Stawberries

MA615

2024-09-25

Preparing data for analysis

Introduction: foundations

Before we begin to work with the strawberry data, let's talk about how we will approach the work.

Data cleaning and organization

Cleaning and organizing data for analysis is an essential skill for data scientists. Serious data analyses must be presented with the data on which the results depend. The credibility of data analysis and modelling depends on the care taken in data preparation and organization.

References

In their handbook "An introduction to data cleaning with R" by Edwin de Jonge and Mark van der Loo, de Jonge and van der Loo go into detail about specific data cleaning isssues and how to handle them in R.

"Problems, Methods, and Challenges in Comprehensive Data Cleansing" by Heiko Müller and Johann-Christoph Freytag is a good companion to the de Jonge and van der Loo handbook, offering additional issues in their discussion.

Attitudes

Mechanistic descriptions of data cleaning methods are insufficient.

Data is the product (or by-product) of purposeful human activity

Much of the data used in analysis accessed on local databases or online which may create the impression that the data have been carefully curated. Beware. Data are produced by people for a purpose, with a point-of-view, and at a time and location that may affect the data. The provenance and lineage of the data are meta data you should include when reporting analysis. Data collection is purposeful human activity with all of the risks and weaknesses that are part of any purposeful human activity.

Data is language

Data has meaning. Data can be included in sentences related to the meaning of the data. Cleaning and organizing data should be informed by the meaning the data convey and how that meaning relates to the research you are doing do achieve this important result.

- Immerse yourself in the data. Put data into context.
- Visualize the data to find problems, confirm your understandings, and plan your data organization. People do a bad job of seeing meaningful patterns in data but a good job of seeing patterns of all kinds when data are rendered as plots. As you product and show visualizations, ask your self and those who view your presentations, "what do you see?" and "what do you wonder?"

Example: Strawberries

Public information

WHO says strawberries may not be so safe for you–2017March16

Pesticides + poison gases = cheap, year-round strawberries 2019March20

Multistate Outbreak of Hepatitis A Virus Infections Linked to Fresh Organic Strawberries-2022March5

Strawberry makes list of cancer-fighting foods-2023May31

What is the question?

- Where they are grown? By whom?
- Are they really loaded with carcinogenic poisons?
- Are they really good for your health? Bad for your health?
- Are organic strawberries carriers of deadly diseases?

- When I go to the Market should I buy conventional or organic strawberries?

The data

The data set for this assignment has been selected from: [USDA_NASS_strawb_2024SEP25 The data have been stored on NASS here: USDA_NASS_strawb_2024SEP25

and has been stored on the blackboard as strawberries25_v3.csv.

USDA NASS

```
library(knitr)
library(kableExtra)
library(tidyverse)
library(stringr)
library(magrittr)
```

Read the file

```
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.
```

```
#glimpse(strawberry)
```

Examine the data. How is it organized?

```r
## is every line associated with a state?

state_all <- strawberry |> distinct(State)

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

## every row is associated with a state

sum(state_all1$n) == dim(strawberry)[1]
```

```
[1] TRUE
```

```r
## to get an idea of the data -- looking at california only

calif_census <- strawberry |> filter((State=="CALIFORNIA") & (Program=="CENSUS"))

calif_census <- calif_census |> select(Year, `Data Item`, Value)

###

calif_survey <- strawberry |> filter((State=="CALIFORNIA") & (Program=="SURVEY"))

calif_survey <- strawberry |> select(Year, Period, `Data Item`, Value)
```

Remove columns with a single value in all columns and county in `Geo Level`

```r
strawberry <- drop_one_value_col(strawberry)

drop_one_value_col(strawberry)
```

```
# A tibble: 12,669 x 0
```

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

Separate strawberry data set into small data sets to understand the data better

We separated `Census` and `Survey` data from the `strawberry` data set in order to examine the data better. Furthermore, `Data Item` column was separated into two column : `Fruit` and `Category`by `-` .

Focusing on the census data first, `Fruit` is further divided into `ORGANIC` and `Organic detail` leading us to get `organic` data set from census.

```r
#|label: split strawberries into census(further organic) and survey data
#|echo: false
census <- strawberry |> filter(Program == "CENSUS")

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

survey <- survey |> drop_one_value_col()

census <- census |>
  separate_wider_delim(  cols = `Data Item`,
                         delim = " - ",
                         names = c("Fruit",
                                 "Category"),
                         too_many = "error",
                         too_few = "align_start"
                       )
census <- census |>
  separate_wider_delim(  cols = Fruit,
                         delim = ", ",
                         names = c("Fruit",
                                 "ORGANIC",
                                 "Organic detail"),

                         too_many = "error",
                         too_few = "align_start"
                       )

census <- census |> drop_one_value_col()
organic <- census |> filter(ORGANIC == "ORGANIC")
census <- census[(is.na(census$ORGANIC)),]
census <- census |> drop_one_value_col()
```

Split `Category` by `"  "` into `Measure` and `Bearing Type` and consequently, removing `WITH` from

Bearing Type.

```
census <- census |>
  separate_wider_delim(  cols = `Category`,
                         delim = " ",
                         names = c("Measure",
                                 "Bearing Type"),
                         too_many = "merge",
                         too_few = "align_start"
                       )

census$`Bearing Type` <- str_replace(census$`Bearing Type`, "WITH ", "")
```

Upon observing Domain Category as per instruction of assignment 1 in strawberry, we just rename Domain Category into size_bracket for the census as it is majorly having size brackets for Domain Category Column. Along with it, NOT SPECIFIED is renamed into TOTAL and AREA GROWN: is removed. This cleans up the census data.

```
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: ", "")
organic <- organic |> drop_one_value_col()
```

Similarly to census, Data Item is split into four columns Fruit,Category,Measure and Metric for survey data by ,. Fruit is further separated into Fruit and Applications

```
survey <- survey |>  separate_wider_delim(cols = `Data Item`,
                                                delim = ", ",
                                                names = c("Fruit",
                                                         "Category",
                                                         "Measure",
                                                         "Metric"
                                                         ),
                                                too_many = "merge",
                                                too_few = "align_start")


survey <- survey |> separate_wider_delim(cols = "Fruit",
                                                 delim = " - ",
                                                 names = c("Fruit",
                                                           "Application"),
                                                 too_many = "merge",
                                                 too_few = "align_start"
                                                 )
```

Fixing Misplaced Values

Using `shift_loc`, some values that are supposed to be in other column (here column to the right of `Application`) are searched in the `Application` and then shifted right to the expressed number rows away .

```
#|label: fix the misplaced values

survey %<>% shift_loc("Application", "PRICE RECEIVED", 2, 1 )

survey %<>% shift_loc("Application", "ACRES HARVESTED", 1, 1 )

survey %<>% shift_loc("Application", "ACRES PLANTED", 1, 1 )

survey %<>% shift_loc("Application", "PRODUCTION", 2, 1 )

survey %<>% shift_loc("Application", "YIELD", 2, 1 )

survey %<>% shift_loc("Application", "APPLICATIONS", 3, 1 )

survey %<>% shift_loc("Application", "TREATED", 3, 1 )

survey %<>% drop_one_value_col()
```

In `survey`, `Domain` is further separated into `Chemical` and `Type`. We then filter `TOTAL(survey_total)`, `CHEMICAL(survey_chem)` and `FERTILIZER(survey_chem)` data sets from the `survey`

```
survey <- survey |>
  separate_wider_delim(cols = Domain,
                      delim = ", ",
                      names = c("Chemical",
                                "Type"),

                      too_many = "merge",
                      too_few = "align_start")
survey_total <- survey |>  filter(Chemical == "TOTAL")
survey_chem <- survey |>  filter(Chemical== "CHEMICAL")
survey_fert <- survey |>  filter(Chemical == "FERTILIZER")
```

Similar to the logic we applied at `Application` , we apply it on the `Measure` as well to fill in the NAs in the right places. Further, `Category` is divided into `Market` and `Action`

```
survey_total %<>% drop_one_value_col()

### align terms

survey_total %<>% shift_loc("Measure", "MEASURED IN $ / CWT", 1, 1 )


survey_total %<>% shift_loc("Measure", "MEASURED IN $", 1, 1 )


survey_total %<>% shift_loc("Measure", "MEASURED IN CWT", 1, 1 )

survey_total %<>% shift_loc("Measure", "MEASURED IN TONS", 1, 1 )


survey_total %<>% shift_loc("Measure", "MEASURED IN CWT / ACRE", 1, 1 )

survey_total %<>% shift_loc("Measure", "MEASURED IN TONS / ACRE", 1, 1 )


survey_total <- survey_total |>
  separate_wider_delim(cols = Category,
                      delim = " - ",
                      names = c("Market",
                                "Action"),
                   too_many = "merge",
                    too_few = "align_start")
```

Shifting values from `Market` to the right places. This cleans up `survey_total`

```
survey_total %<>%
  select(-`State ANSI`)
survey_total <-  survey_total |>
```

```
  shift_loc("Market", "PRODUCTION", 2, 1)

survey_total <-  survey_total |>
  shift_loc("Market", "PRICE RECEIVED", 2, 1)
```

`Category` in `survey_chem` is divided into two categories namely `cat1` and `cat2`. Further due to repeating words, we remove `MEASURED IN` and `CHEMICAL` from the `Measure` and `Domain Category` respectively. We get `Chemical Name` from the `Domain Category` after seperating it into two and removing the first column. Punctuation signs are removed from the `Chemical Name` which we later divide into `Chemical Name` and `Code`. This cleans up `survey_chem`.

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

survey_chem <- survey_chem |> select(-`State ANSI`)

survey_chem <- survey_chem |>
  separate_wider_delim(cols = Category,
                       delim = " - ",
                       names = c("cat1",
                                 "cat2"),
                    too_many = "merge",
                     too_few = "align_start")
survey_chem$Measure <- str_replace(survey_chem$Measure, "MEASURED IN ", "")

survey_chem$`Domain Category` <- str_replace(survey_chem$`Domain Category`, "CHEMICAL, ", ""

survey_chem <- survey_chem |>
          separate_wider_delim(cols = `Domain Category`,
                               delim = ": ",
                               names = c("type",
                                 "Chemical Name"),
                            too_many = "merge",
                             too_few = "align_start")

survey_chem <- survey_chem |> select(-type)

survey_chem$`Chemical Name` <- str_replace(survey_chem$`Chemical Name`, "^\\(", "")

survey_chem$`Chemical Name` <- str_replace(survey_chem$`Chemical Name`, "\\)$", "")

survey_chem <- survey_chem |>
  separate_wider_delim(cols = `Chemical Name`,
                       delim = " = ",
                       names = c("Chemical Name",
                                 "Code"),
                    too_many = "error",
                     too_few = "align_start")
```

Now, we are to clean the `survey_fert` containing fertilizers data from the `survey`. Similarly to the `survey_chem`, we divide `Category` into two columns by `-` and then remove `MEASURED IN` and `CHEMICAL` from the `Domain Category`. Following the same routine, we clean up `survey_fert`.

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

survey_fert <- survey_fert |> select(-`State ANSI`)

survey_fert <- survey_fert |>
  separate_wider_delim(cols = Category,
                      delim = " - ",
                      names = c("cat1",
                                "cat2"),
                   too_many = "merge",
                    too_few = "align_start")

survey_fert$Measure <- str_replace(survey_fert$Measure, "MEASURED IN ", "")

survey_fert$`Domain Category` <- str_replace(survey_fert$`Domain Category`, "CHEMICAL, ", ""

survey_fert <- survey_fert |>
          separate_wider_delim(cols = `Domain Category`,
                              delim = ": ",
                              names = c("type",
                                "Chemical Name"),
                           too_many = "merge",
                            too_few = "align_start")

survey_fert$`Chemical Name` <- str_replace(survey_fert$`Chemical Name`, "^\\(", "")

survey_fert$`Chemical Name` <- str_replace(survey_fert$`Chemical Name`, "\\)$", "")

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

We convert `Value` column in `census` and `survey` into numeric.

```
census$Value <- as.numeric(str_replace(census$Value, ",", ""))
```

```
Warning: NAs introduced by coercion
```

```
organic$Value <- as.numeric(str_replace(organic$Value, ",", ""))
```

```
Warning: NAs introduced by coercion
```

Imputation

We observe that in the `Bearing Type` we have some categories such as `BEARING`, `NON BEARING` and `GROWN`. Corresponding to the `TOTAL` value in the `Domain` and `size_bracket` , we can see the sum of all the previous value in the same category in `Value` Column. Using this logic, we get the function (with the help of chatgpt) `impute_values`. This will impute 0 where there are no previous values to apply the logic for `TOTAL` (these are very less cases). Similarly for more than one NA values in the same category. we get the remainder from subtracting sum of non-NA values from `TOTAL` to distribute remainder equally into NA values.

```
library(dplyr)

impute_values <- function(df) {
  # Create a copy of the original data frame
  original_df <- df

  # Group by the relevant columns without changing the order
  df <- df %>%
    group_by(State, Measure, `Bearing Type`) %>%
    mutate(
      # Get the total value for the group
      total_value = Value[size_bracket == "TOTAL"],
      # Sum non-total values
      sum_non_total = sum(Value[size_bracket != "TOTAL"], na.rm = TRUE),
      # Count the number of NAs in the non-total values
      na_count = sum(is.na(Value[size_bracket != "TOTAL"])),
      # Fill NAs in non-total rows if there's a total value
      Value = ifelse(
        is.na(Value) & size_bracket != "TOTAL" & !is.na(total_value),
        round((total_value - sum_non_total) / na_count, 2),
        Value
      ),
      # Fill TOTAL if it is NA and non-total values are available
      Value = ifelse(
        size_bracket == "TOTAL" & is.na(Value),
        round(sum_non_total, 2),
        Value
      )
    ) %>%
    ungroup() %>%
    select(-total_value, -sum_non_total, -na_count) # Clean up intermediate columns

  # Format Value column to two decimal places
  original_df$Value <- round(df$Value, 2)

  return(original_df)
}

census <- impute_values(census)
```

There are only two states in this data set that are having chemical data i.e. `CALIFORNIA` and `FLORIDA`.We start by the seeing how many chemicals are there in the `survey_chem` (175).

```
unique(survey_chem$State)
```

```
[1] "CALIFORNIA" "FLORIDA"
```

```
chemical_counts <- survey_chem %>%
  group_by(`Chemical Name`) %>%
  summarise(case_count = n()) %>%
  arrange(desc(case_count))

# View the result
print(chemical_counts)
```

```
# A tibble: 175 x 2
   `Chemical Name`     case_count
   <chr>                    <int>
 1 TOTAL                       64
 2 ABAMECTIN                   40
 3 ACETAMIPRID                 40
 4 AZOXYSTROBIN                40
 5 BIFENAZATE                  40
 6 BIFENTHRIN                  40
 7 CAPTAN                      40
 8 CHLORANTRANILIPROLE         40
 9 CYPRODINIL                  40
10 DIFENOCONAZOLE              40
# i 165 more rows
```

Good Chemicals

1. **Neem Oil** (NEEM OIL, NEEM OIL, CLAR. HYD.)
2. **Garlic Oil** (GARLIC OIL)
3. **Canola Oil** (CANOLA OIL)
4. **Sulfur** (SULFUR)
5. **Bacillus Subtilis** (BACILLUS SUBTILIS)
6. **Beauveria Bassiana** (BEAUVERIA BASSIANA)
7. **Trichoderma Harzianum** (TRICHODERMA HARZ.)
8. **Aureobasidium Pullulans** (AUREOBASIDIUM PULLULANS DSM 14940, AUREOBASIDIUM PULLULANS DSM 14941)
9. **Hydrogen Peroxide** (HYDROGEN PEROXIDE)
10. **Mustard Oil** (MUSTARD OIL)

Bad Chemicals

1. **Glyphosate** (GLYPHOSATE ISO. SALT, GLYPHOSATE POT. SALT)
2. **Malathion** (MALATHION)
3. **Chlorpyrifos** (CHLORPYRIFOS)
4. **Paraquat** (PARAQUAT)
5. **Carbaryl** (CARBARYL)
6. **Imidacloprid** (IMIDACLOPRID)
7. **Bifenthrin** (BIFENTHRIN)
8. **Permethrin** (PERMETHRIN)

9. **Thiamethoxam** (THIAMETHOXAM)
10. **Mustard Oil** (MUSTARD OIL)

We check these oils as to which state are they in? Surprisingly they are in `CALIFORNIA`

```
neem_oil_states <- survey_chem %>%
  filter(`Chemical Name` == "NEEM OIL" | `Chemical Name` == "NEEM OIL, CLAR. HYD." ) %>%
  select(State) %>%
  distinct()  # Get distinct states to avoid duplicates

neem_oil_states
```

```
# A tibble: 1 x 1
  State
  <chr>
1 CALIFORNIA
```

```
garlic_oil_states <- survey_chem %>%
  filter(`Chemical Name` == "GARLIC OIL") %>%
  select(State) %>%
  distinct()  # Get distinct states to avoid duplicates

garlic_oil_states
```

```
# A tibble: 1 x 1
  State
  <chr>
1 CALIFORNIA
```

This leads me to do further exploration on the number of cases per chemical for the both states. I see Oils used in `California` as one of the chemicals having `Major` cases i.e. 20

```
# Assuming 'survey_chem' is your data frame and has a column for 'State' and 'Chemical Name'
california_chemicals <- survey_chem[survey_chem$State == "CALIFORNIA", ]

# Display the unique chemicals used in California
unique_california_chemicals <- unique(california_chemicals$`Chemical Name`)

# Count the occurrences of each chemical in California
california_counts <- table(california_chemicals$`Chemical Name`)
```

```r
# Convert the table to a data frame
california_counts_df <- as.data.frame(california_counts)

# Rename the columns for clarity
colnames(california_counts_df) <- c("Chemical", "Count")
california_counts_df <- california_counts_df[order(-california_counts_df$Count), ]
row.names(california_counts_df) <- NULL
# Print the results
print(california_counts_df)
```

```
                                             Chemical Count
1                                               TOTAL    32
2                                           ABAMECTIN    20
3                                         ACEQUINOCYL    20
4                                         ACETAMIPRID    20
5                                ACIBENZOLAR-S-METHYL    20
6                                        AZADIRACHTIN    20
7                                        AZOXYSTROBIN    20
8              BACILLUS AMYLOLIQUEFACIENS STRAIN D747    20
9                                   BACILLUS SUBTILIS    20
10                                         BIFENAZATE    20
11                                         BIFENTHRIN    20
12                                               BLAD    20
13                                  BORAX DECAHYDRATE    20
14                                           BOSCALID    20
15                               BT KURSTAK ABTS-1857    20
16                               BT KURSTAKI ABTS-351    20
17                                  BT KURSTAKI SA-11    20
18                                             CAPTAN    20
19                                CHLORANTRANILIPROLE    20
20                                       CHLOROPICRIN    20
21  CHROMOBAC SUBTSUGAE PRAA4-1 CELLS AND SPENT MEDIA    20
22                                   CYANTRANILIPROLE    20
23                                       CYFLUFENAMID    20
24                                         CYPRODINIL    20
25                                    DICHLOROPROPENE    20
26                                     DIFENOCONAZOLE    20
27                                          ETOXAZOLE    20
28                                   FENBUTATIN-OXIDE    20
29                                         FENHEXAMID    20
30                                      FENPROPATHRIN    20
```

31	FENPYROXIMATE	20
32	FLONICAMID	20
33	FLUDIOXONIL	20
34	FLUMIOXAZIN	20
35	FLUOPYRAM	20
36	FLUPYRADIFURONE	20
37	FLUTRIAFOL	20
38	FLUXAPYROXAD	20
39	FOSETYL-AL	20
40	HEXYTHIAZOX	20
41	IMIDACLOPRID	20
42	IRON PHOSPHATE	20
43	ISOFETAMID	20
44	MALATHION	20
45	MEFENOXAM	20
46	METAM-POTASSIUM	20
47	METHOXYFENOZIDE	20
48	MYCLOBUTANIL	20
49	NALED	20
50	NEEM OIL	20
51	NEEM OIL, CLAR. HYD.	20
52	NOVALURON	20
53	OXYFLUORFEN	20
54	PENDIMETHALIN	20
55	PENTHIOPYRAD	20
56	PIPERONYL BUTOXIDE	20
57	POLYOXIN D ZINC SALT	20
58	PROPICONAZOLE	20
59	PYRACLOSTROBIN	20
60	PYRETHRINS	20
61	PYRIMETHANIL	20
62	QUINOLINE	20
63	REYNOUTRIA SACHALINE	20
64	SPINETORAM	20
65	SPINOSAD	20
66	SULFUR	20
67	TETRACONAZOLE	20
68	THIAMETHOXAM	20
69	THIOPHANATE-METHYL	20
70	THIRAM	20
71	TRIFLOXYSTROBIN	20
72	TRIFLUMIZOLE	20
73	BACILLUS AMYLOLIQUEFACIENS MBI 600	15

74	BACILLUS PUMILUS	15
75	BEAUVERIA BASSIANA	15
76	BT SUB AIZAWAI GC-91	15
77	BT SUBSP KURSTAKI EVB-113-19	15
78	BUPROFEZIN	15
79	BURKHOLDERIA A396 CELLS & MEDIA	15
80	CAPRIC ACID	15
81	CAPRYLIC ACID	15
82	CARFENTRAZONE-ETHYL	15
83	COPPER OCTANOATE	15
84	CYFLUMETOFEN	15
85	GLYPHOSATE ISO. SALT	15
86	GLYPHOSATE POT. SALT	15
87	HYDROGEN PEROXIDE	15
88	METALDEHYDE	15
89	METAM-SODIUM	15
90	MONO-POTASSIUM SALT	15
91	NAPROPAMIDE	15
92	PAECILOMYCES FUMOSOR	15
93	PEROXYACETIC ACID	15
94	POTASSIUM BICARBON.	15
95	POTASSIUM SALTS	15
96	POTASSIUM SILICATE	15
97	PYRIDABEN	15
98	PYRIPROXYFEN	15
99	SPIROMESIFEN	15
100	STREPTOMYCES LYDICUS	15
101	AUREOBASIDIUM PULLULANS DSM 14940	10
102	AUREOBASIDIUM PULLULANS DSM 14941	10
103	BT KURSTAKI SA-12	10
104	CANOLA OIL	10
105	CAPSICUM OLEORESIN EXTRACT	10
106	CARBARYL	10
107	CHLORPYRIFOS	10
108	COPPER HYDROXIDE	10
109	DIAZINON	10
110	GARLIC OIL	10
111	GLIOCLADIUM VIRENS	10
112	HELICOVERPA ZEA NPV	10
113	LAMBDA-CYHALOTHRIN	10
114	PARAQUAT	10
115	PSEUDOMONAS CHLORORAPHIS STRAIN AFS009	10
116	SULFENTRAZONE	10

```
117                                       SULFOXAFLOR    10
118                       BACILLUS AMYLOLIQUEFAC F727     5
119                               BACILLUS SUBT. GB03     5
120                                BT KURSTAKI EG7841     5
121                                    CYCLANILIPROLE     5
122                             CYFLUMETOFEN = 138831     5
123                                EMAMECTIN BENZOATE     5
124                              GLUFOSINATE-AMMONIUM     5
125                                         IPRODIONE     5
126                 ISARIA FUMOSOROSEA STRAIN FE 9901     5
127                                       MINERAL OIL     5
128                                   OXATHIAPIPROLIN     5
129                                        PERMETHRIN     5
130                              PETROLEUM DISTILLATE     5
131                                    PYDIFLUMETOFEN     5
132                                       SOYBEAN OIL     5
133                                     SPIROTETRAMAT     5
134                                 TRICHODERMA HARZ.     5
135                    TRICHODERMA VIRENS STRAIN G-41     5
136                                 ZETA-CYPERMETHRIN     5
```

For the Florida, I observe that they are using Mustard oil in minority but still it is being used.

```r
# Assuming 'survey_chem' is your data frame and has a column for 'State' and 'Chemical Name'
florida_chemicals <- survey_chem[survey_chem$State == "FLORIDA", ]

# Display the unique chemicals used in California
unique_florida_chemicals <- unique(florida_chemicals$`Chemical Name`)

# Count the occurrences of each chemical in California
florida_counts <- table(florida_chemicals$`Chemical Name`)

# Convert the table to a data frame
florida_counts_df <- as.data.frame(florida_counts)

# Rename the columns for clarity
colnames(florida_counts_df) <- c("Chemical", "Count")
florida_counts_df <- florida_counts_df[order(-florida_counts_df$Count), ]
row.names(florida_counts_df) <- NULL
# Print the results
print(florida_counts_df)
```

```
                                  Chemical Count
1                                    TOTAL    32
2                                ABAMECTIN    20
3                              ACETAMIPRID    20
4                             AZOXYSTROBIN    20
5                               BIFENAZATE    20
6                               BIFENTHRIN    20
7                                   CAPTAN    20
8                      CHLORANTRANILIPROLE    20
9                               CYPRODINIL    20
10                          DIFENOCONAZOLE    20
11                              FENHEXAMID    20
12                             FLUDIOXONIL    20
13                    GLYPHOSATE ISO. SALT    20
14                               MALATHION    20
15                               MEFENOXAM    20
16                                   NALED    20
17                               NOVALURON    20
18                            PYRIMETHANIL    20
19                              SPINETORAM    20
20                            THIAMETHOXAM    20
21                      THIOPHANATE-METHYL    20
22                                  THIRAM    20
23                    BT KURSTAKI ABTS-351    15
24                    COPPER CHLORIDE HYD.    15
25                        COPPER HYDROXIDE    15
26                        CYANTRANILIPROLE    15
27                            CYFLUFENAMID    15
28                              CYTOKININS    15
29                             FLUMIOXAZIN    15
30                               FLUOPYRAM    15
31                              FOSETYL-AL    15
32                            IMIDACLOPRID    15
33                              ISOFETAMID    15
34                     MONO-POTASSIUM SALT    15
35                                PARAQUAT    15
36                            PENTHIOPYRAD    15
37                           PROPICONAZOLE    15
38                          PYRACLOSTROBIN    15
39                            SPIROMESIFEN    15
40                                  SULFUR    15
41                           TETRACONAZOLE    15
42                            TRIFLUMIZOLE    15
```

43	2,4-D, DIMETH. SALT	10
44	ACIBENZOLAR-S-METHYL	10
45	BACILLUS SUBTILIS	10
46	BOSCALID	10
47	CARFENTRAZONE-ETHYL	10
48	CLETHODIM	10
49	CYFLUMETOFEN	10
50	DIAZINON	10
51	FENPYROXIMATE	10
52	FLUTRIAFOL	10
53	FLUXAPYROXAD	10
54	HEXYTHIAZOX	10
55	INDOLEBUTYRIC ACID	10
56	IPRODIONE	10
57	METAM-POTASSIUM	10
58	METHOXYFENOZIDE	10
59	PSEUDOMONAS CHLORORAPHIS STRAIN AFS009	10
60	PYDIFLUMETOFEN	10
61	REYNOUTRIA SACHALINE	10
62	SULFOXAFLOR	10
63	2,4-D, TRIISO. SALT	5
64	ALKYL. DIM. BENZ. AM	5
65	BACILLUS AMYLOLIQUEFAC F727	5
66	BETA-CYFLUTHRIN	5
67	BORAX DECAHYDRATE	5
68	BT KURSTAK ABTS-1857	5
69	CARBARYL	5
70	CHLOROPICRIN	5
71	CHLOROTHALONIL	5
72	COPPER ETHANOLAMINE	5
73	CUPRAMMONIUM ACETATE	5
74	CYMOXANIL	5
75	CYPERMETHRIN	5
76	DECYLDIMETHYLOCTYL	5
77	DICHLOROPROPENE	5
78	DIDECYL DIM. AMMON.	5
79	DIMETHENAMID	5
80	DIMETHYL DISULFIDE (DMDS)	5
81	DIMETHYLDIOCTYL	5
82	DODECADIEN-1-OL	5
83	DODINE	5
84	ETHEPHON	5
85	ETHYL (2E;4Z)-DECADIENOATE	5

```
86                                    FAMOXADONE     5
87                                    FENAZAQUIN     5
88                                 FENPROPATHRIN     5
89                                    FLONICAMID     5
90                                   FLUENSULFONE    5
91                                FLUPYRADIFURONE    5
92                               FLUROXYPYR 1-MHE    5
93                                    FLUTOLANIL     5
94                               GIBBERELLIC ACID    5
95                           GLYPHOSATE POT. SALT    5
96                             HALOSULFURON-METHYL   5
97                               HYDROGEN PEROXIDE   5
98                                        KANTOR     5
99                              LAMBDA-CYHALOTHRIN   5
100                                     MANCOZEB     5
101                                     METHOMYL     5
102                             METSULFURON-METHYL   5
103                                  MUSTARD OIL     5
104                                  MYCLOBUTANIL    5
105                                  NAPROPAMIDE     5
106                                       OXAMYL     5
107                                OXATHIAPIPROLIN   5
108                                  OXYFLUORFEN     5
109                                   PENOXSULAM     5
110                               PEROXYACETIC ACID  5
111                             PIPERONYL BUTOXIDE   5
112                                   PYRETHRINS     5
113                                  PYRIOFENONE     5
114                                 S-METOLACHLOR    5
115                                     SPINOSAD     5
116                                     ZOXAMIDE     5
```

I check how many cases for good chemicals (as in less toxicity or considered more beneficial than harmful) are there in both the states.

```
good_chemicals <- c("NEEM OIL", "GARLIC OIL", "CANOLA OIL", "SULFUR",
                    "BACILLUS SUBTILIS", "BEAUVERIA BASSIANA",
                    "TRICHODERMA HARZ.", "AUREOBASIDIUM PULLULANS DSM 14940",
                    "AUREOBASIDIUM PULLULANS DSM 14941", "HYDROGEN PEROXIDE", "MUSTARD OIL")

# Filter for good chemicals used in California
california_good_chemicals <- california_counts_df[california_counts_df$`Chemical` %in% good_c
```

```
row.names(california_good_chemicals) <- NULL
# Print the results
print(california_good_chemicals)
```

```
                           Chemical Count
1                 BACILLUS SUBTILIS    20
2                          NEEM OIL    20
3                            SULFUR    20
4                BEAUVERIA BASSIANA    15
5                 HYDROGEN PEROXIDE    15
6  AUREOBASIDIUM PULLULANS DSM 14940    10
7  AUREOBASIDIUM PULLULANS DSM 14941    10
8                        CANOLA OIL    10
9                        GARLIC OIL    10
10                TRICHODERMA HARZ.     5
```

```
florida_good_chemicals <- florida_counts_df[florida_counts_df$`Chemical` %in% good_chemicals
row.names(florida_good_chemicals) <- NULL
# Print the results
print(florida_good_chemicals)
```

```
           Chemical Count
1            SULFUR    15
2 BACILLUS SUBTILIS    10
3 HYDROGEN PEROXIDE     5
4       MUSTARD OIL     5
```

For `CALIFORNIA`, I see out of 10 good chemicals chatgpt pointed out, there are 9 being used. Whereas for Florida, only 4 out of 10 are being used much to the disappointment but then again there are 175, I assume there would be more good chemicals that I am not observing being used in Florida. For the bad chemicals, both the states uses about 6 out of 10 I am observing.

```
# Define bad chemicals
bad_chemicals <- c("BACILLUS THURINGIENSIS", "BIFENTHRIN", "CHLORPYRIFOS",
                   "DIAZINON", "FIPRONIL", "IMIDACLOPRID",
                   "MALATHION", "METOLACHLOR", "PERMETHRIN",
                   "PENOXSULAM", "OXAMYL", "GLYPHOSATE")

# Filter for bad chemicals used in California
```

```
california_bad_chemicals <- california_counts_df[california_counts_df$`Chemical` %in% bad_che
row.names(california_bad_chemicals) <- NULL
# Print the results for California
print(california_bad_chemicals)
```

```
      Chemical Count
1   BIFENTHRIN    20
2 IMIDACLOPRID    20
3    MALATHION    20
4 CHLORPYRIFOS    10
5     DIAZINON    10
6   PERMETHRIN     5
```

```
# Filter for bad chemicals used in Florida
florida_bad_chemicals <- florida_counts_df[florida_counts_df$`Chemical` %in% bad_chemicals,
row.names(florida_bad_chemicals) <- NULL
# Print the results for Florida
print(florida_bad_chemicals)
```

```
      Chemical Count
1   BIFENTHRIN    20
2    MALATHION    20
3 IMIDACLOPRID    15
4     DIAZINON    10
5       OXAMYL     5
6   PENOXSULAM     5
```

Split Sales , Chemicals, Organic and Non-Organic into Different Dataframes

Writing Code into different CSV Files

```
write.csv(strawberry, file = "strawberry_cleaneddata.csv")
write.csv(census, file = "census_data.csv")
write.csv(survey, file = "survey_data.csv")
write.csv(organic, file = "organic.csv")
#write.csv(census_non_organic, file = "census_non_organic.csv")
write.csv(survey_chem, file = "survey_chemical.csv")
write.csv(survey_total, file = "survey_total.csv")
write.csv(survey_fert, file = "survey_fert.csv")
```

Questions after EDA

1. Is there any connection between majority of Indian being in `California` and Oils being used as pesticides/fungicides especially Neem Oil?
2. Why is Mustard Oil not used in the `California`?

Study Material

epa numbers

Active Pesticide Product Registration Informational Listing

CAS for Methyl Bromide

pesticide chemical search

toxic chemical dashboard

pubChem

The EPA PC (Pesticide Chemical) Code is a unique chemical code number assigned by the EPA to a particular pesticide active ingredient, inert ingredient or mixture of active ingredients.

Investigating toxic pesticides

start here with chem PC code

step 2 to get label (with warnings) for products using the chemical

Pesticide Product and Label System

Search by Chemical

CompTox Chemicals Dashboard

Active Pesticide Product Registration Informational Listing

OSHA chemical database

Pesticide Ingredients

NPIC Product Research Online (NPRO)

Databases for Chemical Information

Pesticide Active Ingredients

TSCA Chemical Substance Inventory

glyphosate