100  CA1904080521SZWA1 FALSE
## 49       Malathion                 8        0.0170 CA1904080638SZWA1P FALSE
## 50       Malathion                 8        0.0220 CA1905130149SZWA1P FALSE
## 51       Malathion                 8        0.0190  CA1906100533SZWA1 FALSE
## 52       Malathion                 8        0.0069  CA1906100549SZWA1 FALSE
## 53       Malathion                 8        0.0058  CA1907080547SZWA1 FALSE
## 54       Malathion                 8        0.0910 CA1907080630SZWA1P FALSE
## 55       Malathion                 8        0.0041 CA1907080638SZWA1P FALSE
## 56       Malathion                 8        0.0100  CA1907080646SZWA1 FALSE
## 57       Malathion                 8        0.0044 CA1907080650SZWA1P FALSE
## 58       Malathion                 8        0.0069 CA1908120004SZWA1P FALSE
## 59       Malathion                 8        0.0470  CO1901070010SZWA1 FALSE
## 60       Malathion                 8        0.0140  CO1904010017SZWA1 FALSE
## 61       Malathion                 8        0.0028  CO1905060017SZWA1 FALSE
## 62       Malathion                 8        0.0079  CO1906030010SZWA1 FALSE
## 63       Malathion                 8        0.0250  FL1901070060SZWA1 FALSE
## 64       Malathion                 8        0.0550 FL1901070084SZWA1P FALSE
## 65       Malathion                 8        0.0240  FL1902040058SZWA1 FALSE
## 66       Malathion                 8        0.0033 FL1902040068SZWA1B FALSE
## 67       Malathion                 8        0.0530  FL1904010040SZWA1 FALSE
## 68       Malathion                 8        0.0290 FL1904010087SZWA1P FALSE
## 69       Malathion                 8        0.0180 FL1905060029SZWA1P FALSE
## 70       Malathion                 8        0.0034 FL1906030040SZWA1P FALSE
## 71       Malathion                 8        0.0230 FL1906030057SZWA1P FALSE
## 72       Malathion                 8        0.0170 FL1907010055SZWA1P FALSE
## 73       Malathion                 8        0.0240 MD1904030005SZWA1P FALSE
## 74       Malathion                 8        0.0260  MI1901080021SZWA1 FALSE
## 75       Malathion                 8        0.0120  MI1901080056SZWA1 FALSE
## 76       Malathion                 8        0.0580 MI1902050008SZWA1P FALSE
## 77       Malathion                 8        0.0160 MI1902050056SZWA1P FALSE
```

```
## 78       Malathion                  8        0.0110  MI1902050103SZWA1 FALSE
## 79       Malathion                  8        0.0620 MI1903050006SZWA1P FALSE
## 80       Malathion                  8        0.0270  MI1904020047SZWA1 FALSE
## 81       Malathion                  8        0.0150  MI1904020103SZWA1 FALSE
## 82       Malathion                  8        0.0330 MI1906040091SZWA1P FALSE
## 83       Malathion                  8        0.0038  MI1907020005SZWA1 FALSE
## 84       Malathion                  8        0.0042 MI1907020008SZWA1P FALSE
## 85       Malathion                  8        0.0230 MI1909030005SZWA1A FALSE
## 86       Malathion                  8        0.0370  MI1909030027SZWA1 FALSE
## 87       Malathion                  8        0.0051 NC1901160547SZWA1A FALSE
## 88       Malathion                  8        0.0053 NC1902130536SZWA1A FALSE
## 89       Malathion                  8        0.0140  NC1907100613SZWA1 FALSE
## 90       Malathion                  8        0.0310 NC1909110520SZWA1A FALSE
## 91       Malathion                  8        0.0240 NY1901150009SZWA1P FALSE
## 92       Malathion                  8        0.0069 NY1902120285SZWA1P FALSE
## 93       Malathion                  8        0.0460 NY1903120272SZWA1P FALSE
## 94       Malathion                  8        0.0140 NY1903120903SZWA1P FALSE
## 95       Malathion                  8        0.0440 NY1905140237SZWA1P FALSE
## 96       Malathion                  8        0.0098 OH1901070250SZWA1P FALSE
## 97       Malathion                  8        0.0045 OH1903040304SZWA1P FALSE
## 98       Malathion                  8        0.0190 OH1903040440SZWA1P FALSE
## 99       Malathion                  8        0.0210 OH1904010203SZWA1P FALSE
## 100      Malathion                  8        0.0160 OH1905060440SZWA1P FALSE
## 101      Malathion                  8        0.0240 TX1901140216SZWA1P FALSE
## 102      Malathion                  8        0.0180  TX1905130105SZWA1 FALSE
## 103      Malathion                  8        0.0330 TX1905130216SZWA1P FALSE
## 104      Malathion                  8        0.0430 TX1906100216SZWA1P FALSE
## 105      Malathion                  8        0.0240  WA1906030011SZWA1 FALSE
## 106      Malathion                  8        0.0082  CA1812170547SZWA1 FALSE
## 107      Malathion                  8        0.0230 NC1812120510SZWA1A FALSE
## 108      Malathion                  8        0.0230  CA1901140557SZWA1 FALSE
## 109      Malathion                  8        0.0110 CA1902110001SZWA1P FALSE
## 110      Malathion                  8        0.0210  CO1908050012SZWA1 FALSE
## 111      Malathion                  8        0.0160 NY1902120222SZWA1P FALSE
## 112      Malathion                  8        0.0150 OH1902120118SZWA1P FALSE
## 113      Malathion                  8        0.0160 TX1904080216SZWA1P FALSE
## 114      Malathion                  8        0.0560  WA1903040011SZWA1 FALSE
```

```
max(mala$Concentration)
```

```
## [1] 0.091
```

There are many Malathion detection events, however, the maximum concentration found was 0.091ppm. This is merely 1.1% of the EPA declared tolerance level.

Finally we will explore Carbendazim, a fungicide associated with reproductive toxicity and potential endocrine disruption.

```
carb <- usda_chem |>
  filter(Pesticide.Name == "Carbendazim (MBC)")
carb
```

```
##       Pesticide.Name EPA.Tolerance..ppm. Concentration          Sample.ID uh_oh
```

```
## 1  Carbendazim (MBC)                7        0.0096 CA1810150001SZWA1P FALSE
## 2  Carbendazim (MBC)                7        0.0120  CA1810150504SZWA1 FALSE
## 3  Carbendazim (MBC)                7        0.0021 CA1811140561SZWA1P FALSE
## 4  Carbendazim (MBC)                7        0.0250  CA1812170450SZWA1 FALSE
## 5  Carbendazim (MBC)                7        0.0250  CO1810220008SZWA1 FALSE
## 6  Carbendazim (MBC)                7        0.0310  CO1810220010SZWA1 FALSE
## 7  Carbendazim (MBC)                7        0.0130  FL1811260040SZWA1 FALSE
## 8  Carbendazim (MBC)                7        0.0110 MD1811280042SZWA1P FALSE
## 9  Carbendazim (MBC)                7        0.0012 MD1811280057SZWA1P FALSE
## 10 Carbendazim (MBC)                7        0.0200  MI1810230005SZWA1 FALSE
## 11 Carbendazim (MBC)                7        0.0200  MI1810230010SZWA1 FALSE
## 12 Carbendazim (MBC)                7        0.0190  MI1811270006SZWA1 FALSE
## 13 Carbendazim (MBC)                7        0.0170  MI1811270010SZWA1 FALSE
## 14 Carbendazim (MBC)                7        0.0180  MI1812040005SZWA1 FALSE
## 15 Carbendazim (MBC)                7        0.0041  MI1812040053SZWA1 FALSE
## 16 Carbendazim (MBC)                7        0.0015 NC1810240537SZWA1B FALSE
## 17 Carbendazim (MBC)                7        0.0250 NC1812120510SZWA1A FALSE
## 18 Carbendazim (MBC)                7        0.0091 NY1812170249SZWA1P FALSE
## 19 Carbendazim (MBC)                7        0.0120 NY1812170265SZWA1P FALSE
## 20 Carbendazim (MBC)                7        0.0011 OH1812260203SZWA1P FALSE
## 21 Carbendazim (MBC)                7        0.0040 OH1812260207SZWA1P FALSE
## 22 Carbendazim (MBC)                7        0.0440  TX1811120103SZWA1 FALSE
```

```
max(carb$Concentration)
```

```
## [1] 0.044
```

Again we can see there are many detection events

Jon Neimann helped me with the final functions

https://pmc.ncbi.nlm.nih.gov/articles/PMC10217756/ https://www.sciencedirect.com/science/article/pii/S2352364615300055