Strawberry EDA

Gary Wang

2024-10-21

Strawberries: Data

This is a project about acquiring strawberry data from the USDA-NASS system and then cleaning, organizing, and exploring the data in preparation for data analysis. To get started, I acquired the data from the USDA NASS system and downloaded them in a csv.

Strawberry data source and parameters

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 .

For the assignment, I stored the csv I downloaded on the MA615 Blackboard as strawberries25_v3.csv.

```r
library(readr)
library(knitr)
library(dplyr)
library(kableExtra)
library(tidyverse)
library(magrittr)
library(ggplot2)
```

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

The data was originally collected at the county, state, and national levels, but the degree of missingness at the state level was too high, so I dropped the county-level data.

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

There are 5,359 rows and 21 column in the initial data set. The only complete year is 2022, although there is data for years 2018 through 2024.

To work with the data, define a function to remove columns with only single value in all its rows.

```r
drop_one_value_col <- function(df, prt_val = FALSE){
 # browser()
  df_id <- ensym(df)
  if(prt_val){
```

```
  msg = paste("Looking for single value columns in data frame: ",as.character(df_id) )
  print(msg)}
  ## takes whole dataframe
dropc <- NULL
val <- NULL
## test each column for a single value
for(i in 1:dim(df)[2]){
  if(dim(distinct(df[,i]))[1] == 1){
    dropc <- c(dropc, i)
    val <- c(val, df[1,i])
  }
}

if(prt_val){
if(is.null(dropc)){
  print("No columns dropped")
  return(df)}else{
   print("Columns dropped:")
   # print(colnames(df)[drop])
   print(unlist(val))
   df <- df[, -1*dropc]
   return(df)
  }
}
 df <- df[, -1*dropc]
   return(df)
}

strawberry <- strawberry |> drop_one_value_col(prt_val = FALSE)
```

After this, our objective will be cleaning up “Data Item”, “Domain”, and “Domain Category” columns.

Data Item Column Clean Up

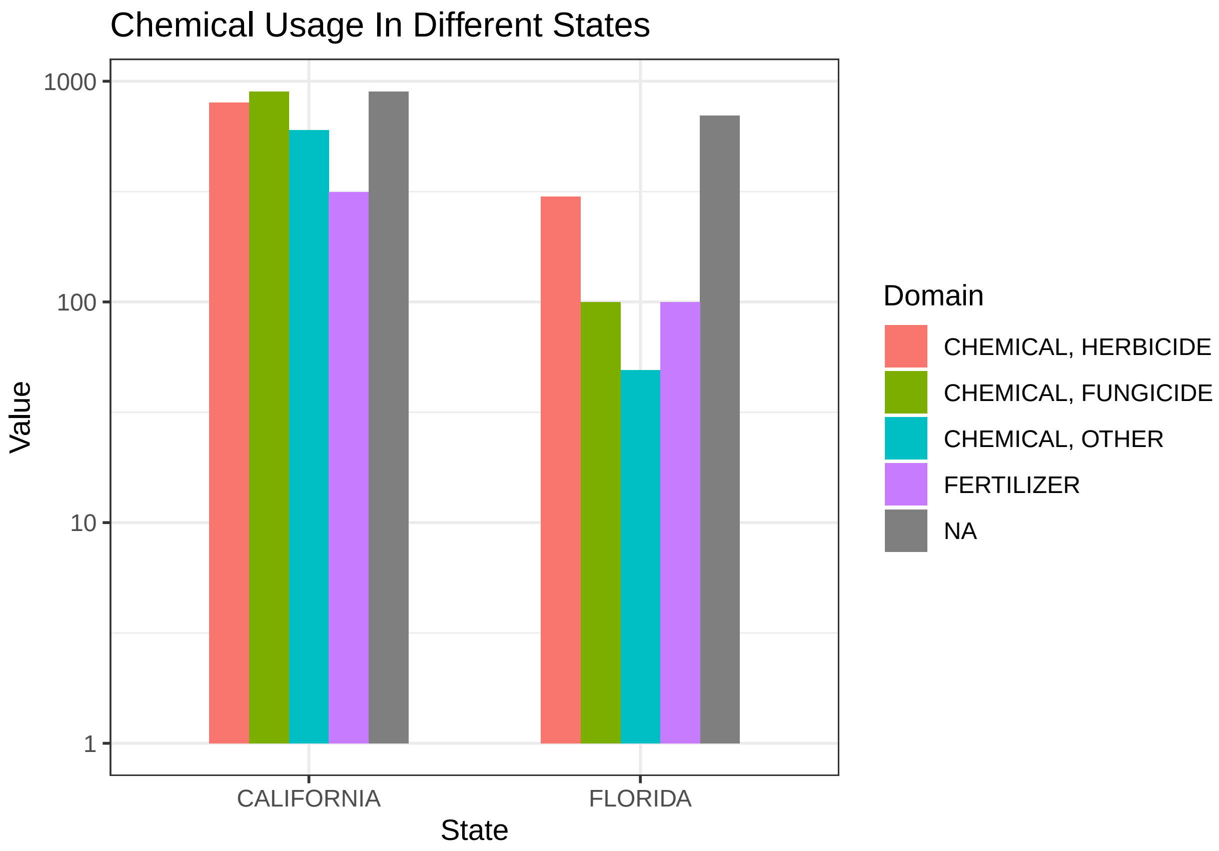
Since there are four different variables crumble in the “Data Item” column, our next task is to separate them into four different columns.

```
strawberry <- strawberry %>%
  separate(col = `Data Item`,
           into = c("Strawberries", "type", "items", "units"),
           sep = ",",
           fill = "right")

head(strawberry)
```

```
## # A tibble: 6 x 14
##   Program  Year Period ‘Geo Level‘ State   ‘State ANSI‘ Strawberries type  items
##   <chr>   <dbl> <chr>  <chr>       <chr>   <chr>        <chr>        <chr> <chr>
## 1 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>  <NA>
## 2 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>  <NA>
## 3 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>  <NA>
## 4 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>  <NA>
## 5 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>   <NA>
## 6 CENSUS   2022 YEAR   NATIONAL    US TOT~ <NA>         STRAWBERRIE~ <NA>   <NA>
## # i 5 more variables: units <chr>, Domain <chr>, ‘Domain Category‘ <chr>,
## #   Value <chr>, ‘CV (%)‘ <chr>
```

After doing so, we should see if there are any patterns for chemicals used in different states and have a initial view of prices for strawberries.

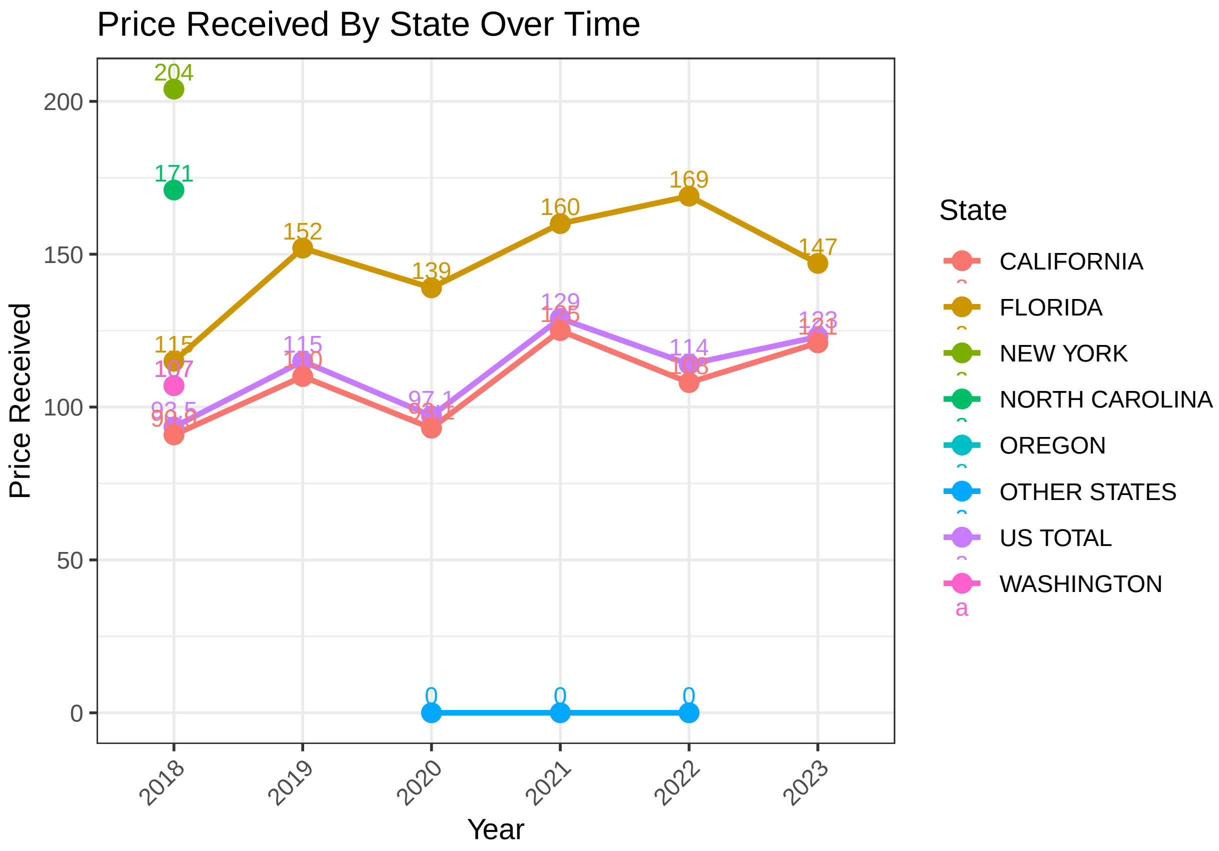
```
plot2.data <- strawberry %>%
    filter(
        str_detect(`Domain Category`, '(CHEMICAL|FERTILIZER)'),
        # State == 'CALIFORNIA',
        str_detect(Value, '^\\d+$')) %>%
    filter(Value != "(NA)" & Value != "(D)") %>%
    mutate(Value = as.numeric(Value),
           State = as.character(State),
           Domain = ordered(Domain, c('CHEMICAL, HERBICIDE', 'CHEMICAL,
                                      INSECTICIDE', 'CHEMICAL, FUNGICIDE',
                                      'CHEMICAL, OTHER', 'FERTILIZER')))

ggplot(plot2.data, aes(x = State, y = Value, fill = Domain)) +
    geom_bar(stat = 'identity', width = 0.6, position = 'dodge') +
    theme_bw() +
    scale_y_continuous(trans = 'log10') +
    scale_fill_hue() +
    labs(title = 'Chemical Usage In Different States')
```



From this plot, we can see that California uses more chemicals overall.

```r
plot3.data <- strawberry %>%
    filter(Strawberries == 'STRAWBERRIES - PRICE RECEIVED') %>%
    mutate(Value = as.numeric(Value),
           Year = as.character(Year))

ggplot(plot3.data, aes(x = Year, y = Value, color = State, group = State)) +
    geom_line(size = 1) +
    geom_point(size = 3) +
    theme_bw() +
    geom_text(aes(label = Value), size = 3, vjust = -0.5) +
    scale_y_continuous(labels = scales::comma) +
    labs(y = 'Price Received',
         title = 'Price Received By State Over Time') +
    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.
```



This plot shows us that Florida have higher values in this category than other states.

Domain Category Column Clean Up

After taking a glimpse, it seems to be a good idea to separate this column into three columns: organic, non-organic, and chemical.

```
# Find all organic entries

# Type: 62 rows
type_organic <- grep("organic",
                     strawberry$type,
                     ignore.case = TRUE)
# Zero rows returned
items_organic <- grep("organic",
                      strawberry$items,
                      ignore.case = TRUE)  ## nothing here
# Domain: 62 rows
Domain_organic <- grep("organic",
                       strawberry$Domain,
                       ignore.case = TRUE)
# Domain Category: 62 rows
Domain_Category_organic <- grep("organic",
                                strawberry$`Domain Category`,
                                ignore.case = TRUE)
org_rows <- intersect(type_organic, Domain_organic)

# Use slice function to get organic and non-organic strawberries subsets
strawberry_organic <- strawberry %>% slice(org_rows)
strawberry_non_organic <- strawberry %>% filter(!row_number() %in% org_rows)

# Chemicals (used in strawberry cultivation) subset
chem_rows <- grep("BEARING - APPLICATIONS",
                  strawberry_non_organic$type,
                  ignore.case = TRUE)
chem_rows_1 <- grep("chemical",
                    strawberry_non_organic$Domain,
                    ignore.case = TRUE)
ins <- intersect(chem_rows, chem_rows_1)
chem_rows_2 <- grep("chemical",
                    strawberry_non_organic$`Domain Category`,
                    ignore.case = TRUE)
ins_2 <- intersect(chem_rows, chem_rows_2)

strawberry_chem <- strawberry_non_organic %>% slice(chem_rows)
```

Chemical

```
# Drop columns with only one unique value
before_cols <- colnames(strawberry_chem)
T <- NULL
x <- length(before_cols)

for(i in 1:x) {
```

```r
  b <- length(unlist(strawberry_chem[, i] %>% unique()))
  T <- c(T, b)
}

drop_cols <- before_cols[which(T == 1)]
strawberry_chem <- strawberry_chem %>% dplyr::select(!all_of(drop_cols))
# Separate 'Domain Category' into 'dc1' and 'chem_name'
strawberry_chem <- strawberry_chem %>% separate(col = `Domain Category`,
                                                into = c("dc1", "chem_name"),
                                                sep = ":",
                                                fill = "right")

# Filter rows with "measured in" in 'items'
aa <- grep("measured in", strawberry_chem$items, ignore.case = TRUE)

# Select key columns
strawberry_chem <- strawberry_chem %>% dplyr::select(Year, State, items,
                                                     units, dc1, chem_name,
                                                     Value)
strawberry_chem$items <- str_remove_all(strawberry_chem$items, "MEASURED IN ")
strawberry_chem <- strawberry_chem %>% rename(c(category = units, units = items))

# Find non-chemical rows (fertilizers)
bb <- grep("CHEMICAL, ", strawberry_chem$dc1, ignore.case = TRUE)
chem <- 1:2112
non_chem_rows <- setdiff(chem, bb)
fertilizers <- strawberry_chem %>% slice(non_chem_rows)

# Clean 'chem_name' column
strawberry_chem$dc1 <- str_remove_all(strawberry_chem$dc1, "CHEMICAL, ")
strawberry_chem <- strawberry_chem %>% rename(chem_types = dc1)

# Remove parentheses and separate 'chem_name' into name and code
strawberry_chem$chem_name <- str_remove_all(strawberry_chem$chem_name, "\\(|\\)")
strawberry_chem <- strawberry_chem %>% separate(col = chem_name,
                                                into = c("chem_name", "chem_code"),
                                                sep = "=",
                                                fill = "right")

# Check if "lb" in 'units' corresponds to NA in 'category'
aa <- which(strawberry_chem$units == " LB")
bb <- which(is.na(strawberry_chem$category))
```

This step cleans and processes a dataset of chemicals used in strawberry cultivation. It first removes columns with only one unique value, then separates entries in the "Domain Category" column to extract chemical information. Rows related to fertilizers are isolated, and unnecessary text is removed from key columns. The code further splits chemical names and codes, removes parentheses, and ensures consistency in units and categories. Finally, it checks for alignment between "lb" in the 'units' column and missing values in the 'category' column. Now, we have a clean chemicals subset.

Organic

Our next move is to clean the "Organic" data.

```r
# Select distinct Year and State combinations
temp1 <- strawberry_organic %>% dplyr::select(Year, State) %>% distinct()

# Remove "MEASURED IN " from 'units' column
strawberry_organic$units <- str_remove_all(strawberry_organic$units, "MEASURED IN ")

# Drop unnecessary columns
strawberry_organic <- strawberry_organic[!names(
  strawberry_organic) %in% c("Period", "State ANSI",
                             "Strawberries", "Domain",
                             "Domain Category")]

strawberry_organic
```

```
## # A tibble: 732 x 9
##    Program  Year `Geo Level` State    type             items units Value `CV (%)`
##    <chr>   <dbl> <chr>       <chr>    <chr>            <chr> <chr> <chr> <chr>
##  1 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - AC~   <NA>  <NA> 5,301 43.5
##  2 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - OP~   <NA>  <NA> 546   6.3
##  3 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - OP~   <NA>  <NA> 546   6.3
##  4 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - PR~  " ME~  <NA> 1,49~ 51.2
##  5 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - SA~  " ME~  <NA> 335,~ 45.7
##  6 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC - SA~  " ME~  <NA> 1,49~ 51.3
##  7 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC"       " FR~  <NA> 540   6.3
##  8 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC"       " FR~ " $"  (D)   (D)
##  9 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC"       " FR~ " CW~ 1,48~ 50.3
## 10 CENSUS   2021 NATIONAL    US TOTAL " ORGANIC"       " PR~  <NA> 18    24.9
## # i 722 more rows
```

Non-Organic

We do the same to the "Non-Organic", but we will divide the non-organic data into marketing year and year subsets for further analysis.

```r
## Begin cleaning non-organic data

# Subset for marketing year data
x5 <- strawberry_non_organic[strawberry_non_organic$Period == "MARKETING YEAR", ]

# Subset excluding marketing year data
x6 <- setdiff(strawberry_non_organic, x5)

# Remove "MEASURED IN " from 'items' column
strawberry_non_organic$items <- str_remove_all(strawberry_non_organic$items, "MEASURED IN ")

# Separate 'Domain Category' into 'types' and 'name'
strawberry_non_organic <- strawberry_non_organic %>% separate(col = `Domain Category`,
                                                              into = c("types", "name"),
                                                              sep = ":",
                                                              fill = "right")

# Identify rows where 'Domain' and 'types' differ
```

```r
x1 <- strawberry_non_organic[(strawberry_non_organic$Domain != strawberry_non_organic$types), ]

# Delete 'Domain' and retain 'types'
# Check if all 'types' entries start with "Chemical"
x2 <- grep("CHEMICAL, ", strawberry_non_organic$types, ignore.case = TRUE)

# Filter entries starting with "Chemical" in 'types'
x3 <- strawberry_non_organic[grepl("CHEMICAL, ", strawberry_non_organic$types), ]

# Find non-chemical types
x4 <- setdiff(strawberry_non_organic, x3)
```

Now, we can divide non-organic dataset into chemical and FERTILIZER subsets.

```r
# Clean chemical names: remove parentheses
strawberry_non_organic$name <- str_remove_all(strawberry_non_organic$name, "\\(")
strawberry_non_organic$name <- str_remove_all(strawberry_non_organic$name, "\\)")

# Separate 'name' into 'name' and 'code'
strawberry_non_organic <- strawberry_non_organic %>% separate(col = name,
                                                              into = c("name", "code"),
                                                              sep = "=",
                                                              fill = "right")

# Delete unnecessary columns
strawberry_non_organic <- strawberry_non_organic[!names(strawberry_non_organic) %in%
                                                   c("State ANSI", "types")]

# Slice non-organic data into subsets:
# 1. Marketing year data
strawberry_non_organic_my <- strawberry_non_organic[
  strawberry_non_organic$Period == "MARKETING YEAR", ]

# 2. Year data excluding marketing year
strawberry_non_organic_y <- setdiff(strawberry_non_organic, strawberry_non_organic_my)

# 3. Chemical and fertilizer data
strawberry_non_organic_chemical <- strawberry_non_organic[
  grepl("CHEMICAL, ", strawberry_non_organic$Domain), ]
strawberry_non_organic_fertilizers <- setdiff(
  strawberry_non_organic_y, strawberry_non_organic_chemical)
```

Domain Column Clean Up

Now, we separate the “Total” value from “Domain” column in the non-organic dataset to a new column along with its correspond values in the “Value” column.

```r
strawberry_non_organic <- strawberry_non_organic %>%
    mutate(
        Total = ifelse(Domain == "TOTAL", Value, NA),  # Move Value to Total if Domain is "TOTAL"
        Domain = ifelse(Domain == "TOTAL", NA, Domain), # Set Domain to NA where it was "TOTAL"
        Value = ifelse(!is.na(Total), NA, Value)       # Set Value to NA where it was moved to Total
    )
```

Store New Datasets

```
write_csv(strawberry_non_organic, "strawberry_non_organic.csv")
write_csv(strawberry_organic, "strawberry_organic.csv")
```

Checking the non_organic dataset, we realize it is a good idea to separate “Census” and “Survey” data as well.

```
non_organic_data <- read.csv("strawberry_non_organic.csv")

# Filter rows where Program is "SURVEY" and create a new dataset
survey_data <- non_organic_data %>%
  filter(Program == "SURVEY")

# Filter rows where Program is "CENSUS" and create another dataset
census_data <- non_organic_data %>%
  filter(Program == "CENSUS")

write.csv(survey_data, "strawberry_non_organic_survey.csv", row.names = FALSE)
write.csv(census_data, "strawberry_non_organic_census.csv", row.names = FALSE)
old_file <- "strawberry_non_organic.csv"
file.remove(old_file)
```

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

Chemical EDA

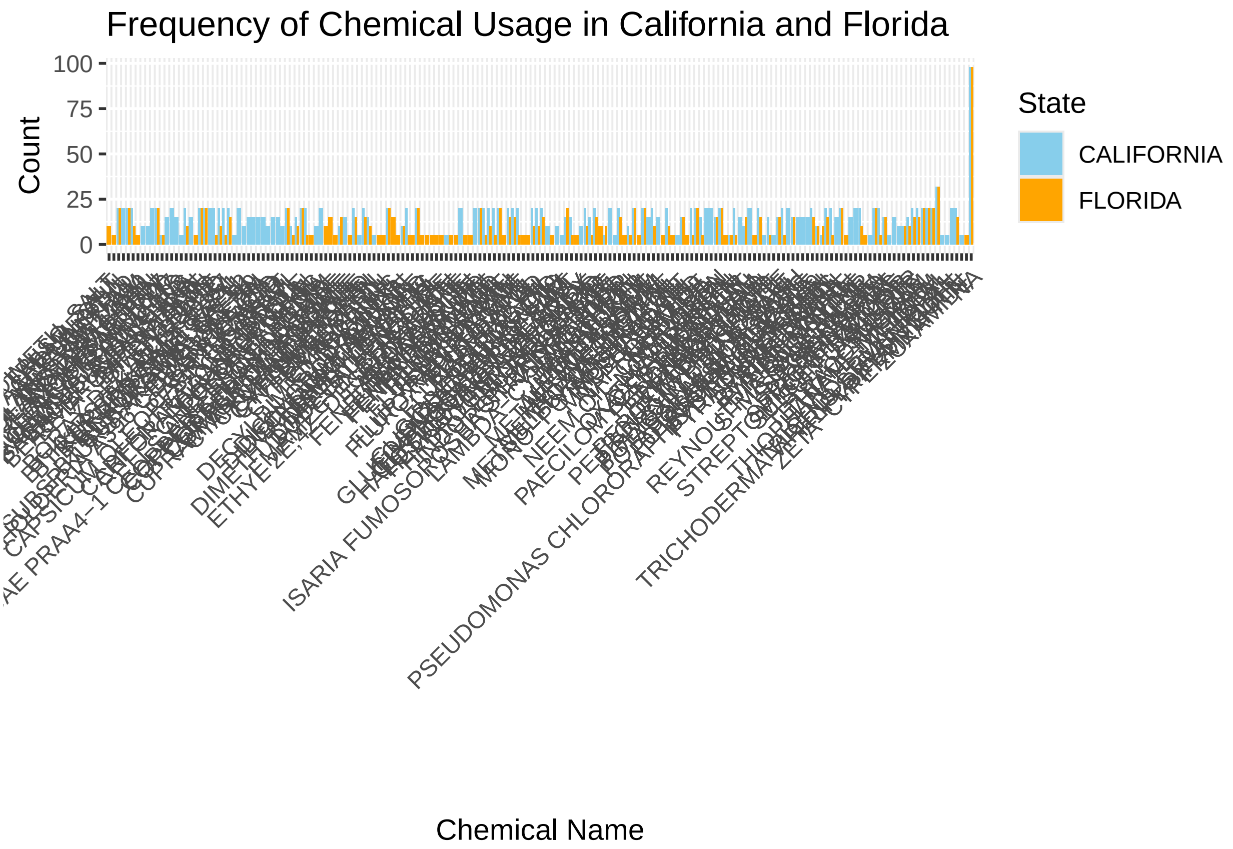
For this exploratory data analysis, we focus on California and Florida because these two states are among the largest agricultural producers in the United States, with significant strawberry farming activities. Their climates and large-scale agricultural practices make them key locations for studying pesticide and chemical usage, which can have substantial environmental and health impacts due to the high volume of application in these regions.

First, we will take a glimpse of the overall chemical usages in these two states.

```
survey_data <- read.csv("strawberry_non_organic_survey.csv")

# Filter data for California and Florida
state_data <- survey_data %>%
  filter(State == "CALIFORNIA" | State == "FLORIDA")

# Count the frequency of each chemical by state
chemical_count <- state_data %>%
  group_by(State, name) %>%
  summarise(Count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'State'. You can override using the
## `.groups` argument.
```

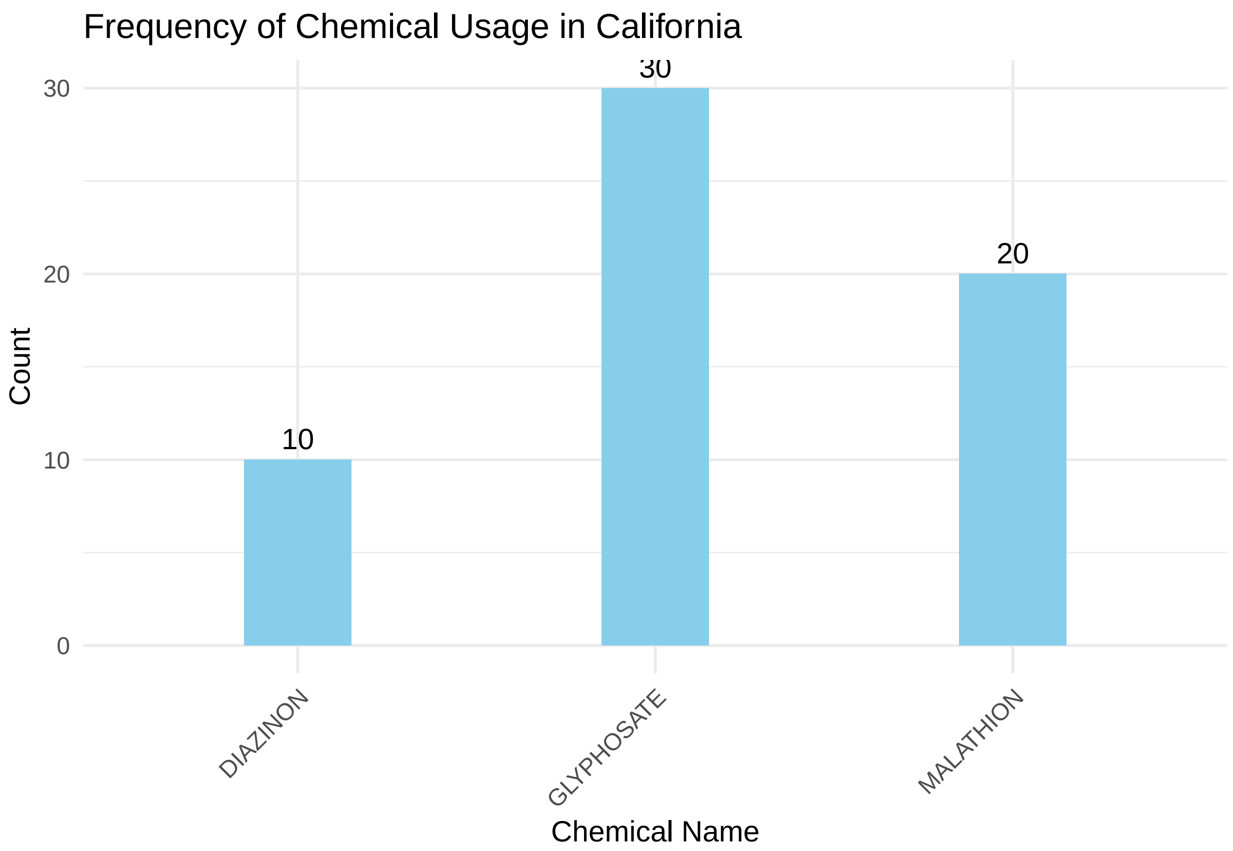
```
# Plot the frequency of chemical usage for each state
ggplot(chemical_count, aes(x = name, y = Count, fill = State)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Frequency of Chemical Usage in California and Florida",
       x = "Chemical Name", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("CALIFORNIA" = "skyblue", "FLORIDA" = "orange"))
```



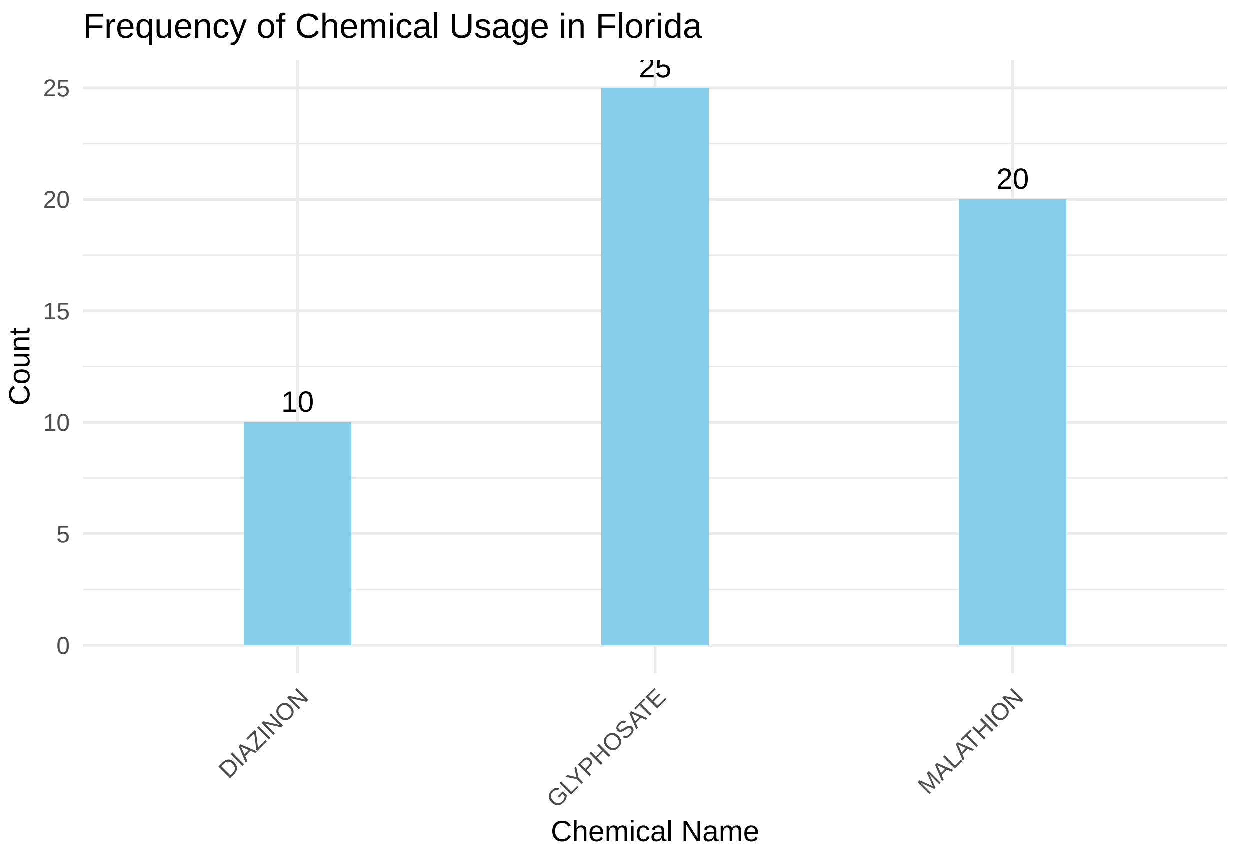
This plot is challenging to interpret due to several issues. First, the x-axis labels are overcrowded, causing the chemical names to overlap and become unreadable, making it difficult to identify individual chemicals and compare their usage counts. Additionally, displaying a large number of chemicals at once leads to a cluttered and visually overwhelming plot, which obscures patterns or trends in the data. Finally, including all chemicals regardless of their usage frequency dilutes the focus, making it harder to see the most relevant information. A more effective approach would be to filter the data to show only the most frequently used chemicals or to create separate plots for California and Florida to improve clarity.

In this case, we will specifically focus on Diazinon, Malathion, and Glyphosate, three widely used chemicals that are known for their toxicity. These chemicals are commonly applied in agriculture but have raised concerns due to their potential effects on human health, wildlife, and ecosystems. Diazinon and Malathion, both organophosphates, are toxic to the nervous systems of insects and animals, while Glyphosate is a controversial herbicide linked to possible carcinogenic effects. By analyzing the usage patterns of these chemicals in California and Florida, we aim to gain insights into potential risks and the scale of chemical reliance in high-production areas.

```r
# Filter data for California
cali_data <- survey_data[survey_data$State == "CALIFORNIA", ]

# Define the chemical names to check for
top_chemicals <- c("DIAZINON", "MALATHION", "GLYPHOSATE")

# Count occurrences for each chemical in California
diazinon_count <- sum(grepl("DIAZINON", cali_data$name, ignore.case = TRUE))
malathion_count <- sum(grepl("MALATHION", cali_data$name, ignore.case = TRUE))
glyphosate_count <- sum(grepl("GLYPHOSATE", cali_data$name, ignore.case = TRUE))

# Create a data frame for plotting
chemical_data <- data.frame(
  Chemical = c("DIAZINON", "MALATHION", "GLYPHOSATE"),
  Count = c(diazinon_count, malathion_count, glyphosate_count)
)

# Plot the data
ggplot(chemical_data, aes(x = Chemical, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue", width = 0.3) +
  geom_text(aes(label = Count), vjust = -0.5) +
  theme_minimal() +
  labs(title = "Frequency of Chemical Usage in California",
       x = "Chemical Name",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# Filter data for Florida
florida_data <- survey_data[survey_data$State == "FLORIDA", ]

# Define the chemical names to check for
top_chemicals <- c("DIAZINON", "MALATHION", "GLYPHOSATE")

# Count occurrences for each chemical in Florida
diazinon_count <- sum(grepl("DIAZINON", florida_data$name, ignore.case = TRUE))
malathion_count <- sum(grepl("MALATHION", florida_data$name, ignore.case = TRUE))
glyphosate_count <- sum(grepl("GLYPHOSATE", florida_data$name, ignore.case = TRUE))

# Create a data frame for plotting
chemical_data <- data.frame(
  Chemical = c("DIAZINON", "MALATHION", "GLYPHOSATE"),
  Count = c(diazinon_count, malathion_count, glyphosate_count)
)

# Plot the data
ggplot(chemical_data, aes(x = Chemical, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue", width = 0.3) +
  geom_text(aes(label = Count), vjust = -0.5) +
  theme_minimal() +
  labs(title = "Frequency of Chemical Usage in Florida",
       x = "Chemical Name",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Both states show the highest usage for Glyphosate, with California recording a count of 30 and Florida a slightly lower count of 25. Malathion ranks as the second most frequently used chemical in both states, with an equal count of 20 in each. Diazinon has the lowest usage in both states, with a count of 10 in both California and Florida.

The similarity in usage patterns suggests that these chemicals are widely applied in similar quantities in both agricultural regions. However, the slightly higher count of Glyphosate in California may indicate a greater reliance on this herbicide, potentially due to differences in crop types, farming practices, or pest management needs. Overall, the data highlights that both states rely heavily on Glyphosate, followed by Malathion, with Diazinon being the least used among the three.

Acknowledgement: Chatgpt was used in assisting correcting errors and plotting graphs.