chemical EDA

Zilu Sun

2024-10-28

- Some parts of this assignment's code have sought help from ChatGPT.

read and explore the data

```
library(knitr)
library(kableExtra)
library(tidyverse)
```

Read the data and take a first look

```
strawberry <- read.csv("strawberries25_v3.csv",header = TRUE)

unique(strawberry$Data.Item)
```

```
##  [1] "STRAWBERRIES - ACRES BEARING"
##  [2] "STRAWBERRIES - ACRES GROWN"
##  [3] "STRAWBERRIES - ACRES NON-BEARING"
##  [4] "STRAWBERRIES - OPERATIONS WITH AREA BEARING"
##  [5] "STRAWBERRIES - OPERATIONS WITH AREA GROWN"
##  [6] "STRAWBERRIES - OPERATIONS WITH AREA NON-BEARING"
##  [7] "STRAWBERRIES, ORGANIC - ACRES HARVESTED"
##  [8] "STRAWBERRIES, ORGANIC - OPERATIONS WITH AREA HARVESTED"
##  [9] "STRAWBERRIES, ORGANIC - OPERATIONS WITH SALES"
## [10] "STRAWBERRIES, ORGANIC - PRODUCTION, MEASURED IN CWT"
## [11] "STRAWBERRIES, ORGANIC - SALES, MEASURED IN $"
## [12] "STRAWBERRIES, ORGANIC - SALES, MEASURED IN CWT"
## [13] "STRAWBERRIES, ORGANIC, FRESH MARKET - OPERATIONS WITH SALES"
## [14] "STRAWBERRIES, ORGANIC, FRESH MARKET - SALES, MEASURED IN $"
## [15] "STRAWBERRIES, ORGANIC, FRESH MARKET - SALES, MEASURED IN CWT"
## [16] "STRAWBERRIES, ORGANIC, PROCESSING - OPERATIONS WITH SALES"
## [17] "STRAWBERRIES, ORGANIC, PROCESSING - SALES, MEASURED IN $"
## [18] "STRAWBERRIES, ORGANIC, PROCESSING - SALES, MEASURED IN CWT"
## [19] "STRAWBERRIES, FRESH MARKET - PRICE RECEIVED, ADJUSTED BASE, MEASURED IN $ / CWT"
## [20] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, ADJUSTED BASE, MEASURED IN $ / TON"
## [21] "STRAWBERRIES - PRICE RECEIVED, MEASURED IN $ / CWT"
## [22] "STRAWBERRIES, FRESH MARKET - PRICE RECEIVED, MEASURED IN $ / CWT"
## [23] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, MEASURED IN $ / CWT"
## [24] "STRAWBERRIES - ACRES HARVESTED"
## [25] "STRAWBERRIES - ACRES PLANTED"
```

```
## [26] "STRAWBERRIES - PRODUCTION, MEASURED IN $"
## [27] "STRAWBERRIES - PRODUCTION, MEASURED IN CWT"
## [28] "STRAWBERRIES - PRODUCTION, MEASURED IN TONS"
## [29] "STRAWBERRIES - YIELD, MEASURED IN CWT / ACRE"
## [30] "STRAWBERRIES - YIELD, MEASURED IN TONS / ACRE"
## [31] "STRAWBERRIES, FRESH MARKET - PRICE RECEIVED, 10 YEAR AVG FOR PARITY PURPOSES, MEASURED IN $ / C
## [32] "STRAWBERRIES, FRESH MARKET - PRICE RECEIVED, 10 YEAR AVG, MEASURED IN $ / CWT"
## [33] "STRAWBERRIES, FRESH MARKET - PRODUCTION, MEASURED IN $"
## [34] "STRAWBERRIES, FRESH MARKET, UTILIZED - PRODUCTION, MEASURED IN CWT"
## [35] "STRAWBERRIES, NOT SOLD - PRODUCTION, MEASURED IN CWT"
## [36] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, 10 YEAR AVG FOR PARITY PURPOSES, MEASURED IN $ / TO
## [37] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, 10 YEAR AVG, MEASURED IN $ / TON"
## [38] "STRAWBERRIES, PROCESSING - PRODUCTION, MEASURED IN $"
## [39] "STRAWBERRIES, PROCESSING, UTILIZED - PRODUCTION, MEASURED IN CWT"
## [40] "STRAWBERRIES, UTILIZED - PRODUCTION, MEASURED IN CWT"
## [41] "STRAWBERRIES, UTILIZED - PRODUCTION, MEASURED IN TONS"
## [42] "STRAWBERRIES - APPLICATIONS, MEASURED IN LB"
## [43] "STRAWBERRIES - APPLICATIONS, MEASURED IN LB / ACRE / APPLICATION, AVG"
## [44] "STRAWBERRIES - APPLICATIONS, MEASURED IN LB / ACRE / YEAR, AVG"
## [45] "STRAWBERRIES - APPLICATIONS, MEASURED IN NUMBER, AVG"
## [46] "STRAWBERRIES - TREATED, MEASURED IN PCT OF AREA BEARING, AVG"
## [47] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB"
## [48] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB / ACRE / APPLICATION, AVG"
## [49] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB / ACRE / YEAR, AVG"
## [50] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN NUMBER, AVG"
## [51] "STRAWBERRIES, BEARING - TREATED, MEASURED IN PCT OF AREA BEARING, AVG"
## [52] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, MEASURED IN $ / TON"
## [53] "STRAWBERRIES, PROCESSING, UTILIZED - PRODUCTION, MEASURED IN TONS"
```

```
# Replace "(D)", "(L)", "(NA)" with NA
strawberry <- strawberry |>
  mutate(Value = ifelse(Value %in% c("(D)", "(L)", "(NA)"), NA, Value),
         )
strawberry$Value <- as.numeric(str_replace(strawberry$Value, ",", ""))
```

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

```
# View the processed data
glimpse(strawberry)
```

```
## Rows: 12,669
## Columns: 21
## $ Program          <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CE~
## $ Year             <int> 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022,~
## $ Period           <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR~
## $ Week.Ending      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Geo.Level        <chr> "COUNTY", "COUNTY", "COUNTY", "COUNTY", "COUNTY", "CO~
## $ State            <chr> "ALABAMA", "ALABAMA", "ALABAMA", "ALABAMA", "ALABAMA"~
## $ State.ANSI       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ Ag.District      <chr> "BLACK BELT", "BLACK BELT", "BLACK BELT", "BLACK BELT~
## $ Ag.District.Code <int> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 4~
## $ County           <chr> "BULLOCK", "BULLOCK", "BULLOCK", "BULLOCK", "BULLOCK"~
## $ County.ANSI      <int> 11, 11, 11, 11, 11, 11, 101, 101, 101, 101, 119, 119,~
```

```
## $ Zip.Code          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Region            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ watershed_code    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Watershed         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Commodity         <chr> "STRAWBERRIES", "STRAWBERRIES", "STRAWBERRIES", "STRA~
## $ Data.Item         <chr> "STRAWBERRIES - ACRES BEARING", "STRAWBERRIES - ACRES~
## $ Domain            <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL",~
## $ Domain.Category   <chr> "NOT SPECIFIED", "NOT SPECIFIED", "NOT SPECIFIED", "N~
## $ Value             <dbl> NA, 3, NA, 1, 6, 5, NA, NA, 2, 2, NA, NA, 2, 2, 1, 1,~
## $ CV....            <chr> "(D)", "15.7", "(D)", "(L)", "52.7", "47.6", "(D)", "~
```

```
# Create two new datasets: CENSUS and SURVEY
census_data <- strawberry |> filter(Program == "CENSUS")
survey_data <- strawberry |> filter(Program == "SURVEY")

# Filter organic data
organic_data <- census_data |>
  filter(str_detect(Data.Item, "ORGANIC"))

# Filter non-organic data
non_organic_data <- census_data |>
  filter(!str_detect(Data.Item, "ORGANIC"))

# Split the 'Data Item' column for non-organic data
non_organic_data <- non_organic_data |>
  separate_wider_delim(  cols = Data.Item,
                         delim = "-",
                         names = c("Fruit", "Category"),
                         too_many = "merge",
                         too_few = "align_start"
                       )
unique(non_organic_data$Category)
```

```
## [1] " ACRES BEARING"                    " ACRES GROWN"
## [3] " ACRES NON-BEARING"                " OPERATIONS WITH AREA BEARING"
## [5] " OPERATIONS WITH AREA GROWN"       " OPERATIONS WITH AREA NON-BEARING"
```

```
# Split the 'Data Item' column for organic data (unsure how to proceed here)
# Step 1: Split by "," delimiter
organic_data <- organic_data |>
  separate_wider_delim(cols = Data.Item,
                       delim = ",",
                       names = c("Fruit_Type", "Sales", "Others"),
                       too_many = "merge",  # Merge the extra parts
                       too_few = "align_start")

# Step 2: Further split the 'Fruit_Type' column by "-" delimiter
organic_data <- organic_data |>
  separate_wider_delim(cols = `Fruit_Type`,
                       delim = "-",
                       names = c("Fruit", "Type"),
                       too_many = "merge",
                       too_few = "align_start")
```

```r
# Filter chemical-free data
# Filter out non-chemical data (without chemical substances)
non_chemical_data <- survey_data |>
  filter(!str_detect(Domain, "CHEMICAL"))

# Filter out chemical data (with chemical substances)
chemical_data <- survey_data |>
  filter(str_detect(Domain, "CHEMICAL"))

# Split the 'Domain Category' column into 'type', 'name', and 'code'
chemical_data <- chemical_data |>
  # Step 1: Split by comma to get 'type' and remaining part
  separate_wider_delim(cols = Domain.Category,
                       delim = ",",
                       names = c("type", "Rest"),
                       too_many = "merge",  # Merge the extra parts
                       too_few = "align_start") |>
  # Step 2: Split the remaining part by colon to get 'name' and 'code'
  separate_wider_delim(cols = Rest,
                       delim = ":",
                       names = c("name", "code"),
                       too_many = "merge",  # Merge the extra parts
                       too_few = "align_start") |>
  # Trim the extra spaces
  mutate(across(c(type, name, code), ~ str_trim(.)))
```

```r
# Filter the data for California based on the 'State' column
CA_data <- non_organic_data |>
  filter(State == "CALIFORNIA")
# Find NA in the 'Value' column and fill the first 288 rows with 4
CA_data <- CA_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 288, 4, Value))
# Find NA in the 'Value' column and fill the first 285 rows with 4
CA_data <- CA_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 285, 4, Value))

# Filter the data for Florida and save it as FL_data
FL_data <- non_organic_data |>
  filter(State == "FLORIDA")
# Find NA in the 'Value' column and fill the first 134 rows with 3489
FL_data <- FL_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 134, 3489, Value))
# Find NA in the 'Value' column and fill the first 135 rows with 3489
FL_data <- FL_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 135, 3489, Value))
# Find NA in the 'Value' column and fill the first 136 rows with 3489
FL_data <- FL_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 136, 3489, Value))
# Find NA in the 'Value' column and fill the first 149 rows with 43
FL_data <- FL_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 149, 43, Value))
# Find NA in the 'Value' column and fill the first 150 rows with 43
FL_data <- FL_data |>
```

```
  mutate(Value = ifelse(is.na(Value) & row_number() <= 150, 43, Value))
# Find NA in the 'Value' column and fill the first 151 rows with 43
FL_data <- FL_data |>
  mutate(Value = ifelse(is.na(Value) & row_number() <= 151, 43, Value))
write.csv(chemical_data, "chemical_data_clean.csv", row.names = FALSE)
```

```
library(dplyr)
library(ggplot2)
```

#CA data EDA

```
chemical <- read.csv("chemical_data_clean.csv")
# View the structure and column names of the data
str(chemical)
```

```
## 'data.frame':    3359 obs. of  23 variables:
##  $ Program         : chr  "SURVEY" "SURVEY" "SURVEY" "SURVEY" ...
##  $ Year            : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ Period          : chr  "YEAR" "YEAR" "YEAR" "YEAR" ...
##  $ Week.Ending     : logi  NA NA NA NA NA NA ...
##  $ Geo.Level       : chr  "STATE" "STATE" "STATE" "STATE" ...
##  $ State           : chr  "CALIFORNIA" "CALIFORNIA" "CALIFORNIA" "CALIFORNIA" ...
##  $ State.ANSI      : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ Ag.District     : logi  NA NA NA NA NA NA ...
##  $ Ag.District.Code: logi  NA NA NA NA NA NA ...
##  $ County          : logi  NA NA NA NA NA NA ...
##  $ County.ANSI     : logi  NA NA NA NA NA NA ...
##  $ Zip.Code        : logi  NA NA NA NA NA NA ...
##  $ Region          : logi  NA NA NA NA NA NA ...
##  $ watershed_code  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Watershed       : logi  NA NA NA NA NA NA ...
##  $ Commodity       : chr  "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" ...
##  $ Data.Item       : chr  "STRAWBERRIES - APPLICATIONS, MEASURED IN LB" "STRAWBERRIES - APPLICATIONS
##  $ Domain          : chr  "CHEMICAL, FUNGICIDE" "CHEMICAL, INSECTICIDE" "CHEMICAL, INSECTICIDE" "CHE
##  $ type            : chr  "CHEMICAL" "CHEMICAL" "CHEMICAL" "CHEMICAL" ...
##  $ name            : chr  "FUNGICIDE" "INSECTICIDE" "INSECTICIDE" "OTHER" ...
##  $ code            : chr  "(OXATHIAPIPROLIN = 128111)" "(CYCLANILIPROLE = 26202)" "(PERMETHRIN = 109
##  $ Value           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ CV....          : logi  NA NA NA NA NA NA ...
```

```
head(chemical)
```

```
##   Program Year Period Week.Ending Geo.Level      State State.ANSI Ag.District
## 1  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
## 2  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
## 3  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
## 4  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
## 5  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
## 6  SURVEY 2023   YEAR          NA     STATE CALIFORNIA          6          NA
##   Ag.District.Code County County.ANSI Zip.Code Region watershed_code Watershed
## 1               NA     NA          NA       NA     NA              0        NA
## 2               NA     NA          NA       NA     NA              0        NA
```

```
## 3               NA     NA          NA       NA     NA              0        NA
## 4               NA     NA          NA       NA     NA              0        NA
## 5               NA     NA          NA       NA     NA              0        NA
## 6               NA     NA          NA       NA     NA              0        NA
##      Commodity
## 1 STRAWBERRIES
## 2 STRAWBERRIES
## 3 STRAWBERRIES
## 4 STRAWBERRIES
## 5 STRAWBERRIES
## 6 STRAWBERRIES
##                                                                 Data.Item
## 1                           STRAWBERRIES - APPLICATIONS, MEASURED IN LB
## 2                           STRAWBERRIES - APPLICATIONS, MEASURED IN LB
## 3                           STRAWBERRIES - APPLICATIONS, MEASURED IN LB
## 4                           STRAWBERRIES - APPLICATIONS, MEASURED IN LB
## 5 STRAWBERRIES - APPLICATIONS, MEASURED IN LB / ACRE / APPLICATION, AVG
## 6 STRAWBERRIES - APPLICATIONS, MEASURED IN LB / ACRE / APPLICATION, AVG
##                  Domain     type        name
## 1   CHEMICAL, FUNGICIDE CHEMICAL   FUNGICIDE
## 2 CHEMICAL, INSECTICIDE CHEMICAL INSECTICIDE
## 3 CHEMICAL, INSECTICIDE CHEMICAL INSECTICIDE
## 4       CHEMICAL, OTHER CHEMICAL       OTHER
## 5   CHEMICAL, FUNGICIDE CHEMICAL   FUNGICIDE
## 6 CHEMICAL, INSECTICIDE CHEMICAL INSECTICIDE
##                                          code Value CV....
## 1                   (OXATHIAPIPROLIN = 128111)    NA     NA
## 2                     (CYCLANILIPROLE = 26202)    NA     NA
## 3                        (PERMETHRIN = 109701)    NA     NA
## 4 (ISARIA FUMOSOROSEA STRAIN FE 9901 = 115003)    NA     NA
## 5                   (OXATHIAPIPROLIN = 128111)    NA     NA
## 6                     (CYCLANILIPROLE = 26202)    NA     NA
```

```
# Remove rows where the Value column is NA
chemical_clean <- chemical |> filter(!is.na(Value))
# Filter strawberry data for California and Florida
ca_chemical <- filter(chemical_clean, State == "CALIFORNIA")
fl_chemical <- filter(chemical_clean, State == "FLORIDA")
unique(ca_chemical$name)
```

```
## [1] "FUNGICIDE"   "HERBICIDE"   "INSECTICIDE" "OTHER"
```

```
unique(fl_chemical$name)
```

```
## [1] "FUNGICIDE"   "HERBICIDE"   "INSECTICIDE" "OTHER"
```

```
unique(ca_chemical$code)
```

```
##  [1] "(AZOXYSTROBIN = 128810)"
##  [2] "(BORAX DECAHYDRATE = 11102)"
##  [3] "(BOSCALID = 128008)"
##  [4] "(CAPTAN = 81301)"
```

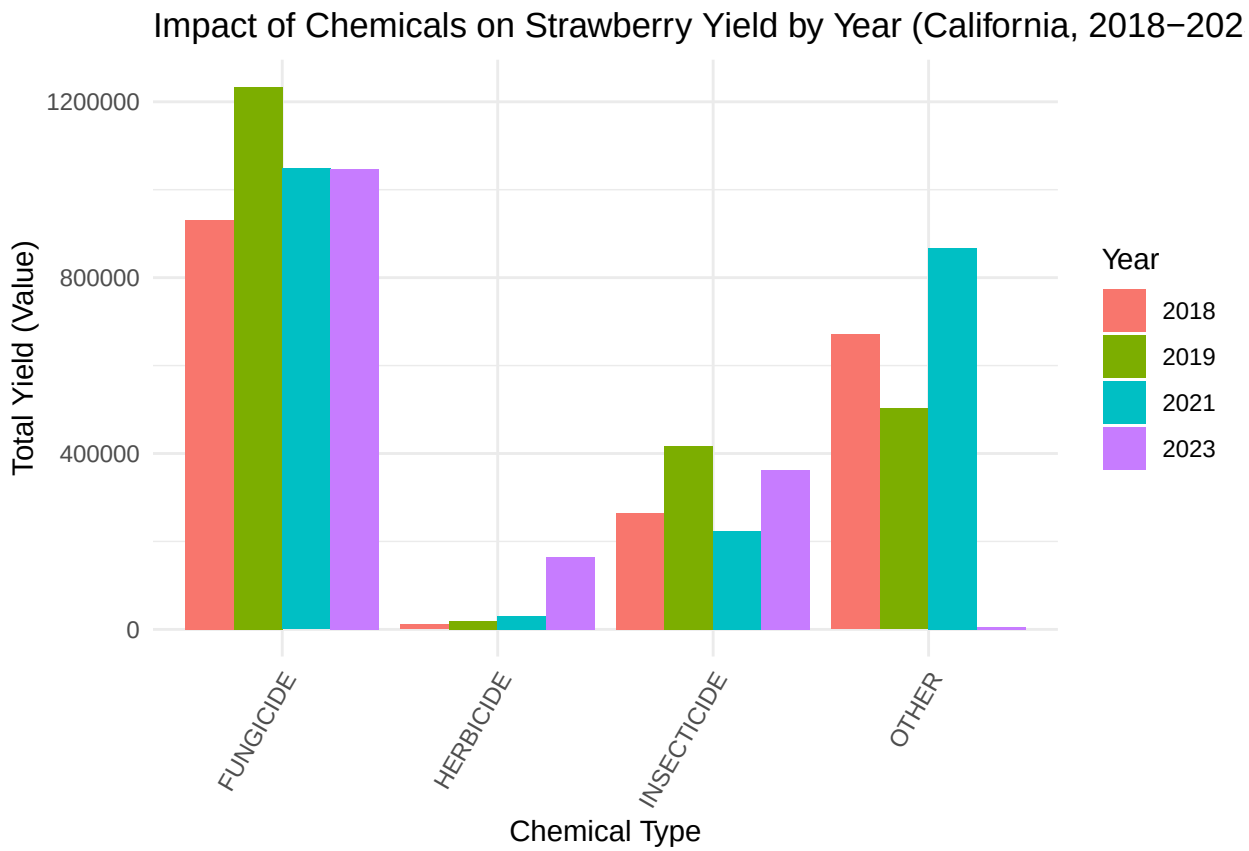
```
##  [5] "(CYPRODINIL = 288202)"
##  [6] "(FENHEXAMID = 90209)"
##  [7] "(FLUDIOXONIL = 71503)"
##  [8] "(FLUOPYRAM = 80302)"
##  [9] "(FLUXAPYROXAD = 138009)"
## [10] "(MEFENOXAM = 113502)"
## [11] "(MYCLOBUTANIL = 128857)"
## [12] "(PENTHIOPYRAD = 90112)"
## [13] "(POLYOXIN D ZINC SALT = 230000)"
## [14] "(PROPICONAZOLE = 122101)"
## [15] "(PYRACLOSTROBIN = 99100)"
## [16] "(PYRIMETHANIL = 288201)"
## [17] "(QUINOLINE = 55459)"
## [18] "(TETRACONAZOLE = 120603)"
## [19] "(THIOPHANATE-METHYL = 102001)"
## [20] "(THIRAM = 79801)"
## [21] "(TRIFLOXYSTROBIN = 129112)"
## [22] "(FLUMIOXAZIN = 129034)"
## [23] "(PENDIMETHALIN = 108501)"
## [24] "(TOTAL)"
## [25] "(ABAMECTIN = 122804)"
## [26] "(ACEQUINOCYL = 6329)"
## [27] "(ACETAMIPRID = 99050)"
## [28] "(AZADIRACHTIN = 121701)"
## [29] "(BIFENAZATE = 586)"
## [30] "(BIFENTHRIN = 128825)"
## [31] "(CHLORANTRANILIPROLE = 90100)"
## [32] "(CYANTRANILIPROLE = 90098)"
## [33] "(CYFLUMETOFEN = 138831)"
## [34] "(FENPROPATHRIN = 127901)"
## [35] "(FLONICAMID = 128016)"
## [36] "(FLUPYRADIFURONE = 122304)"
## [37] "(HEXYTHIAZOX = 128849)"
## [38] "(IMIDACLOPRID = 129099)"
## [39] "(MALATHION = 57701)"
## [40] "(METHOXYFENOZIDE = 121027)"
## [41] "(NOVALURON = 124002)"
## [42] "(SPINETORAM = 110007)"
## [43] "(SPINOSAD = 110003)"
## [44] "(THIAMETHOXAM = 60109)"
## [45] "(FLUTRIAFOL = 128940)"
## [46] "(SULFUR = 77501)"
## [47] "(CHLOROPICRIN = 81501)"
## [48] "(DICHLOROPROPENE = 29001)"
## [49] "(BT KURSTAKI ABTS-351 = 6522)"
## [50] "(BT KURSTAKI SA-11 = 6519)"
## [51] "(CYFLUFENAMID = 555550)"
## [52] "(POTASSIUM BICARBON. = 73508)"
## [53] "(TRIFLUMIZOLE = 128879)"
## [54] "(OXYFLUORFEN = 111601)"
## [55] "(FENPYROXIMATE = 129131)"
## [56] "(NALED = 34401)"
## [57] "(NEEM OIL = 25006)"
## [58] "(NEEM OIL, CLAR. HYD. = 25007)"
```

```
## [59] "(PYRETHRINS = 69001)"
## [60] "(SPIROMESIFEN = 24875)"
## [61] "(HYDROGEN PEROXIDE = 595)"
## [62] "(PEROXYACETIC ACID = 63201)"
## [63] "(REYNOUTRIA SACHALINE = 55809)"
## [64] "(METAM-POTASSIUM = 39002)"
## [65] "(BACILLUS AMYLOLIQUEFACIENS STRAIN D747 = 16482)"
## [66] "(BACILLUS SUBTILIS = 6479)"
## [67] "(BT KURSTAK ABTS-1857 = 6523)"
## [68] "(CHROMOBAC SUBTSUGAE PRAA4-1 CELLS AND SPENT MEDIA = 16329)"
## [69] "(BLAD = 30006)"
## [70] "(DIFENOCONAZOLE = 128847)"
## [71] "(FOSETYL-AL = 123301)"
## [72] "(CARFENTRAZONE-ETHYL = 128712)"
## [73] "(ETOXAZOLE = 107091)"
## [74] "(PIPERONYL BUTOXIDE = 67501)"
## [75] "(PYRIPROXYFEN = 129032)"
## [76] "(SULFOXAFLOR = 5210)"
## [77] "(IRON PHOSPHATE = 34903)"
## [78] "(METAM-SODIUM = 39003)"
## [79] "(BT SUBSP KURSTAKI EVB-113-19 = 6544)"
## [80] "(BT SUB AIZAWAI GC-91 = 6426)"
## [81] "(BURKHOLDERIA A396 CELLS & MEDIA = 6534)"
## [82] "(CYFLUMETOFEN = 138831)"
```

```
unique(fl_chemical$code)
```

```
##  [1] "(AZOXYSTROBIN = 128810)"       "(CAPTAN = 81301)"
##  [3] "(CYPRODINIL = 288202)"         "(FLUDIOXONIL = 71503)"
##  [5] "(THIRAM = 79801)"              "(TOTAL)"
##  [7] "(ABAMECTIN = 122804)"          "(ACETAMIPRID = 99050)"
##  [9] "(BIFENTHRIN = 128825)"         "(CHLORANTRANILIPROLE = 90100)"
## [11] "(NOVALURON = 124002)"          "(THIAMETHOXAM = 60109)"
## [13] "(SPINETORAM = 110007)"         "(THIOPHANATE-METHYL = 102001)"
## [15] "(BIFENAZATE = 586)"
```

```
# Calculate the total yield of each chemical per year
ca_chemical_filtered <- ca_chemical[ca_chemical$Year %in% 2018:2023, ]

ca_chemical_value <- aggregate(Value ~ name + Year, data = ca_chemical_filtered, FUN = sum, na.rm = TRU

# Create a bar plot
ggplot(ca_chemical_value, aes(x = name, y = Value, fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +  # Use position = "dodge" to separate the bars for
  labs(title = "Impact of Chemicals on Strawberry Yield by Year (California, 2018-2023)",
       x = "Chemical Type",
       y = "Total Yield (Value)",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



The bar chart titled "Impact of Chemicals on Strawberry Yield by Year (California, 2018-2023)" illustrates the total strawberry yield in California under the effects of different chemical treatments (as indicated in the name column) for the years 2018, 2019, 2021, and 2023. The name column specifies four categories of chemicals: fungicide, herbicide, insecticide, and other. According to the chart, fungicide is the most commonly used chemical in strawberry production, and its application has the most significant impact on strawberry yields. Areas that only use fungicides reached peak strawberry yields in 2019, followed by a decline, with yields stabilizing in 2021 and 2023.

The influence of insecticides on strawberry yields shows some fluctuations, but their use also had a considerable impact on yields in 2019. Among the three types of chemicals, areas that only use herbicides exhibited relatively lower strawberry yields but showed a clear upward trend during the assessed years.

Analysis: Fungicide is the most commonly used chemical in strawberry production in California, likely because strawberries are highly susceptible to fungal diseases, which can significantly reduce yield and fruit quality. California's climate, particularly in coastal regions, creates favorable conditions for fungal growth. Therefore, farmers heavily rely on fungicides to protect their crops and ensure consistent yields.

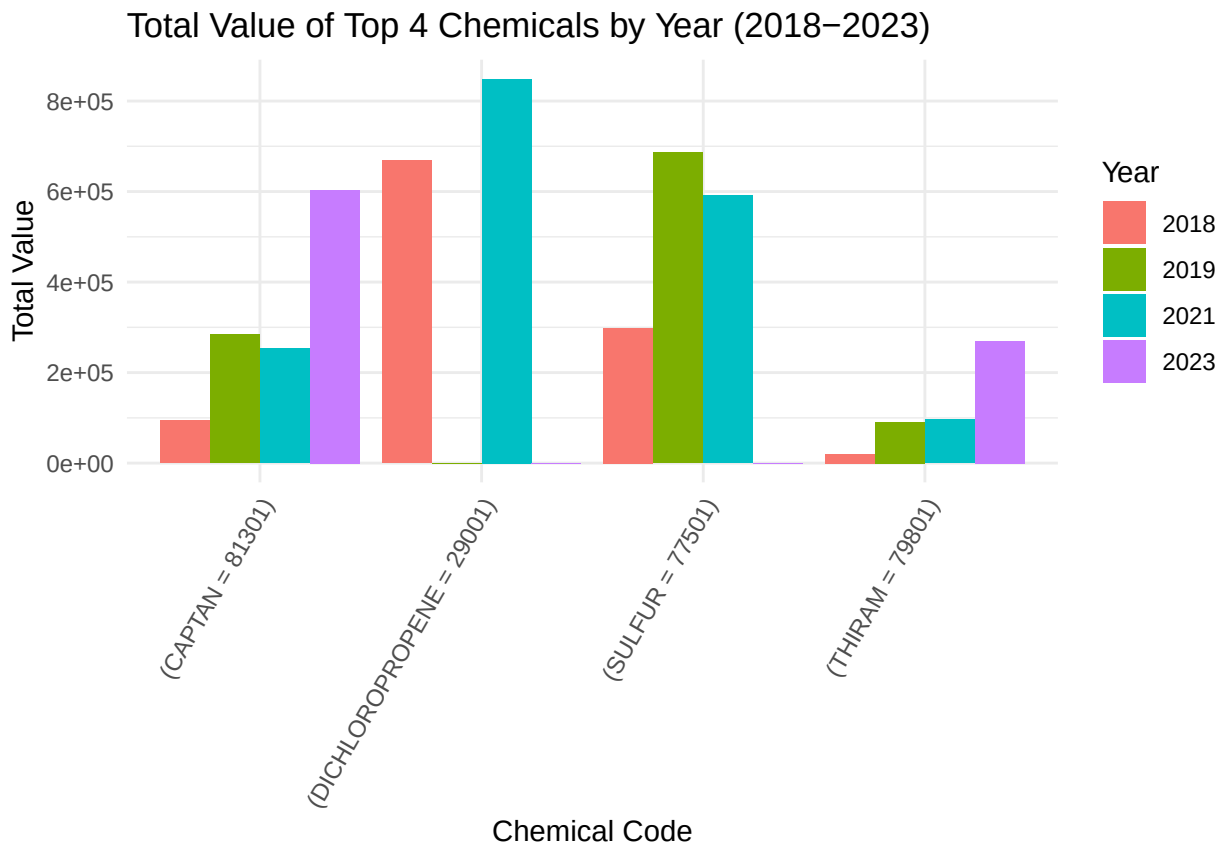
As for herbicides, the increasing yields in areas using herbicides could be attributed to improved weed management over time. Weeds compete with strawberry plants for nutrients, water, and sunlight, so effective weed control helps maximize plant health and fruit production. The rising yields in these areas may indicate that farmers are adopting better weed control practices, contributing to improved overall strawberry production.

```
# Group by chemical code and calculate the total value
ca_chemical_grouped <- aggregate(Value ~ code, data = ca_chemical, FUN = sum, na.rm = TRUE)

# Sort by total value in descending order
ca_chemical_sorted <- ca_chemical_grouped[order(-ca_chemical_grouped$Value), ]
```

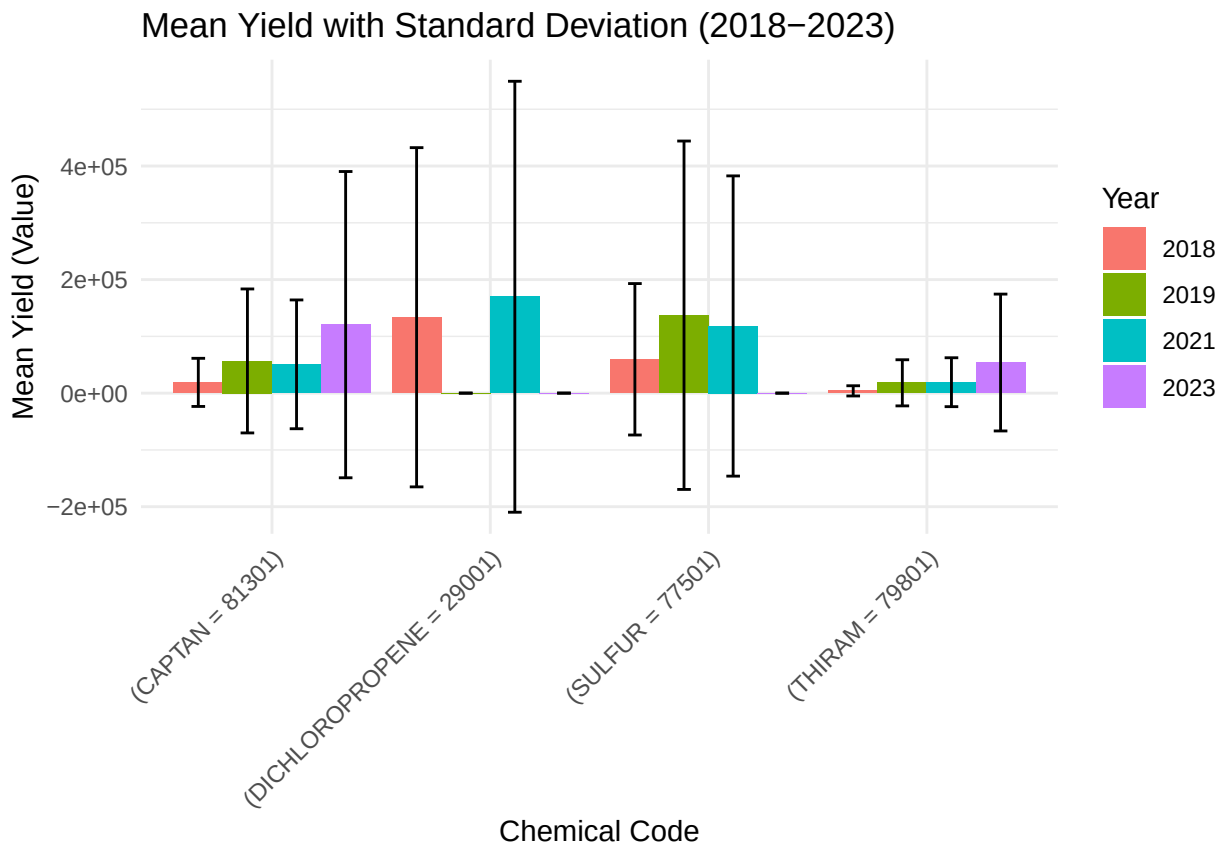
```r
# Extract the top 5 chemicals with the highest total values
ca_chemical_top5 <- head(ca_chemical_sorted, 5)

# Print the top 4 chemicals and their total values
print(ca_chemical_top5)
```

```
##                          code     Value
## 75          (SULFUR = 77501) 1575833.9
## 29 (DICHLOROPROPENE = 29001) 1517475.1
## 80                   (TOTAL) 1287249.0
## 19          (CAPTAN = 81301) 1235331.5
## 79          (THIRAM = 79801)  477024.6
```

```r
# Define the codes for the top 4 chemicals(except TOTAL)
top_chemical_codes <- c("(SULFUR = 77501)",
                        "(DICHLOROPROPENE = 29001)",
                        "(CAPTAN = 81301)",
                        "(THIRAM = 79801)")

# Filter to include the relevant columns for the top 4 chemicals
ca_5_no_total <- ca_chemical[ca_chemical$code %in% top_chemical_codes,
                             c("Year", "name", "code", "Value")]

# Calculate the total value of each chemical from 2018 to 2023
ca_chemical_summary <- aggregate(Value ~ code + Year,
                                 data = ca_5_no_total, FUN = sum)

# Step 2: Create a bar chart
ggplot(ca_chemical_summary, aes(x = code, y = Value,
                                fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +
  # Use position = "dodge" to separate bars by year
  labs(title = "Total Value of Top 4 Chemicals by Year (2018-2023)",
       x = "Chemical Code",
       y = "Total Value",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

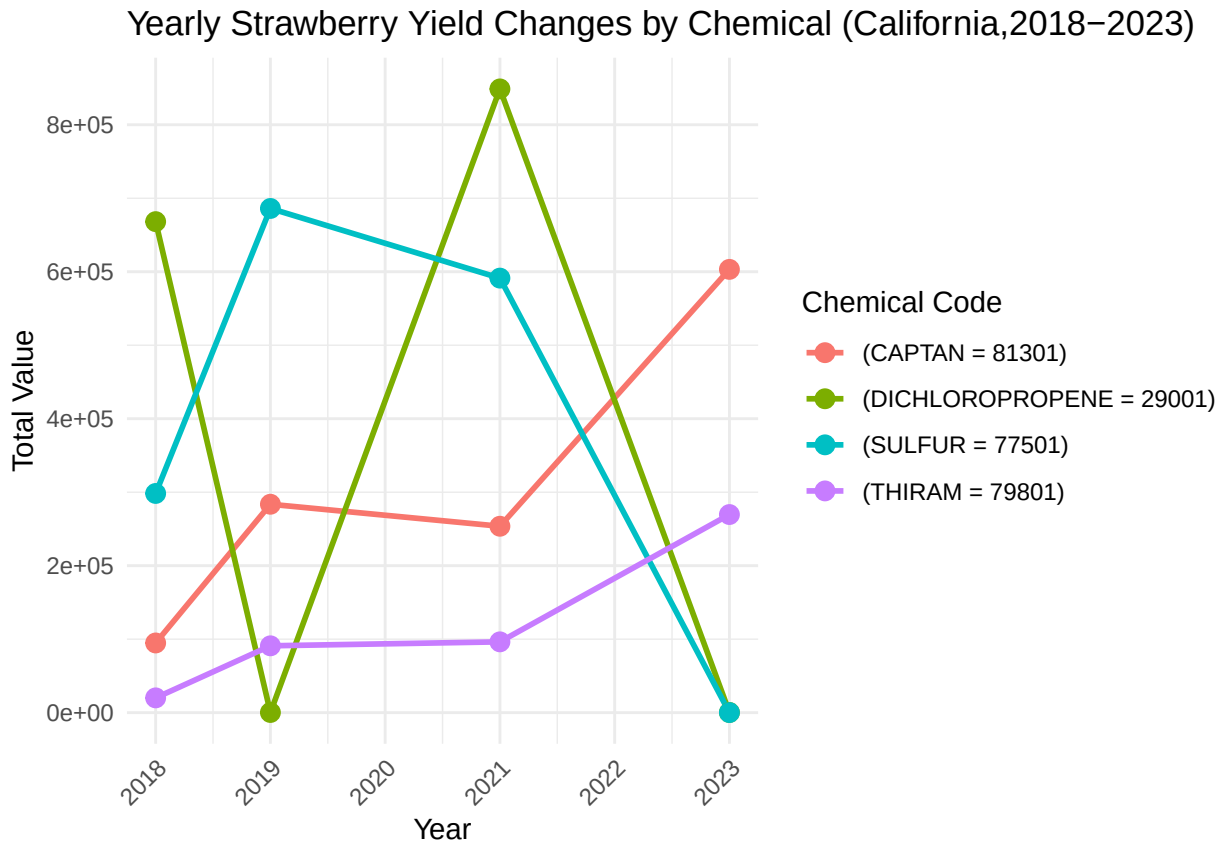



```r
# Filter data for the top four chemicals
top_chemical_codes <- c("(SULFUR = 77501)",
                        "(DICHLOROPROPENE = 29001)",
                        "(CAPTAN = 81301)",
                        "(THIRAM = 79801)")
top4_data <- subset(ca_chemical, code %in% top_chemical_codes & Year %in% 2018:2023)

# Calculate the mean and standard deviation of the total yield for each chemical per year
top4_summary <- aggregate(Value ~ code + Year, data = top4_data,
                          FUN = function(x) c(mean = mean(x),
                                              sd = sd(x)))
top4_summary <- do.call(data.frame, top4_summary)  # Expand the mean and sd columns
names(top4_summary)[3:4] <- c("Mean", "SD")  # Rename the columns

# Step 3: Create a bar chart with standard deviation
ggplot(top4_summary,
       aes(x = code, y = Mean, fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = Mean - SD, ymax = Mean + SD), position = position_dodge(width = 0.9), width =
  labs(title = "Mean Yield with Standard Deviation (2018-2023)",
       x = "Chemical Code",
       y = "Mean Yield (Value)",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Summarize yearly data for the top 4 chemicals
top4_yearly_summary <- aggregate(Value ~ code + Year, data = top4_data,
                                 FUN = sum)
ggplot(top4_yearly_summary, aes(x = Year, y = Value, group = code,
                                color = code)) +
                                geom_line(size = 1) +
                                geom_point(size = 3) +
                                labs(title =
      "Yearly Strawberry Yield Changes by Chemical (California,2018-2023)",
       x = "Year",
       y = "Total Value",
       color = "Chemical Code") +
                                theme_minimal() +
                                theme(axis.text.x =
                                      element_text(angle = 45, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The chart titled "Total Value of Top 4 Chemicals by Year (2018-2023)" is a bar graph that uses the code column to illustrate the total strawberry yield for the four chemicals with the highest production volumes between 2018, 2019, 2021, and 2023. SULFUR = 77501 and THIRAM = 79801 have the broadest usage, being utilized in all evaluation years. Both chemicals show a trend of increasing use over the evaluation years, which is also evident from the line graph titled "Yearly Strawberry Yield Changes by Chemical (2018-2023)."

The larger standard deviations for DICHLOROPROPENE and SULFUR indicate that the yield of these chemicals fluctuated more significantly across different years. In contrast, other chemicals like CAPTAN and THIRAM show smaller standard deviations, suggesting that their yields were relatively stable.

Analysis: Sulfur has inhibitory effects on various fungal diseases, particularly effective in controlling powdery mildew and rust diseases. It can also suppress the breeding of certain pests, playing a vital role in integrated pest management for crops.

As a natural mineral source, sulfur has low toxicity and is relatively environmentally friendly, making it widely used for disease prevention in various crops such as strawberries, grapes, and vegetables. In strawberry cultivation, sulfur is primarily used to prevent powdery mildew and other fungal diseases, ensuring the healthy growth of the crops and increasing yields. Due to its efficacy, low toxicity, and easy availability, sulfur holds a significant position in the cultivation of strawberries and other crops.

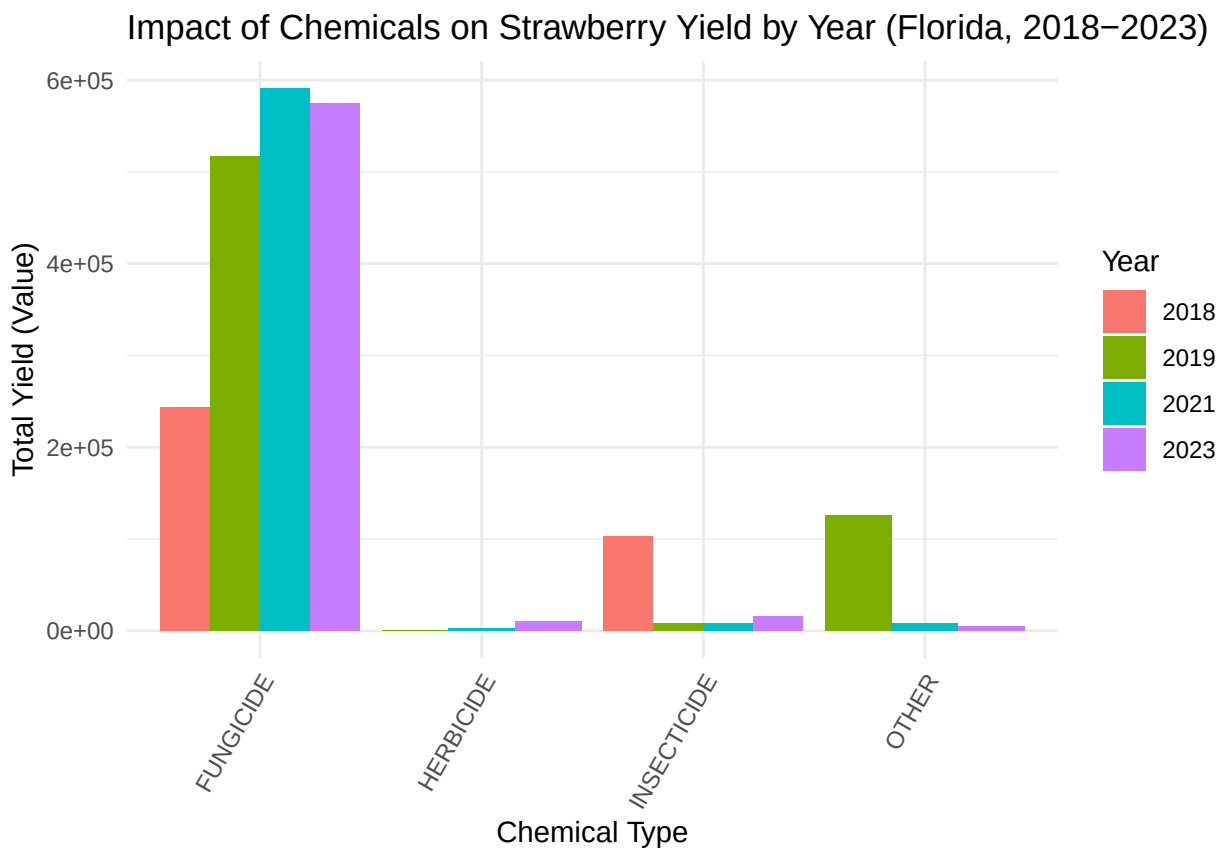
Thiram, like sulfur, is commonly used to control gray mold and anthracnose diseases caused by fungi. It works by disrupting the metabolic processes of pathogens to protect crops. Farmers regard Thiram as a reliable broad-spectrum fungicide suitable for disease management under various climatic and environmental conditions.

However, due to its certain toxicity, long-term exposure to Thiram can adversely affect the environment and human health. Therefore, its use is strictly regulated in modern agriculture, especially in organic and environmentally friendly farming practices, where the application of Thiram may face certain restrictions.

Dichloropropene and CAPTAN have not been used as widely as the first two chemicals. Although these chemicals are effective for strawberry cultivation, they also pose risks to human health and the environment. The non-use of these chemicals in strawberry fields in 2023 may be related to regulatory restrictions, the adoption of alternatives, crop management strategies, or changes in environmental conditions.

#FL data EDA

```
# Calculate the total yield of each chemical per year in Florida
fl_chemical_filtered <- fl_chemical[fl_chemical$Year %in% 2018:2023, ]

fl_chemical_value <- aggregate(Value ~ name + Year, data = fl_chemical_filtered, FUN = sum, na.rm = TRU

# Plot the bar chart
ggplot(fl_chemical_value,
       aes(x = name, y = Value, fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +
  # Use position = "dodge" to separate the bars for different years
  labs(title = "Impact of Chemicals on Strawberry Yield by Year (Florida, 2018-2023)",
       x = "Chemical Type",
       y = "Total Yield (Value)",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



Compared to California, Florida shows a greater variety of chemical types used in strawberry production, with a stronger reliance on fungicides and less dependence on other chemical agents.

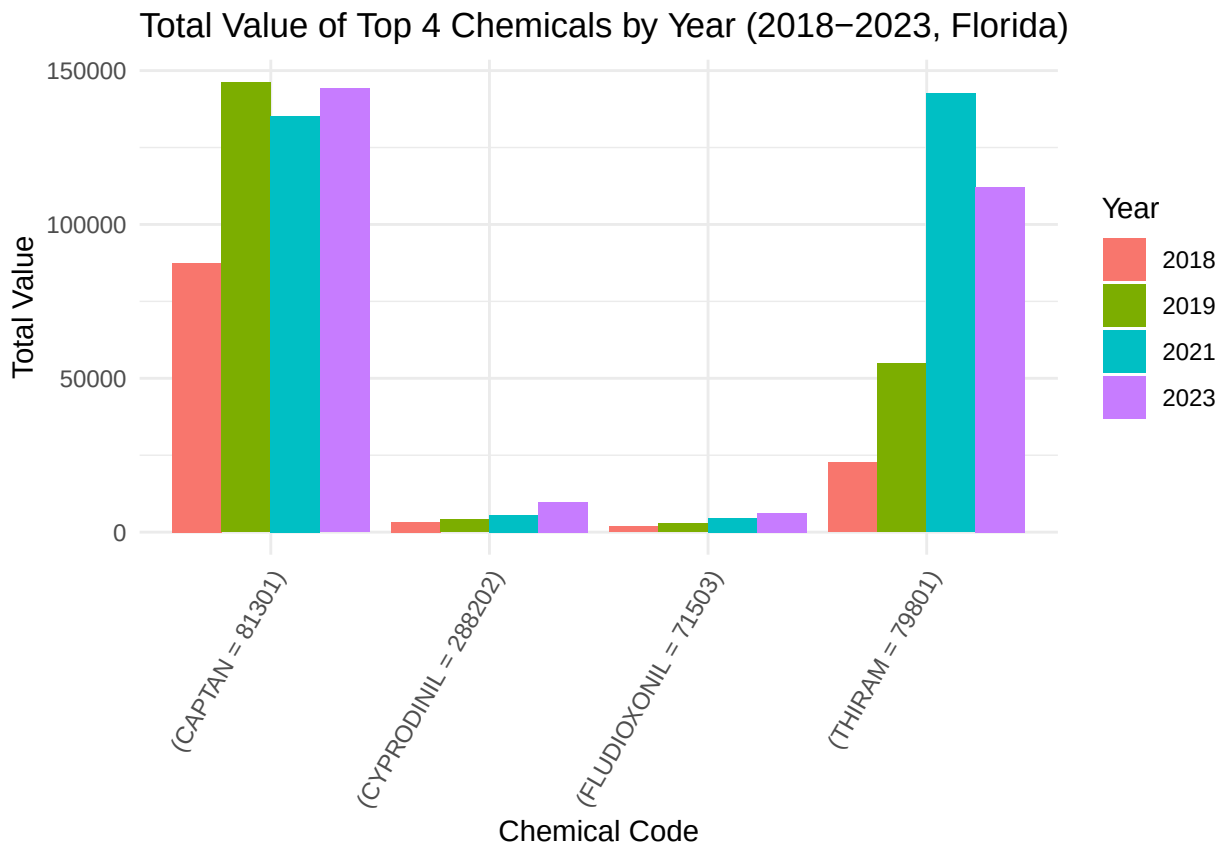
Analysis: The strawberry farming in Florida faces different threats from pests and diseases, which may

necessitate the use of fungicides to combat fungal-related diseases. This makes fungicides a preferred control method. In contrast, California's climate and soil conditions may lead to different pest and disease challenges, prompting farmers to choose other types of chemical agents, such as insecticides and herbicides, to address specific pest and weed issues.

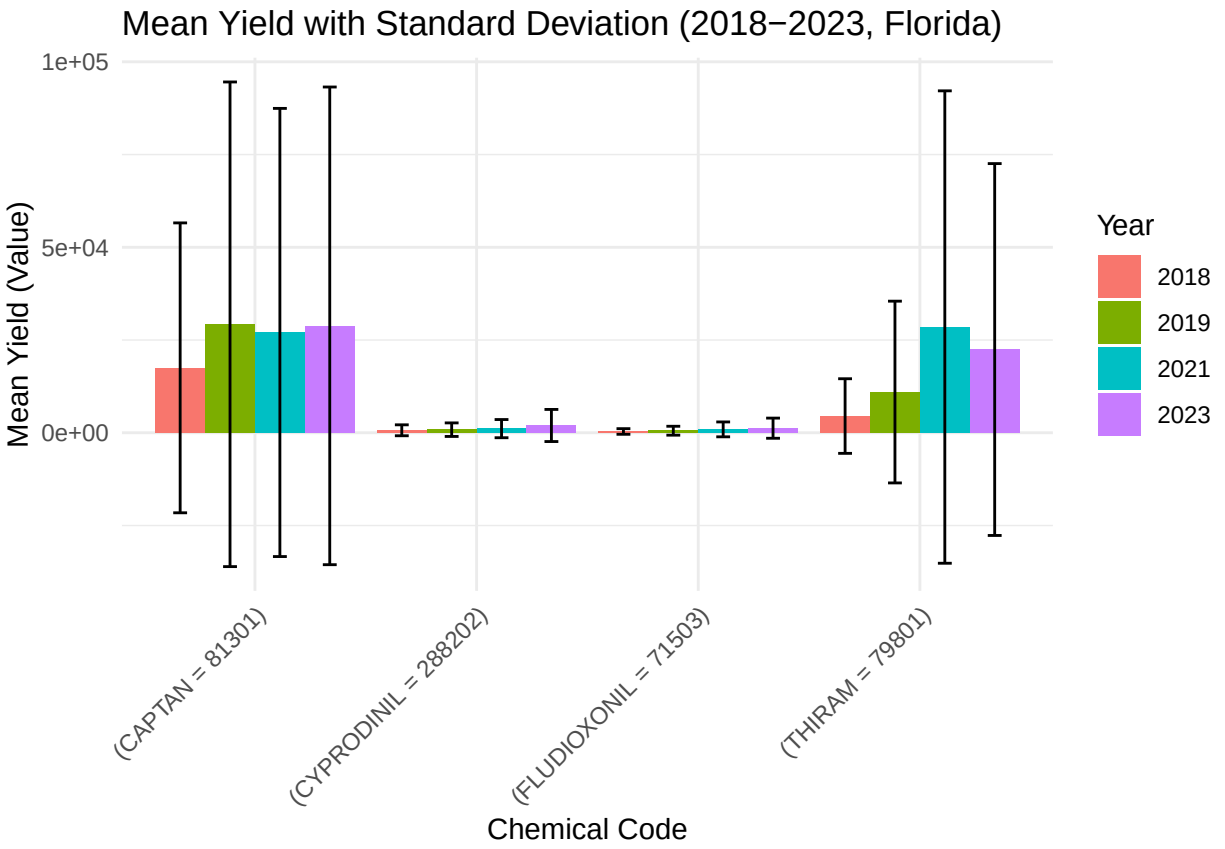
```
# Grouping by chemical code and calculating the total value
fl_chemical_grouped <- aggregate(Value ~ code, data = fl_chemical, FUN = sum, na.rm = TRUE)

# Sorting in descending order by total value
fl_chemical_sorted <- fl_chemical_grouped[order(-fl_chemical_grouped$Value), ]

# Extracting the top 5 chemicals by total value
fl_chemical_top5 <- head(fl_chemical_sorted, 5)

# Printing the top 5 chemicals and their total values
print(fl_chemical_top5)
```

```
##                     code      Value
## 15               (TOTAL) 1300660.00
## 6       (CAPTAN = 81301)  513035.49
## 14      (THIRAM = 79801)  332098.38
## 8  (CYPRODINIL = 288202)   22868.39
## 9  (FLUDIOXONIL = 71503)   15267.72
```

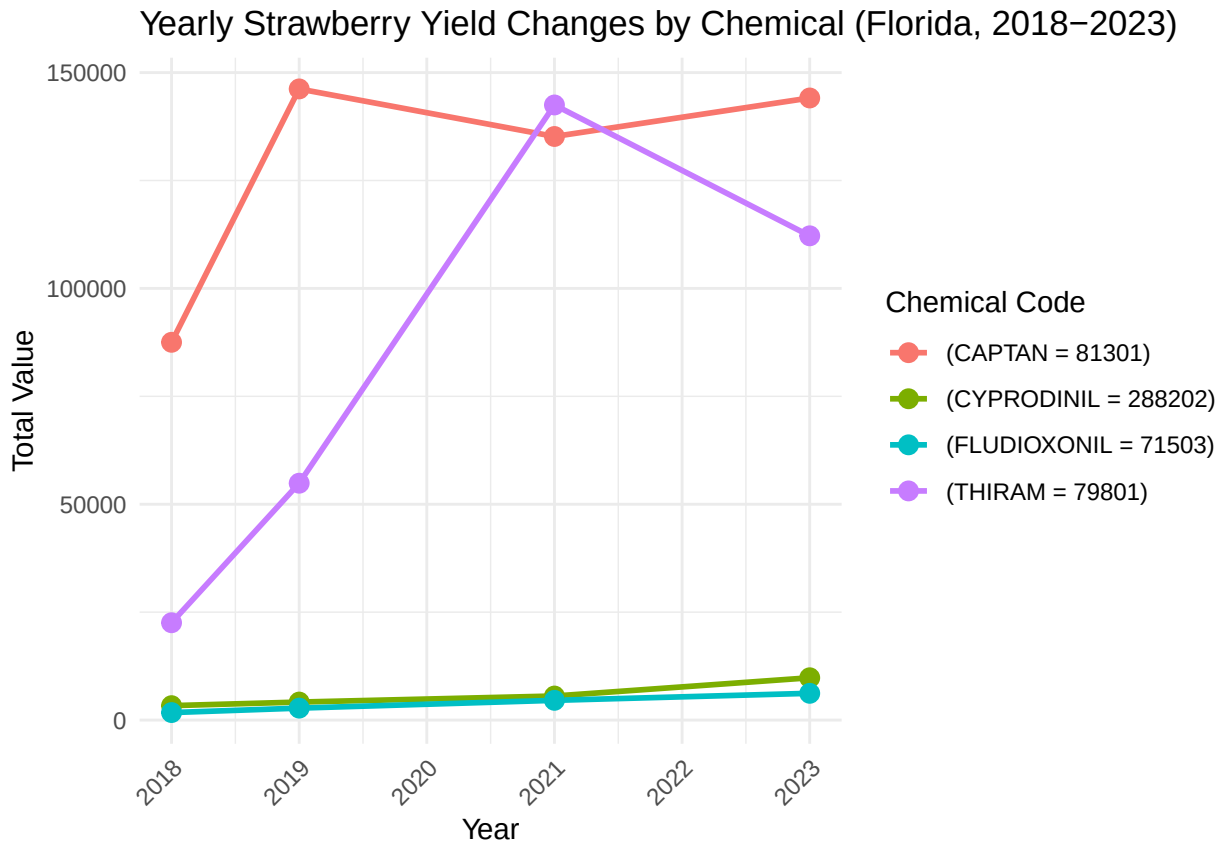
```
# Defining the codes for the top 4 chemicals (excluding TOTAL)
top_chemical_codes_fl <- c("(CAPTAN = 81301)",
                           "(THIRAM = 79801)",
                           "(FLUDIOXONIL = 71503)",
                           "(CYPRODINIL = 288202)")

# Filtering the relevant columns for the top 4 chemicals
fl_5_no_total <- fl_chemical[fl_chemical$code %in% top_chemical_codes_fl, c("Year", "name", "code", "Va

# Calculating the total value for each chemical from 2018 to 2023
fl_chemical_summary <- aggregate(Value ~ code + Year, data = fl_5_no_total, FUN = sum)

# Creating a bar plot
ggplot(fl_chemical_summary, aes(x = code, y = Value, fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +  # Using position = "dodge" to separate bars by yea
  labs(title = "Total Value of Top 4 Chemicals by Year (2018-2023, Florida)",
       x = "Chemical Code",
       y = "Total Value",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```




```r
# Filtering the data for the top 4 chemicals
top4_data_fl <- subset(fl_chemical,
                       code %in% top_chemical_codes_fl & Year %in% 2018:2023)

# Calculating the mean and standard deviation of total yield for each chemical each year
top4_summary_fl <- aggregate(Value ~ code + Year, data = top4_data_fl,
                             FUN = function(x) c(mean = mean(x), sd = sd(x)))
top4_summary_fl <- do.call(data.frame, top4_summary_fl)  # Flattening mean and sd columns
names(top4_summary_fl)[3:4] <- c("Mean", "SD")

# Creating a bar plot with standard deviation
ggplot(top4_summary_fl, aes(x = code, y = Mean, fill = as.factor(Year))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = Mean - SD, ymax = Mean + SD),
                position = position_dodge(width = 0.9), width = 0.25) +
  labs(title = "Mean Yield with Standard Deviation (2018-2023, Florida)",
       x = "Chemical Code",
       y = "Mean Yield (Value)",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Filtering the data for the top 4 chemicals again
fl_top4_data <- subset(fl_chemical, code %in% top_chemical_codes_fl & Year %in% 2018:2023)

# Calculating the total value for each chemical each year
fl_top4_yearly_summary <- aggregate(Value ~ code + Year, data = fl_top4_data, FUN = sum)

# Creating a line plot
ggplot(fl_top4_yearly_summary,
       aes(x = Year, y = Value, group = code, color = code)) +
       geom_line(size = 1) +
       geom_point(size = 3) +
       labs(title = "Yearly Strawberry Yield Changes by Chemical (Florida, 2018-2023)",
       x = "Year",
       y = "Total Value",
       color = "Chemical Code") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Unlike California, strawberry cultivation in Florida relies heavily on CAPTAN (81301) and THIRAM (79801), while California uses a wider variety of chemicals. The larger standard deviations for CAPTAN and THIRAM, shown by the longer black lines, indicate that the yields of these chemicals fluctuated more significantly across the years. In contrast, CYPRODINIL and FLUDIOXONIL have smaller standard deviations, with shorter black lines, suggesting that their yields were relatively stable with less variation between years.

Analysis: Florida's climate is relatively humid, which is conducive to the growth of fungi, particularly common strawberry diseases such as gray mold and powdery mildew. In such an environment, farmers are more inclined to use effective fungicides like CAPTAN and THIRAM to control diseases and ensure yield. Additionally, compared to California's strawberry production, Florida's overall strawberry acreage is smaller, leading farmers to be less concerned about the potential environmental impact of large-scale CAPTAN and THIRAM usage.