Strawberries_Assignment

2024-10-02

Preparing data for analysis 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.

USDA NASS

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

#| label: read data - glimpse

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

glimpse(strawberry)
```

```
## Rows: 12,669
## Columns: 21
## $ Program             <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "~
## $ Year                <dbl> 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 202~
## $ Period              <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YE~
## $ `Week Ending`       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ `Geo Level`         <chr> "COUNTY", "COUNTY", "COUNTY", "COUNTY", "COUNTY", "~
## $ State               <chr> "ALABAMA", "ALABAMA", "ALABAMA", "ALABAMA", "ALABAM~
## $ `State ANSI`        <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01~
## $ `Ag District`       <chr> "BLACK BELT", "BLACK BELT", "BLACK BELT", "BLACK BE~
## $ `Ag District Code`  <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,~
## $ County              <chr> "BULLOCK", "BULLOCK", "BULLOCK", "BULLOCK", "BULLOC~
## $ `County ANSI`       <chr> "011", "011", "011", "011", "011", "011", "101", "1~
## $ `Zip Code`          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Region              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ watershed_code      <chr> "00000000", "00000000", "00000000", "00000000", "00~
## $ Watershed           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Commodity           <chr> "STRAWBERRIES", "STRAWBERRIES", "STRAWBERRIES", "ST~
## $ `Data Item`         <chr> "STRAWBERRIES - ACRES BEARING", "STRAWBERRIES - ACR~
## $ Domain              <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL~
## $ `Domain Category`   <chr> "NOT SPECIFIED", "NOT SPECIFIED", "NOT SPECIFIED", ~
## $ Value               <chr> "(D)", "3", "(D)", "1", "6", "5", "(D)", "(D)", "2"~
## $ `CV (%)`            <chr> "(D)", "15.7", "(D)", "(L)", "52.7", "47.6", "(D)",~
```

From the data, we can see that there are 12,669 rows with 21 columns. Some variables are not available, and hence will nedd some modification.

```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 rows

```r
#|label: drop 1-item columns

drop_one_value_col <- function(df){
drop <- NULL
for(i in 1:dim(df)[2]){
if((df |> distinct(df[,i]) |> count()) == 1){
drop = c(drop, i)
} }

if(is.null(drop)){return("none")}else{

   print("Columns dropped:")
   print(colnames(df)[drop])
   strawberry <- df[, -1*drop]
   }
}


## use the function

strawberry <- drop_one_value_col(strawberry)
```

```
## [1] "Columns dropped:"
## [1] "Week Ending"    "Zip Code"       "Region"         "watershed_code"
## [5] "Watershed"      "Commodity"
```

```r
drop_one_value_col(strawberry)
```

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

Separate composite columns Split Data Item into (fruit, category, item)

```r
#|label: split Data Item

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

## Use too_many and too_few to set up the separation operation.

# Save the updated dataframe to a new CSV file

#|label: fix the leading space

 # note
strawberry$Category[1]
```

```
## [1] NA
```

```r
# strawberry$Item[2]
# strawberry$Metric[6]
# strawberry$Domain[1]
##
## trim white space

strawberry$Category <- str_trim(strawberry$Category, side = "both")
strawberry$Item <- str_trim(strawberry$Item, side = "both")
strawberry$Metric <- str_trim(strawberry$Metric, side = "both")


write.csv(strawberry, file = "strawberry_separated.csv", row.names = FALSE)
```

Further processing and cleaning

```r
# Load required libraries
library(tidyverse)

# Step 1: Read the CSV file
strawberry <- read_csv("strawberry_separated.csv")
```

```
## Rows: 12669 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (16): Program, Period, Geo Level, State, State ANSI, Ag District, County...
## dbl  (2): Year, Ag District Code
##
## 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.
```

```
# Step 2: Clean and organize the 'Use', 'Name', and 'Code' columns, and remove 'Domain' and 'Domain Cat
strawberry_clean <- strawberry %>%
  # Extract 'Use' from the 'Domain' column
  mutate(
    Use = case_when(
      str_detect(`Domain`, "FUNGICIDE") ~ "FUNGICIDE",
      str_detect(`Domain`, "INSECTICIDE") ~ "INSECTICIDE",
      str_detect(`Domain`, "HERBICIDE") ~ "HERBICIDE",
      TRUE ~ NA_character_
    ),
    # Extract 'Name' from the 'Domain Category' column, removing the '= CODE' part
    Name = str_extract(`Domain Category`, "\\((.*?)\\)"),
    Name = str_replace_all(Name, " = \\d+", ""),  # Remove the '= CODE' part
    Name = str_replace_all(Name, "[()]", ""),    # Remove parentheses around 'Name'
    # Extract 'Code' from the 'Domain Category' column (after the '=' sign)
    Code = str_extract(`Domain Category`, "\\d+"),  # Extract only the numeric part of the code
    Code = str_trim(Code)  # Clean up any remaining whitespace
  ) %>%
  # Remove rows where 'Use', 'Name', or 'Code' are NA
  drop_na(Use, Name, Code) %>%
  # Remove the unwanted 'Domain' and 'Domain Category' columns
  select(-Domain, -`Domain Category`)

# Step 3: Save the cleaned dataset to a new CSV file
write_csv(strawberry_clean, "strawberry_separated_clean.csv")

# Output to verify the cleaned data
print(strawberry_clean)
```

```
## # A tibble: 2,805 x 19
##    Program  Year Period `Geo Level` State      `State ANSI` `Ag District`
##    <chr>   <dbl> <chr>  <chr>       <chr>      <chr>        <chr>
##  1 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  2 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  3 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  4 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  5 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  6 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  7 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  8 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
##  9 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
## 10 SURVEY   2023 YEAR   STATE       CALIFORNIA 06           <NA>
## # i 2,795 more rows
## # i 12 more variables: `Ag District Code` <dbl>, County <chr>,
## #   `County ANSI` <chr>, Fruit <chr>, Category <chr>, Item <chr>, Metric <chr>,
```

```
## #   Value <chr>, `CV (%)` <chr>, Use <chr>, Name <chr>, Code <chr>
```

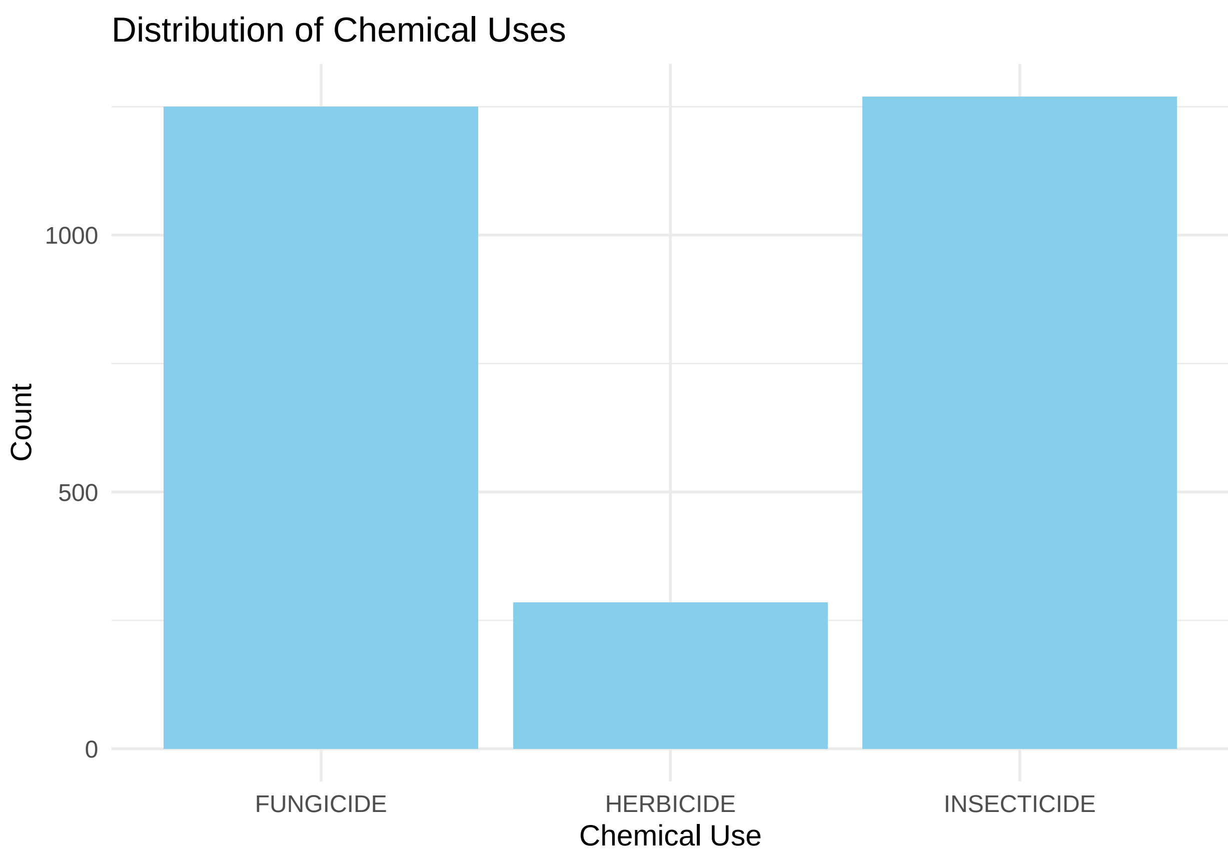
Display the cleaned data in a table

```r
# Display the first few rows of the cleaned data as a table
knitr::kable(head(strawberry_clean), format = "latex", booktabs = TRUE)
```

Program	Year	Period	Geo Level	State	State ANSI	Ag District	Ag District Code	County	Coun
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA
SURVEY	2023	YEAR	STATE	CALIFORNIA	06	NA	NA	NA	NA

PLOTS

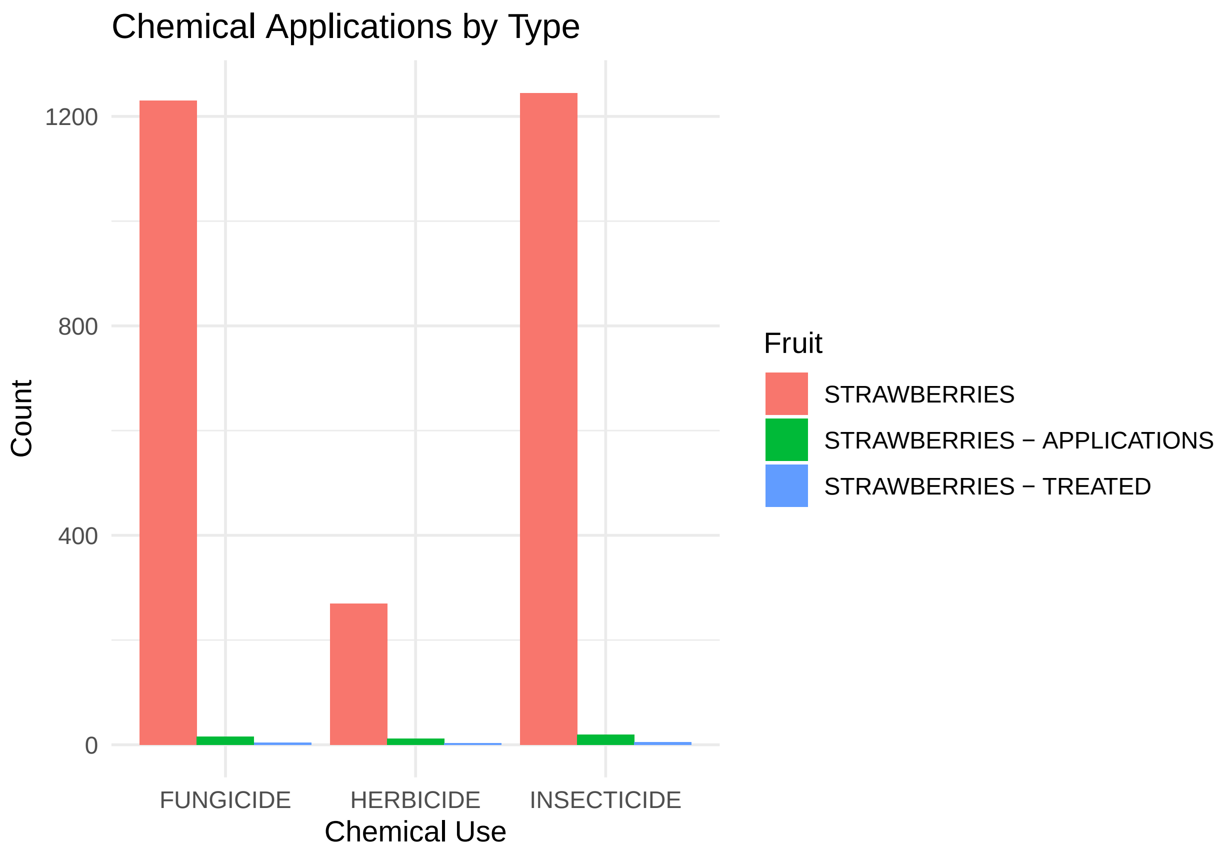
```r
# Bar plot of chemical use types (FUNGICIDE, INSECTICIDE, etc.)
ggplot(strawberry_clean, aes(x = Use)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Distribution of Chemical Uses", x = "Chemical Use", y = "Count") +
  theme_minimal()
```



From the graph above, we see that Fungicide and Insecticide are the most commonly used chemicals. They both have a high count of approximately 1000 observations, while Herbicide has a significantly lower count with fewer than 500 observations.

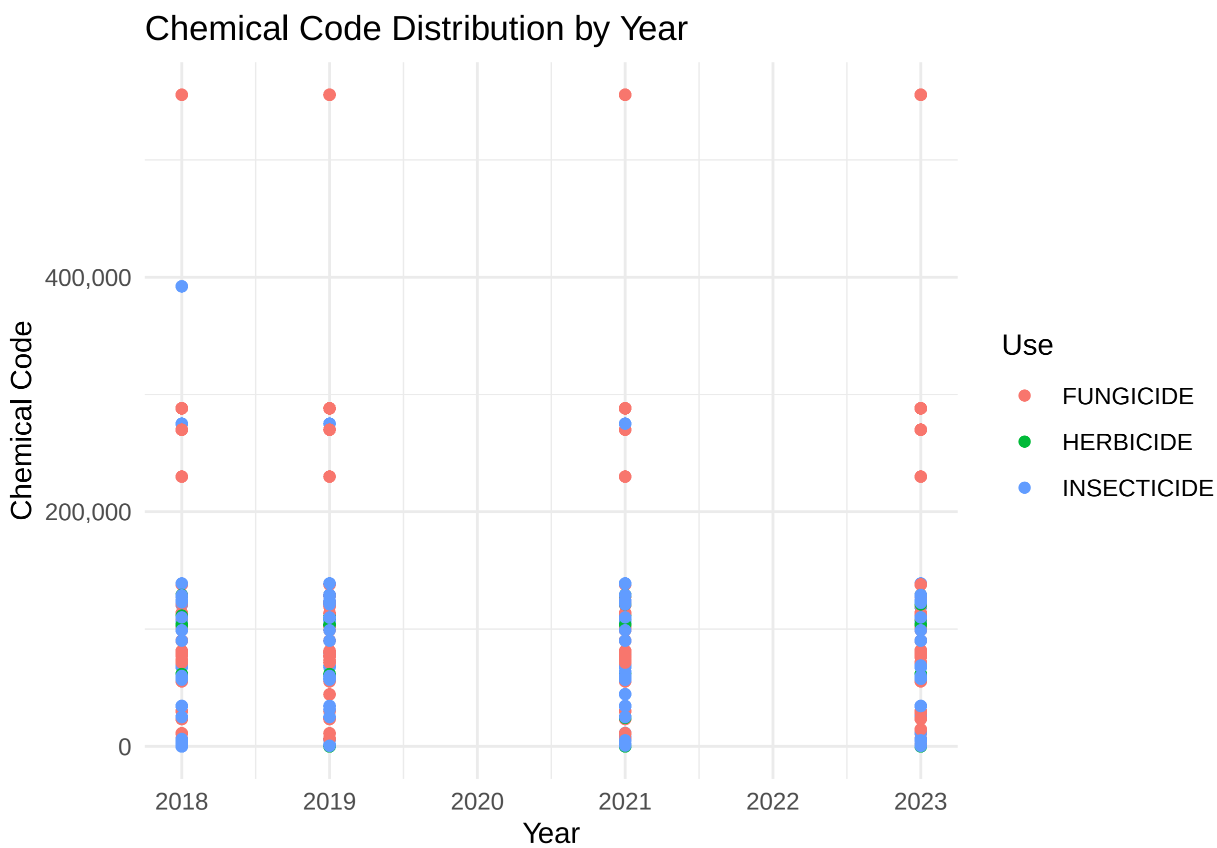
This suggests that Fungicide and Insecticide are commonly used in strawberry cultivation, and further investigation could explore why herbicides are used less frequently or how the use of these chemicals has changed over time.

```
# Plot the number of chemical applications (grouped by 'Use') across different measurement types
ggplot(strawberry_clean, aes(x = Use, fill = Fruit)) +
  geom_bar(position = "dodge") +
  labs(title = "Chemical Applications by Type", x = "Chemical Use", y = "Count", fill = "Fruit") +
  theme_minimal()
```



From this graph, we see that most applications are associated with strawberries, but we also see some minimal contributions from other categories (applications and treated).

```
# Scatter plot of chemical code distribution by year
ggplot(strawberry_clean, aes(x = Year, y = as.numeric(Code), color = Use)) +
  geom_point() +
  labs(title = "Chemical Code Distribution by Year", x = "Year", y = "Chemical Code") +
  scale_y_continuous(labels = scales::comma) +  # This will format the y-axis with commas instead of sc
  theme_minimal()
```



This scatter plot shows a stable trend in the use of chemical codes over the years. The usage of Fungicide and Insecticide has remained consistent, with no noticeable large fluctuations. However, data points for certain years like 2020 and 2022 seem to be missing. Further investigation could explore the reasons behind this or whether other similar data can fill in the gaps.

Conclusion Through cleaning, organizing, and visualizing the strawberry dataset, we have identified the predominant chemical types used in strawberry production and explored their trends over the years. This analysis provides a foundation for future exploration, including the reasons behind missing data points and the difference in chemical use across different periods and treatments.

```r
# Load the required libraries
library(tidyverse)

# Step 1: Read the original strawberries dataset
strawberries_orig <- read_csv("strawberries25_v3.csv")
```

```
## 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.
```

```r
# Step 2: Filter for Organic Strawberries
organic_strawberry <- strawberries_orig %>%
  filter(str_detect(`Data Item`, "ORGANIC"))

# Step 3: Save the filtered organic strawberries data to a new CSV file
write_csv(organic_strawberry, "organic_strawberries.csv")

# Optional: Print the first few rows to verify the data
print(head(organic_strawberry))
```

```
## # A tibble: 6 x 21
##   Program  Year Period `Week Ending` `Geo Level` State    `State ANSI`
##   <chr>   <dbl> <chr>  <lgl>         <chr>       <chr>    <chr>
## 1 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## 2 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## 3 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## 4 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## 5 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## 6 CENSUS   2021 YEAR   NA            NATIONAL    US TOTAL <NA>
## # i 14 more variables: `Ag District` <chr>, `Ag District Code` <dbl>,
## #   County <chr>, `County ANSI` <chr>, `Zip Code` <lgl>, Region <lgl>,
## #   watershed_code <chr>, Watershed <lgl>, Commodity <chr>, `Data Item` <chr>,
## #   Domain <chr>, `Domain Category` <chr>, Value <chr>, `CV (%)` <chr>
```

```r
# Load the required libraries
library(tidyverse)
library(stringr)

# Step 1: Read the original strawberries dataset
strawberries_orig <- read_csv("strawberries25_v3.csv")
```

```
## 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.
```

```r
# Step 2: Filter for Organic Strawberries
organic_strawberry <- strawberries_orig %>%
  filter(str_detect(`Data Item`, "ORGANIC"))

# Step 3: Remove columns with a single value in all rows
drop_one_value_col <- function(df) {
  drop <- NULL
  for (i in 1:ncol(df)) {
    if (n_distinct(df[[i]]) == 1) {
      drop <- c(drop, i)
    }
```

```r
  }

  if (is.null(drop)) {
    return(df)  # Return the original dataframe if no columns are dropped
  } else {
    print("Columns dropped:")
    print(colnames(df)[drop])
    df <- df[, -drop]  # Remove columns with only a single value
    return(df)
  }
}

# Apply the function to drop columns with a single value
organic_strawberry <- drop_one_value_col(organic_strawberry)
```

```
## [1] "Columns dropped:"
##  [1] "Program"          "Period"           "Week Ending"      "Ag District"
##  [5] "Ag District Code" "County"           "County ANSI"      "Zip Code"
##  [9] "Region"           "watershed_code"   "Watershed"        "Commodity"
## [13] "Domain"           "Domain Category"
```

```r
# Step 4: Split composite columns (e.g., "Data Item")
# Use a more flexible splitting approach to handle inconsistencies
organic_strawberry <- organic_strawberry %>%
  separate(`Data Item`, into = c("Fruit", "Category", "Item", "Metric"), sep = ",", extra = "merge", fi

# Step 5: Clean up leading/trailing spaces in the new columns
organic_strawberry <- organic_strawberry %>%
  mutate(across(c(Category, Item, Metric), ~ str_trim(., side = "both")))

# Step 6: Handle non-numeric values in the 'Value' column
# Convert non-numeric entries like (D), (H) to NA and remove commas
organic_strawberry <- organic_strawberry %>%
  mutate(Value = as.numeric(str_replace_all(Value, "[^0-9]", NA_character_)))

# Step 7: Save the cleaned organic strawberries data to a new CSV file
write_csv(organic_strawberry, "organic_strawberries_cleaned.csv")

# Step 8: Display a sample of the cleaned data for verification
print(head(organic_strawberry))
```

```
## # A tibble: 6 x 10
##    Year `Geo Level` State    `State ANSI` Fruit      Category Item  Metric Value
##   <dbl> <chr>       <chr>    <chr>        <chr>      <chr>    <chr> <chr>  <dbl>
## 1  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ <NA>  <NA>      NA
## 2  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ <NA>  <NA>     546
## 3  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ <NA>  <NA>     546
## 4  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ MEAS~ <NA>      NA
## 5  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ MEAS~ <NA>      NA
## 6  2021 NATIONAL    US TOTAL <NA>         STRAWBERR~ ORGANIC~ MEAS~ <NA>      NA
## # i 1 more variable: `CV (%)` <chr>
```

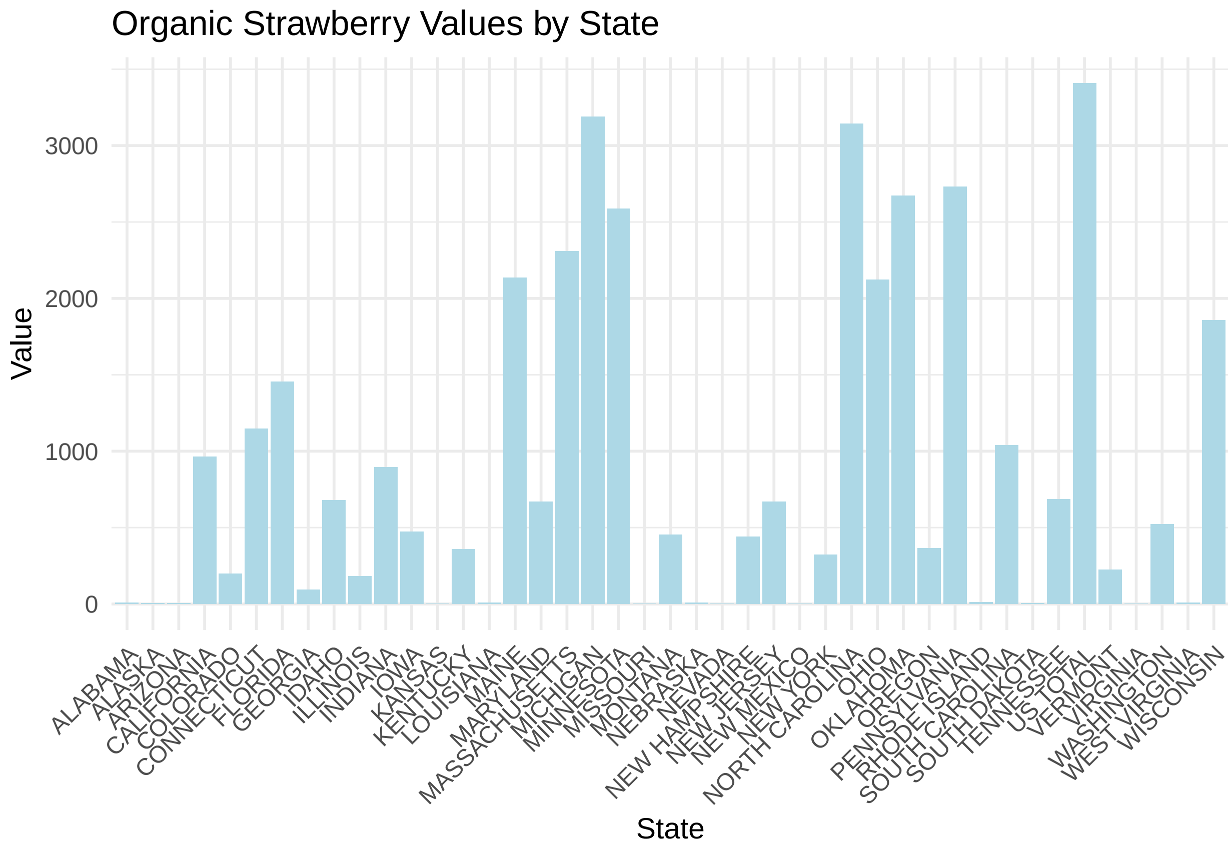
```r
# Optional: Check the summary to verify that the 'Value' column is numeric and other columns are correc
summary(organic_strawberry)
```

```
##       Year       Geo Level            State            State ANSI
##  Min.   :2019   Length:732         Length:732         Length:732
##  1st Qu.:2019   Class :character   Class :character   Class :character
##  Median :2019   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2020
##  3rd Qu.:2021
##  Max.   :2021
##
##     Fruit             Category             Item              Metric
##  Length:732         Length:732         Length:732         Length:732
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      Value           CV (%)
##  Min.   :  1.00   Length:732
##  1st Qu.:  4.00   Class :character
##  Median : 14.00   Mode  :character
##  Mean   : 95.23
##  3rd Qu.: 74.00
##  Max.   :880.00
##  NA's   :332
```

```r
# Load required library
library(ggplot2)

# Step 1: Filter the data to make sure you focus on relevant rows (e.g., exclude rows with NA in Value)
organic_strawberry_state <- organic_strawberry %>%
  filter(!is.na(Value), !is.na(State))  # Ensure Value and State are not NA

# Step 2: Plot the bar graph of Value by State
ggplot(organic_strawberry_state, aes(x = State, y = Value)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Organic Strawberry Values by State",
       x = "State",
       y = "Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for clarity
```



I collected the organic strawberry data from the dataset and cleaned it as I would do in the previous dataset. I then looked at what states produce the most amount of organic strawberries.