615 Assignment Strawberry 2

Yiming Chen

2024-10-21

#EDA

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

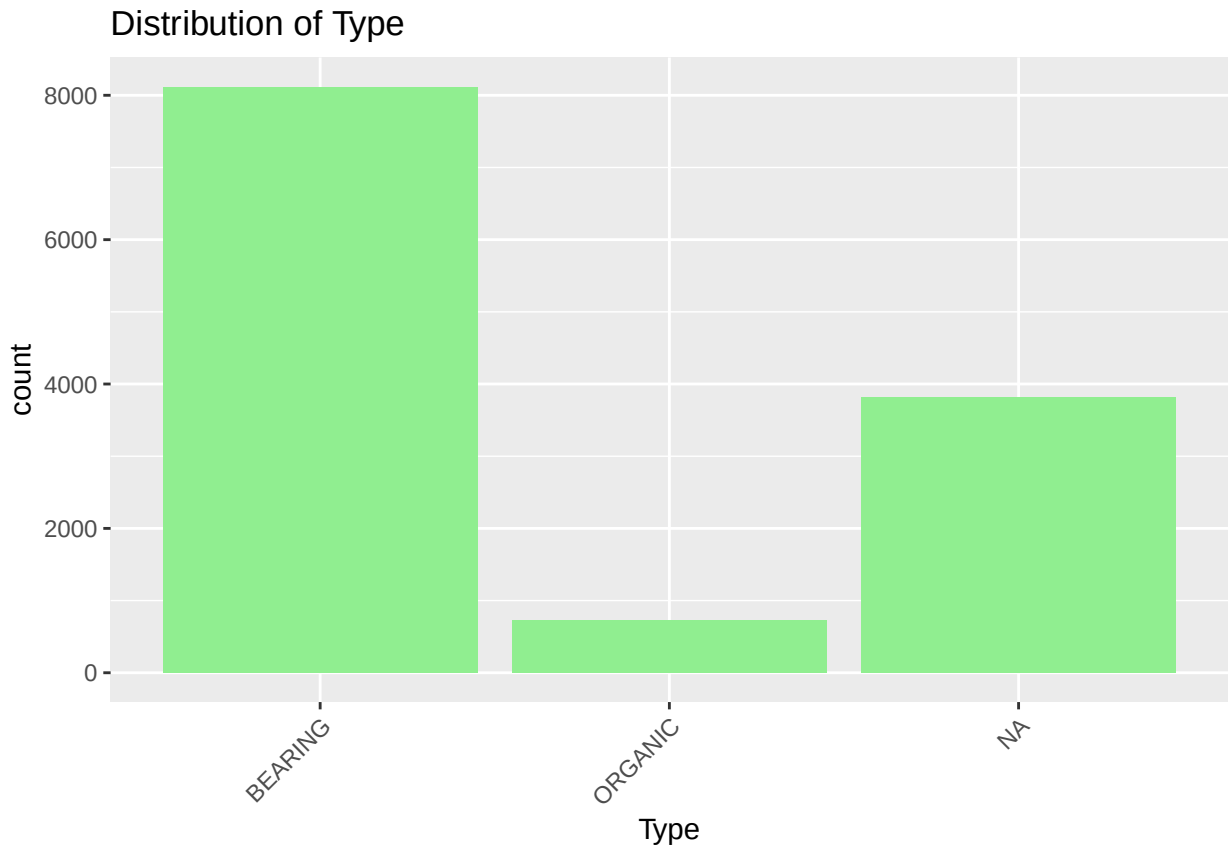
```
library(ggplot2)
strawberry = read.csv("cleaned_strawberries.csv")
view(strawberry)
```

```
# Check data types
str(strawberry)
```

```
## 'data.frame':    12669 obs. of  29 variables:
##  $ Program         : chr  "CENSUS" "CENSUS" "CENSUS" "CENSUS" ...
##  $ Year            : int  2022 2022 2022 2022 2022 2022 2022 2022 2022 2022 ...
##  $ Period          : chr  "YEAR" "YEAR" "YEAR" "YEAR" ...
##  $ Week.Ending     : logi  NA NA NA NA NA NA ...
##  $ Geo.Level       : chr  "COUNTY" "COUNTY" "COUNTY" "COUNTY" ...
##  $ State           : chr  "ALABAMA" "ALABAMA" "ALABAMA" "ALABAMA" ...
##  $ State.ANSI      : int  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 ...
##  $ County          : chr  "BULLOCK" "BULLOCK" "BULLOCK" "BULLOCK" ...
##  $ County.ANSI     : int  11 11 11 11 11 11 101 101 101 101 ...
##  $ 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 - ACRES BEARING" "STRAWBERRIES - ACRES GROWN" "STRAWBERRIES -
##  $ Domain          : chr  "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
##  $ Domain.Category : chr  "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" ...
##  $ Value           : chr  NA "3" NA "1" ...
##  $ CV....          : chr  NA "15.7" NA "(L)" ...
```

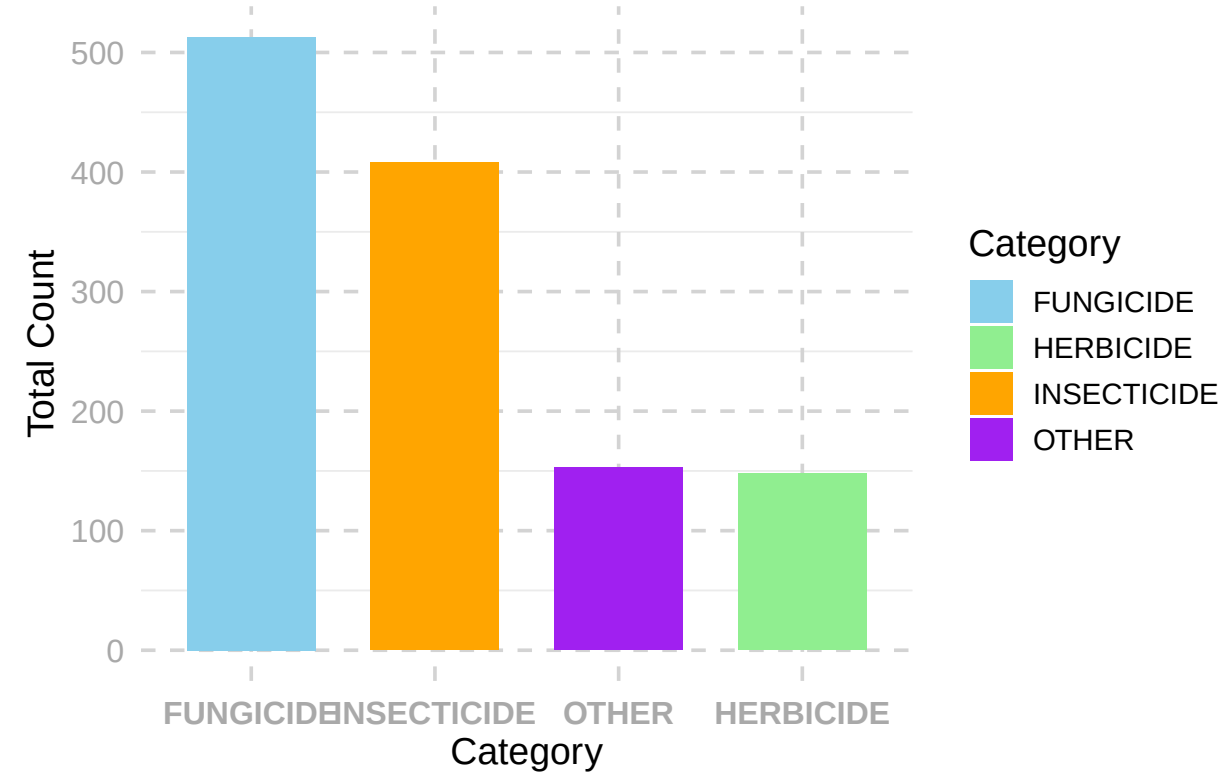
```
##  $ Category        : chr  NA NA NA NA ...
##  $ Name            : chr  NA NA NA NA ...
##  $ Number          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Min             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Max             : chr  NA NA NA NA ...
##  $ Unit            : chr  NA NA NA NA ...
##  $ Type            : chr  "BEARING" NA "BEARING" "BEARING" ...
##  $ Operation       : chr  "ACRES" "ACRES GROWN" "ACRES NON" "OPERATIONS WITH AREA" ...
```

```r
# Bar plot for 'Type' column
ggplot(strawberry, aes(x=Type)) +
  geom_bar(fill="lightgreen") +
  theme(axis.text.x = element_text(angle=45, hjust=1)) +
  labs(title="Distribution of Type")
```



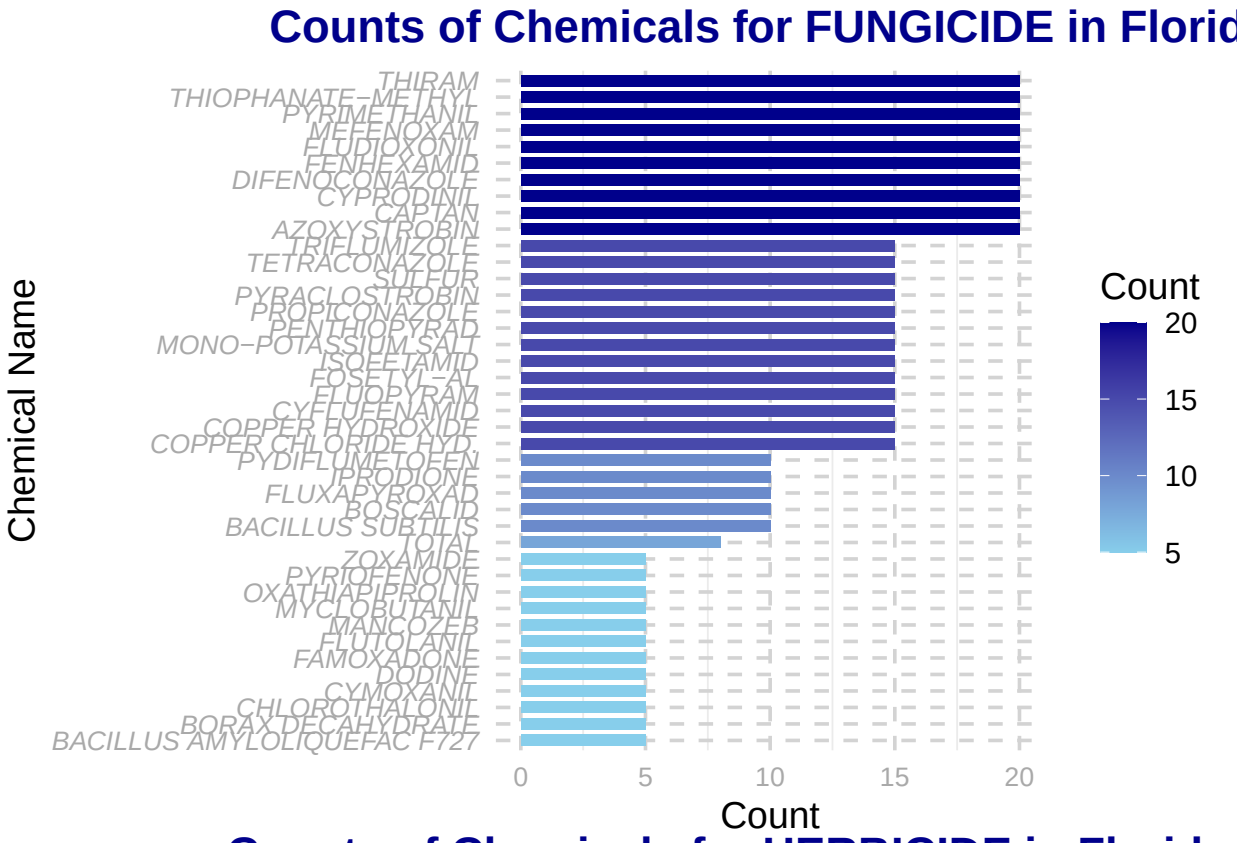

```r
filtered_data <- strawberry %>%
  filter(State == "FLORIDA" &
         Category %in% c("FUNGICIDE", "OTHER", "HERBICIDE", "INSECTICIDE"))

# Count the total number of occurrences of chemicals in each category
category_total_counts <- filtered_data %>%
  group_by(Category) %>%
  summarise(Total_Count = n()) %>%
  arrange(desc(Total_Count))

# Create a bar chart for the total counts of each category
ggplot(category_total_counts, aes(x = reorder(Category, -Total_Count), y = Total_Count, fill = Category
  geom_bar(stat = "identity", width = 0.7) +
```

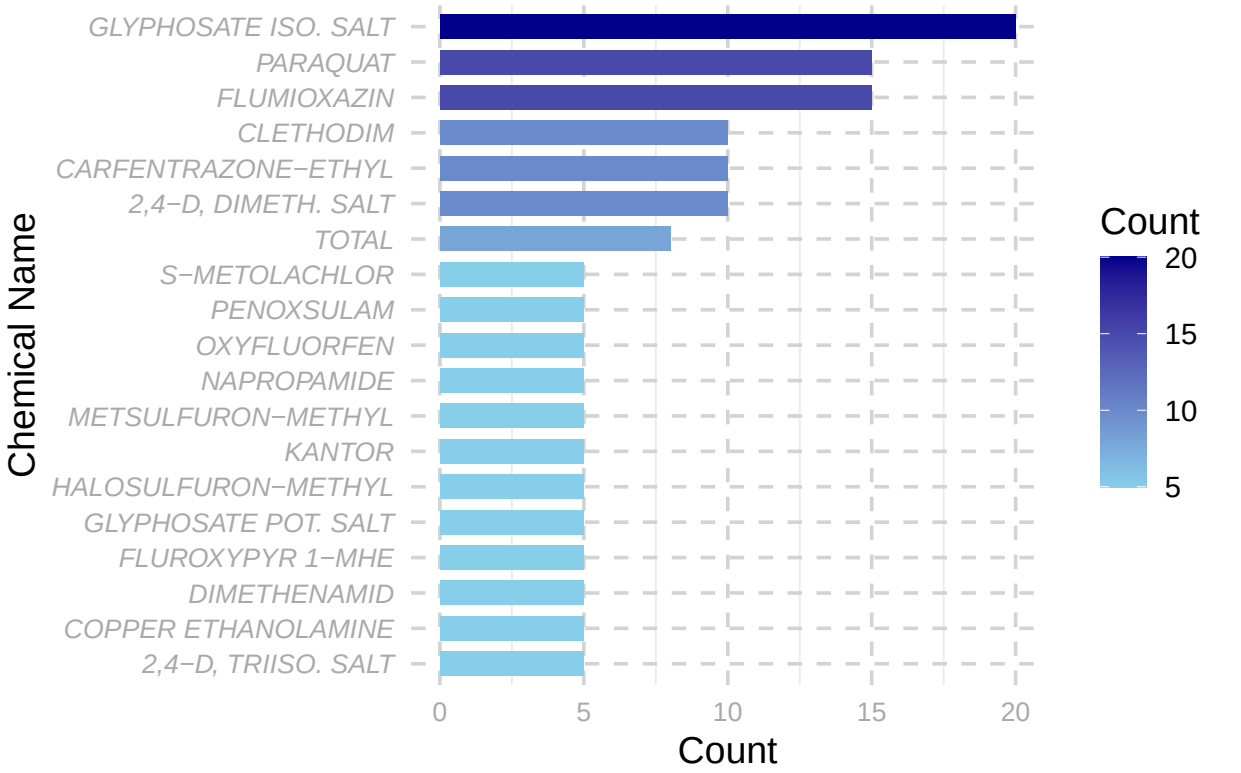
```
  scale_fill_manual(values = c("FUNGICIDE" = "skyblue", "HERBICIDE" = "lightgreen",
                               "INSECTICIDE" = "orange", "OTHER" = "purple")) +  # Custom colors for ea
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5, vjust = 0.5, face = "bold"),  # Style x-axis
        axis.text = element_text(size = 12, color = "darkgray"),  # Adjust font size and color
        plot.title = element_text(hjust = 0.5, face = "bold", color = "darkblue"),  # Center and style
        panel.grid.major = element_line(color = "lightgray", linetype = "dashed")) +  # Dashed grid lin
  labs(title = "Total Count of Chemicals by Category in Florida",
       x = "Category", y = "Total Count")
```

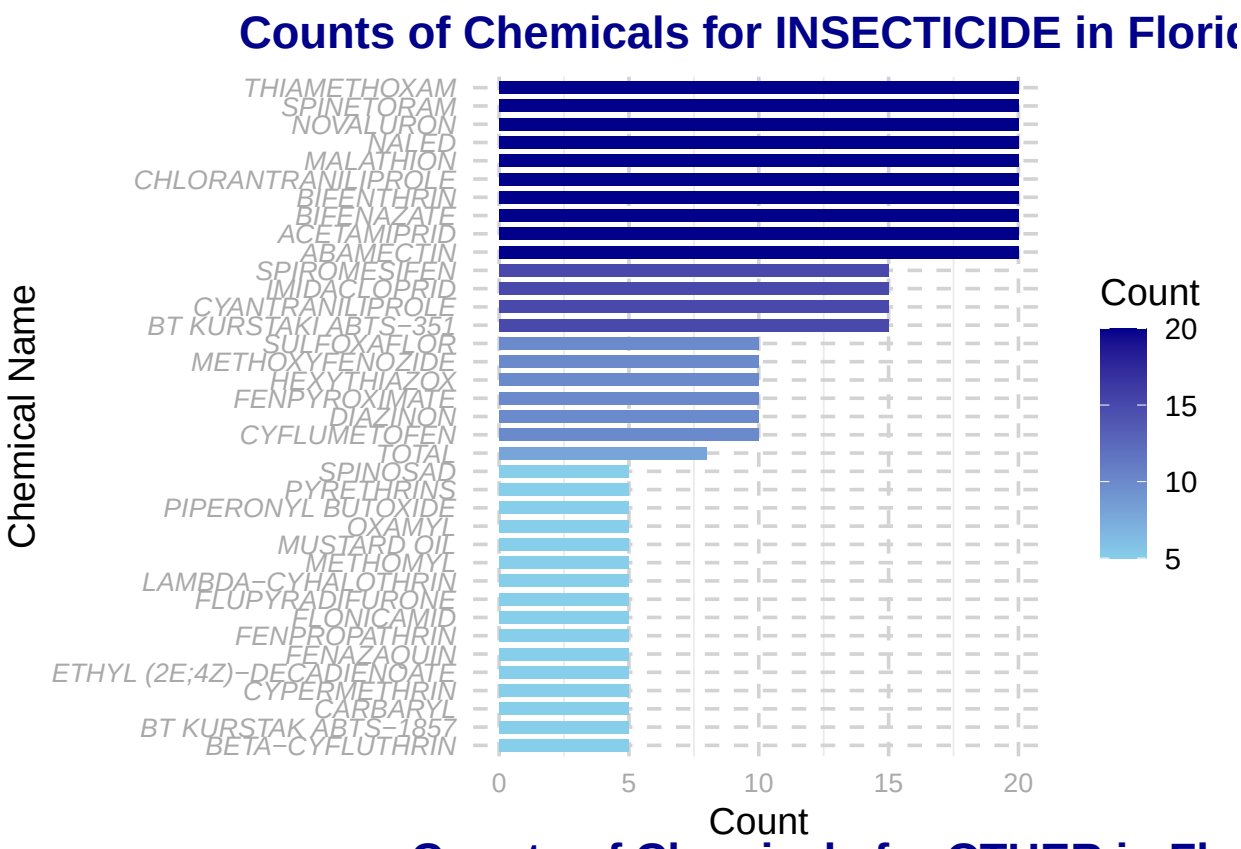


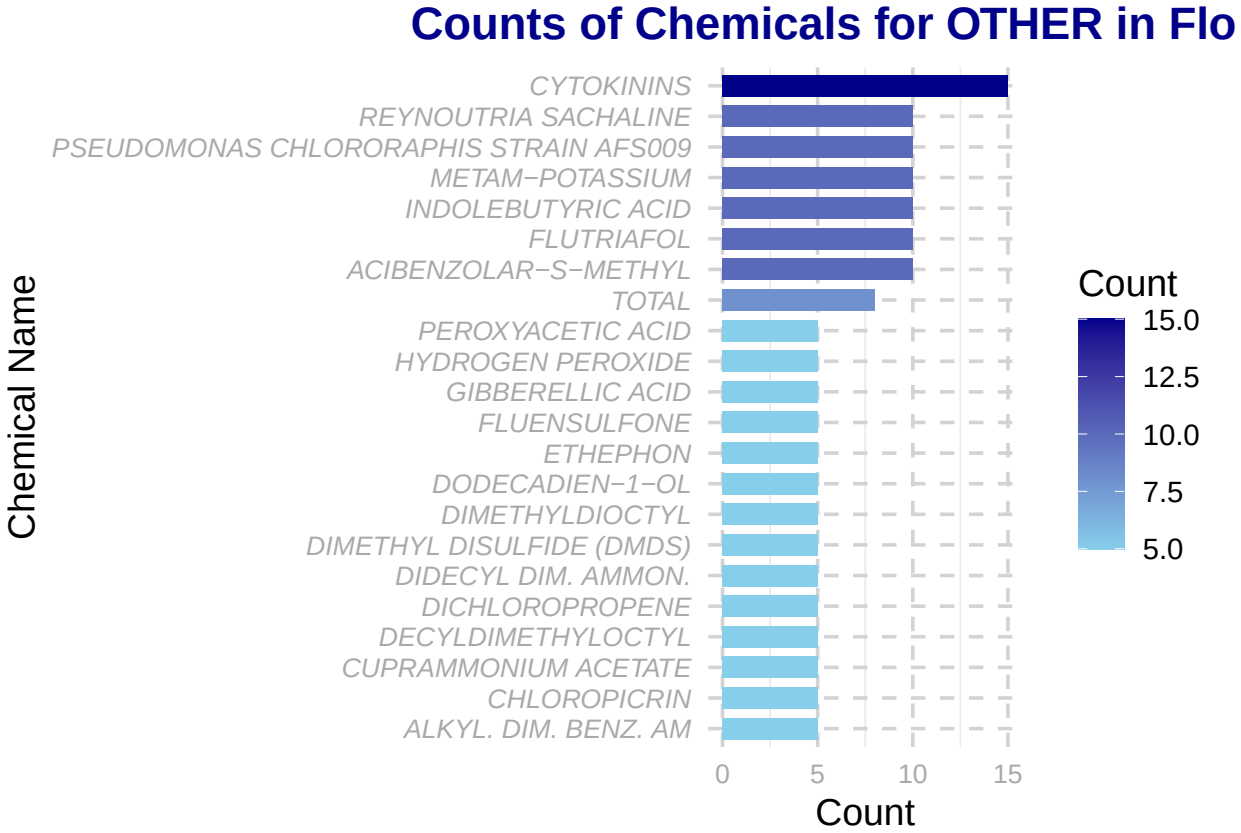

```
# Filter data to include only the categories FUNGICIDE, OTHER, HERBICIDE, INSECTICIDE,
# State = New York, and Program = SURVEY
filtered_data <- strawberry %>%
  filter(State == "FLORIDA" & Program == "SURVEY" &
         Category %in% c("FUNGICIDE", "OTHER", "HERBICIDE", "INSECTICIDE"))

# Count the number of occurrences of each chemical name within each category
category_chemical_counts <- filtered_data %>%
  group_by(Category, Name) %>%
  summarise(Count = n()) %>%
  arrange(Category, desc(Count))

# Create a function to plot bar chart for each category
plot_category <- function(category_name) {
  subset_data <- category_chemical_counts %>%
    filter(Category == category_name)
```

```r
  # Check if there's data to plot
  if(nrow(subset_data) == 0) {
    message(paste("No data available for category:", category_name))
    return(NULL)
  }

  ggplot(subset_data, aes(y = reorder(Name, Count), x = Count, fill = Count)) +
    geom_bar(stat = "identity", width = 0.7) +
    scale_fill_gradient(low = "skyblue", high = "darkblue") +  # Add a gradient fill
    theme_minimal(base_size = 14) +
    theme(axis.text.y = element_text(angle = 0, hjust = 1, vjust = 0.5, face = "italic"),  # Style y-ax
          axis.text = element_text(size = 10, color = "darkgray"),  # Adjust font size and color
          plot.title = element_text(hjust = 0.5, face = "bold", color = "darkblue"),  # Center and styl
          panel.grid.major = element_line(color = "lightgray", linetype = "dashed")) +  # Dashed grid l
    labs(title = paste("Counts of Chemicals for", category_name, "in Florida"),
         y = "Chemical Name", x = "Count")
}

# Generate and print plots for each category
categories <- c("FUNGICIDE", "HERBICIDE", "INSECTICIDE", "OTHER")
plots <- lapply(categories, plot_category)

# Print only plots that were successfully created
for (plot in plots) {
  if (!is.null(plot)) {
    print(plot)
  }
}
```

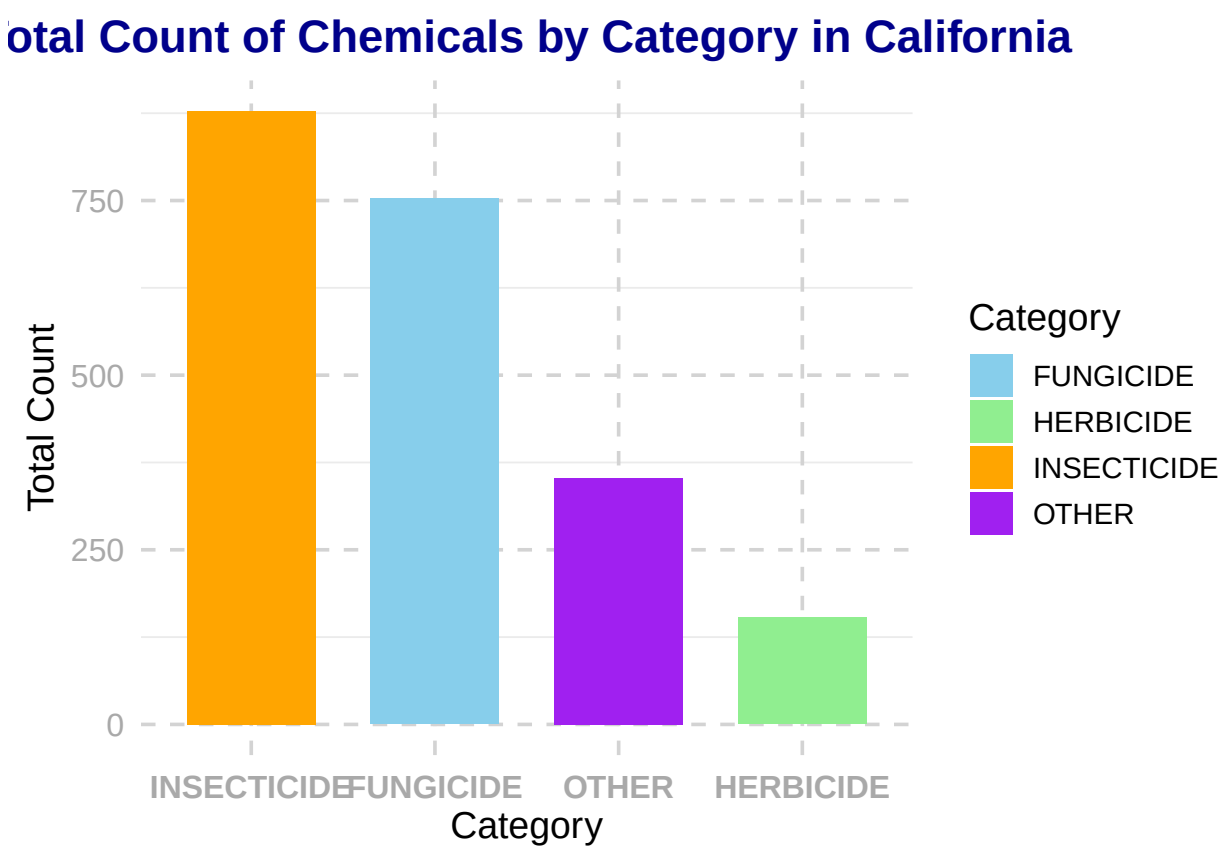

Counts of Chemicals for FUNGICIDE in Florid
Chemical Name
THIRAM
THIOPHANATE−METHYL
PYRIMETHANIL
MEFENOXAM
FLUDIOXONIL
FENHEXAMID
DIFENOCONAZOLE
CYPRODINIL
CAPTAN
AZOXYSTROBIN
TRIFLUMIZOLE
TETRACONAZOLE
SULFUR
PYRACLOSTROBIN
PROPICONAZOLE
PENTHIOPYRAD
MONO−POTASSIUM SALT
ISOFETAMID
FOSETYL−AL
FLUOPYRAM
CYFLUFENAMID
COPPER HYDROXIDE
COPPER CHLORIDE HYD.
PYDIFLUMETOFEN
IPRODIONE
FLUXAPYROXAD
BOSCALID
BACILLUS SUBTILIS
TOTAL
ZOXAMIDE
PYRIOFENONE
OXATHIAPIPROLIN
MYCLOBUTANIL
MANCOZEB
FLUTOLANIL
FAMOXADONE
DODINE
CYMOXANIL
CHLOROTHALONIL
BORAX DECAHYDRATE
BACILLUS AMYLOLIQUEFAC F727
0
5
10
15
20
Count
Count
20
15
10
5


Counts of Chemicals for HERBICIDE in Florida
Chemical Name
GLYPHOSATE ISO. SALT
PARAQUAT
FLUMIOXAZIN
CLETHODIM
CARFENTRAZONE−ETHYL
2,4−D, DIMETH. SALT
TOTAL
S−METOLACHLOR
PENOXSULAM
OXYFLUORFEN
NAPROPAMIDE
METSULFURON−METHYL
KANTOR
HALOSULFURON−METHYL
GLYPHOSATE POT. SALT
FLUROXYPYR 1−MHE
DIMETHENAMID
COPPER ETHANOLAMINE
2,4−D, TRIISO. SALT
0
5
10
15
20
Count
Count
20
15
10
5


Counts of Chemicals for INSECTICIDE in Floric
Chemical Name
THIAMETHOXAM
SPINETORAM
NOVALURON
NALED
MALATHION
CHLORANTRANILIPROLE
BIFENTHRIN
BIFENAZATE
ACETAMIPRID
ABAMECTIN
SPIROMESIFEN
IMIDACLOPRID
CYANTRANILIPROLE
BT KURSTAKI ABTS−351
SULFOXAFLOR
METHOXYFENOZIDE
HEXYTHIAZOX
FENPYROXIMATE
DIAZINON
CYFLUMETOFEN
TOTAL
SPINOSAD
PYRETHRINS
PIPERONYL BUTOXIDE
OXAMYL
MUSTARD OIL
METHOMYL
LAMBDA−CYHALOTHRIN
FLUPYRADIFURONE
FLONICAMID
FENPROPATHRIN
FENAZAQUIN
ETHYL (2E;4Z)−DECADIENOATE
CYPERMETHRIN
CARBARYL
BT KURSTAK ABTS−1857
BETA−CYFLUTHRIN
0
5
10
15
20
Count
Count
20
15
10
5


Counts of Chemicals for OTHER in Flo
Chemical Name
CYTOKININS
REYNOUTRIA SACHALINE
PSEUDOMONAS CHLORORAPHIS STRAIN AFS009
METAM−POTASSIUM
INDOLEBUTYRIC ACID
FLUTRIAFOL
ACIBENZOLAR−S−METHYL
TOTAL
PEROXYACETIC ACID
HYDROGEN PEROXIDE
GIBBERELLIC ACID
FLUENSULFONE
ETHEPHON
DODECADIEN−1−OL
DIMETHYLDIOCTYL
DIMETHYL DISULFIDE (DMDS)
DIDECYL DIM. AMMON.
DICHLOROPROPENE
DECYLDIMETHYLOCTYL
CUPRAMMONIUM ACETATE
CHLOROPICRIN
ALKYL. DIM. BENZ. AM
0
5
10
15
Count
Count
15.0
12.5
10.0
7.5
5.0

```r
filtered_data <- strawberry %>%
  filter(State == "CALIFORNIA" &
         Category %in% c("FUNGICIDE", "OTHER", "HERBICIDE", "INSECTICIDE"))

# Count the total number of occurrences of chemicals in each category
category_total_counts <- filtered_data %>%
  group_by(Category) %>%
  summarise(Total_Count = n()) %>%
  arrange(desc(Total_Count))

# Create a bar chart for the total counts of each category
ggplot(category_total_counts, aes(x = reorder(Category, -Total_Count), y = Total_Count, fill = Category)
  geom_bar(stat = "identity", width = 0.7) +
  scale_fill_manual(values = c("FUNGICIDE" = "skyblue", "HERBICIDE" = "lightgreen",
                               "INSECTICIDE" = "orange", "OTHER" = "purple")) +  # Custom colors for ea
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5, vjust = 0.5, face = "bold"),  # Style x-axis
        axis.text = element_text(size = 12, color = "darkgray"),  # Adjust font size and color
        plot.title = element_text(hjust = 0.5, face = "bold", color = "darkblue"),  # Center and style
        panel.grid.major = element_line(color = "lightgray", linetype = "dashed")) +  # Dashed grid lin
  labs(title = "Total Count of Chemicals by Category in California",
       x = "Category", y = "Total Count")
```

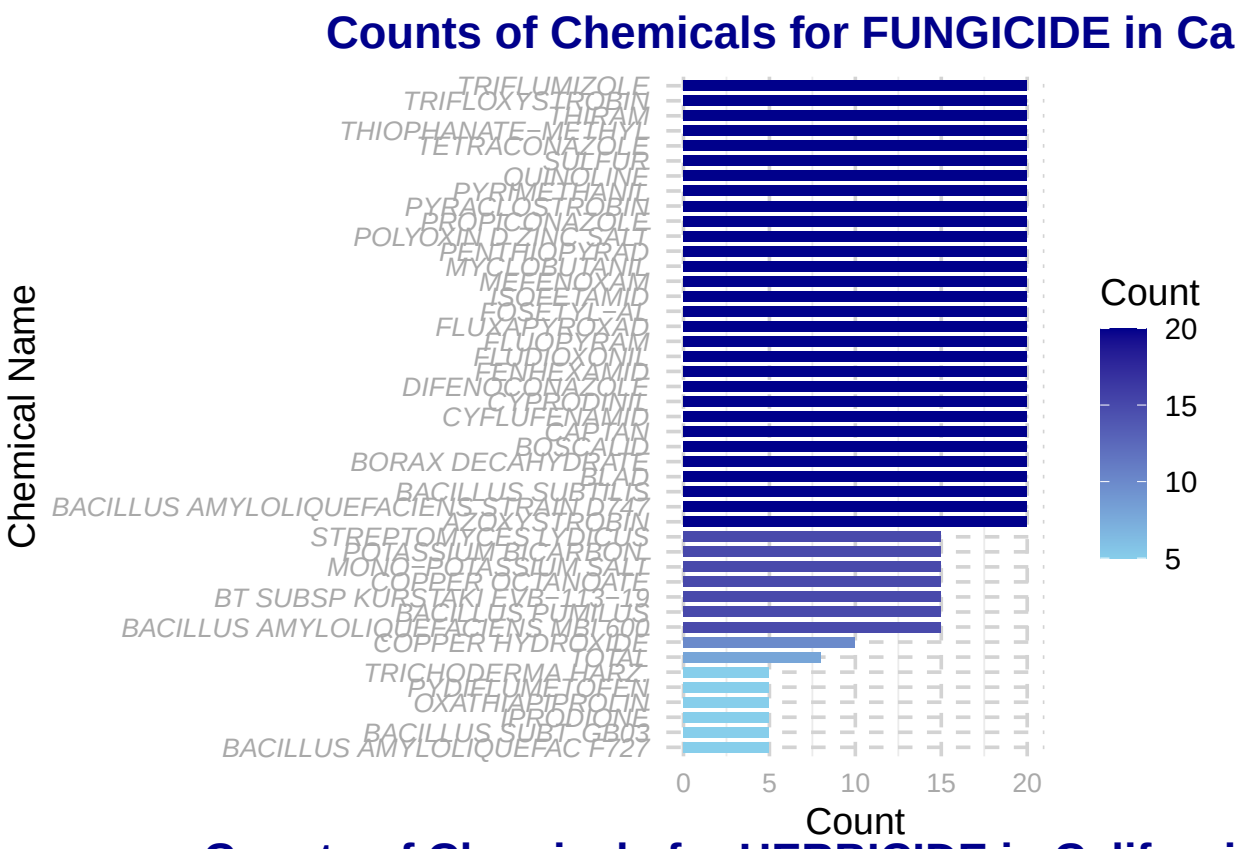


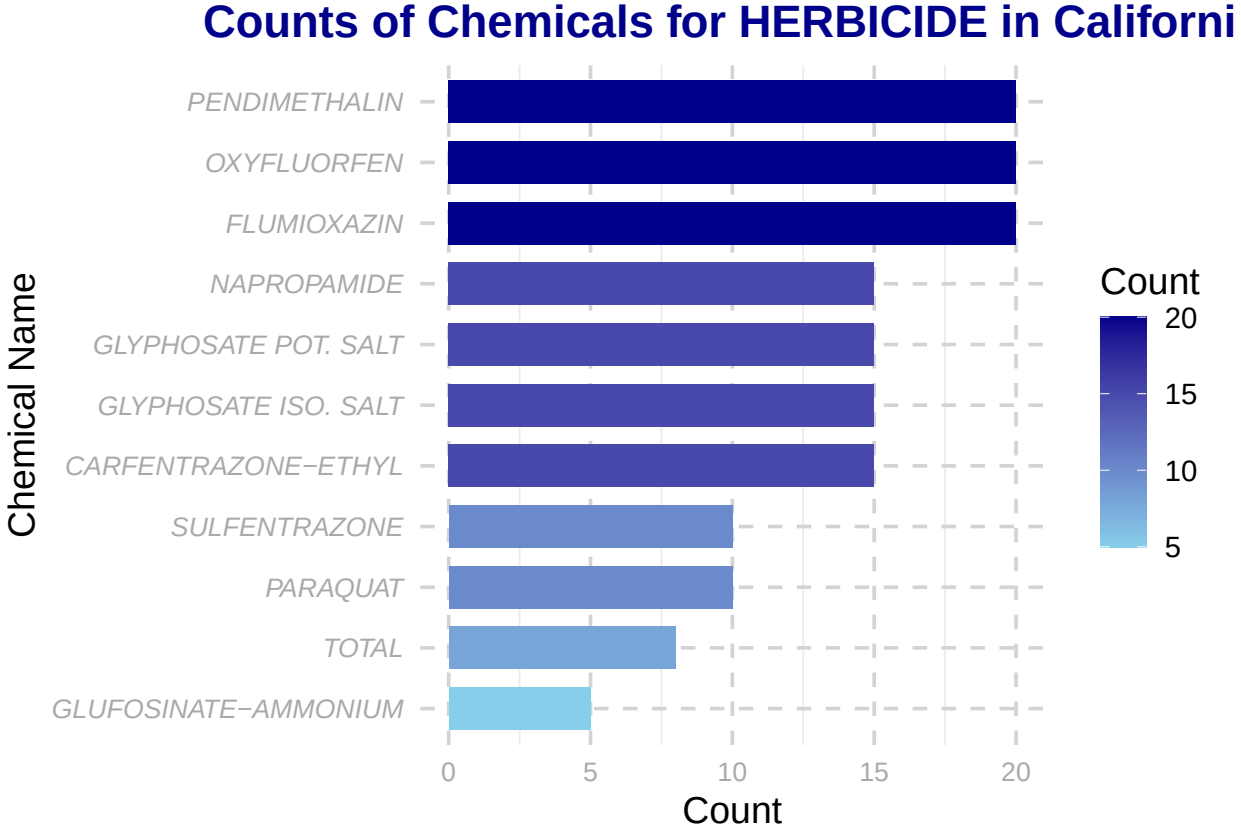

```r
# Filter data to include only the categories FUNGICIDE, OTHER, HERBICIDE, INSECTICIDE,
# State = New York, and Program = SURVEY
filtered_data <- strawberry %>%
  filter(State == "CALIFORNIA" & Program == "SURVEY" &
```

```
        Category %in% c("FUNGICIDE", "OTHER", "HERBICIDE", "INSECTICIDE"))

# Count the number of occurrences of each chemical name within each category
category_chemical_counts <- filtered_data %>%
  group_by(Category, Name) %>%
  summarise(Count = n()) %>%
  arrange(Category, desc(Count))
```

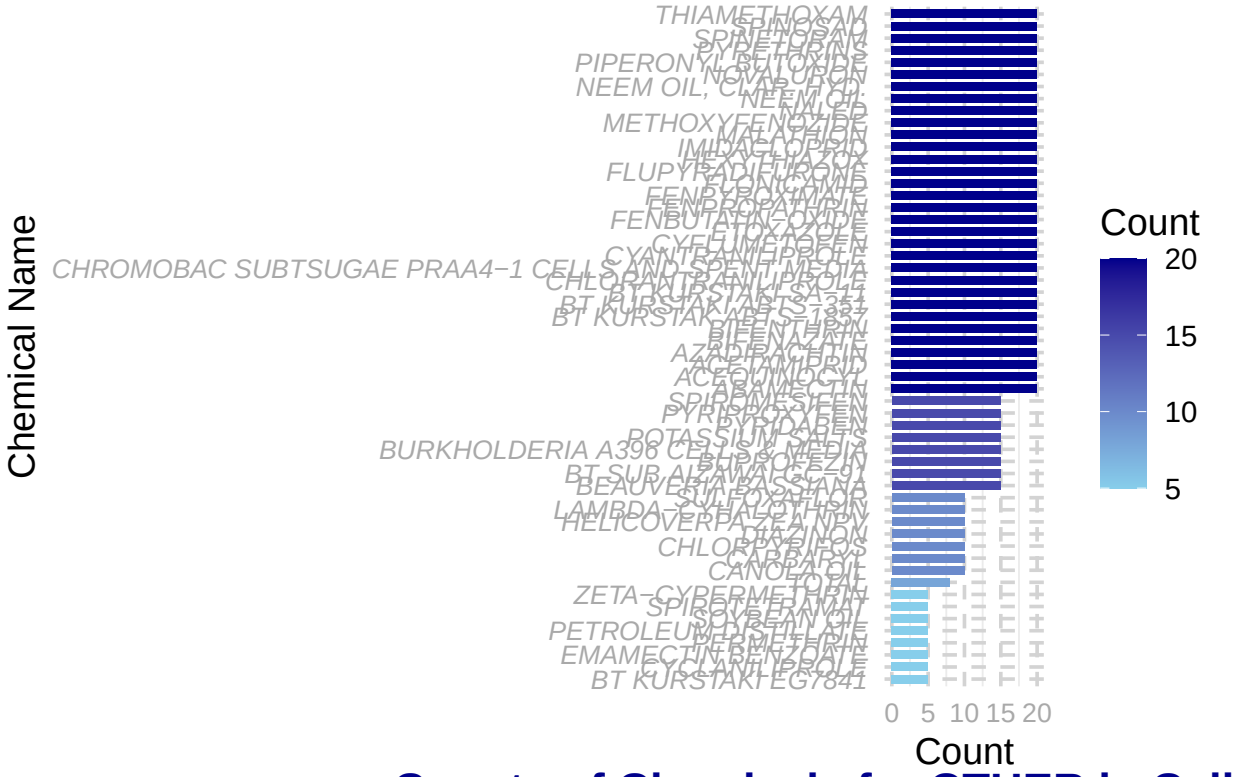
```
## `summarise()` has grouped output by 'Category'. You can override using the
## `.groups` argument.
```

```
# Create a function to plot bar chart for each category
plot_category <- function(category_name) {
  subset_data <- category_chemical_counts %>%
    filter(Category == category_name)

  # Check if there's data to plot
  if(nrow(subset_data) == 0) {
    message(paste("No data available for category:", category_name))
    return(NULL)
  }

  ggplot(subset_data, aes(y = reorder(Name, Count), x = Count, fill = Count)) +
    geom_bar(stat = "identity", width = 0.7) +
    scale_fill_gradient(low = "skyblue", high = "darkblue") +  # Add a gradient fill
    theme_minimal(base_size = 14) +
    theme(axis.text.y = element_text(angle = 0, hjust = 1, vjust = 0.5, face = "italic"),  # Style y-ax
          axis.text = element_text(size = 10, color = "darkgray"),  # Adjust font size and color
          plot.title = element_text(hjust = 0.5, face = "bold", color = "darkblue"),  # Center and styl
          panel.grid.major = element_line(color = "lightgray", linetype = "dashed")) +  # Dashed grid l
    labs(title = paste("Counts of Chemicals for", category_name, "in California"),
         y = "Chemical Name", x = "Count")
}

# Generate and print plots for each category
categories <- c("FUNGICIDE", "HERBICIDE", "INSECTICIDE", "OTHER")
plots <- lapply(categories, plot_category)

# Print only plots that were successfully created
for (plot in plots) {
  if (!is.null(plot)) {
    print(plot)
  }
}
```

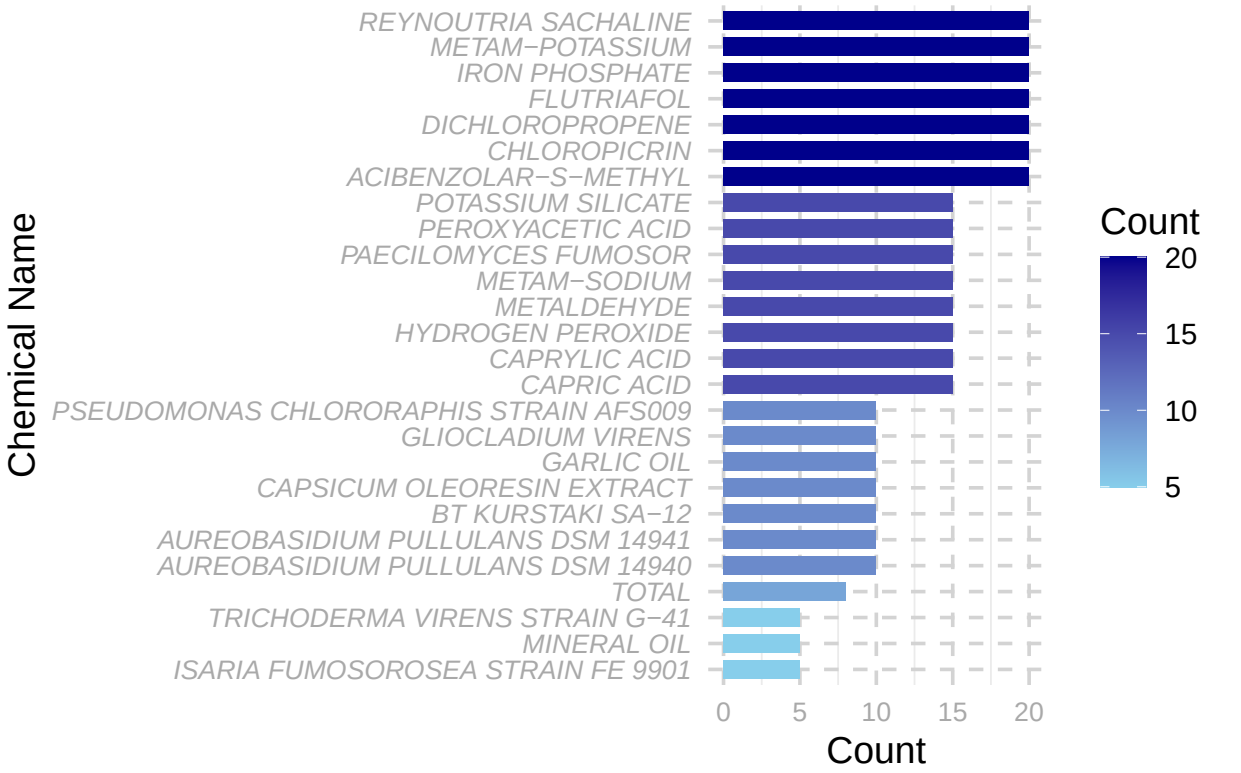

Counts of Chemicals for FUNGICIDE in Ca
Chemical Name
TRIFLUMIZOLE
TRIFLOXYSTROBIN
THIRAM
THIOPHANATE−METHYL
TETRACONAZOLE
SULFUR
QUINOLINE
PYRIMETHANIL
PYRACLOSTROBIN
PROPICONAZOLE
POLYOXIN D ZINC SALT
PENTHIOPYRAD
MYCLOBUTANIL
MEFENOXAM
ISOPYRAZAM
FOSETYL−AL
FLUXAPYROXAD
FLUOPYRAM
FLUDIOXONIL
FENHEXAMID
DIFENOCONAZOLE
CYPRODINIL
CYFLUFENAMID
CAPTAN
BOSCALID
BORAX DECAHYDRATE
BLAD
BACILLUS SUBTILIS
BACILLUS AMYLOLIQUEFACIENS STRAIN D747
AZOXYSTROBIN
STREPTOMYCES LYDICUS
POTASSIUM BICARBONATE
MONO−POTASSIUM SALT
COPPER OCTANOATE
BT SUBSP KURSTAKI EVB−113−19
BACILLUS PUMILUS
BACILLUS AMYLOLIQUEFACIENS MBI 600
COPPER HYDROXIDE
TOTAL
TRICHODERMA HARZ
PYDIFLUMETOFEN
OXATHIAPIPROLIN
IPRODIONE
BACILLUS SUBT GB03
BACILLUS AMYLOLIQUEFAC F727
0
5
10
15
20
Count
Count
20
15
10
5


Counts of Chemicals for HERBICIDE in Californi
Chemical Name
PENDIMETHALIN
OXYFLUORFEN
FLUMIOXAZIN
NAPROPAMIDE
GLYPHOSATE POT. SALT
GLYPHOSATE ISO. SALT
CARFENTRAZONE−ETHYL
SULFENTRAZONE
PARAQUAT
TOTAL
GLUFOSINATE−AMMONIUM
0
5
10
15
20
Count
Count
20
15
10
5

Counts of Chemicals for INSECTICIDE



Counts of Chemicals for OTHER in Cali



```r
library(tidyverse)
library(PubChemR)

GHS_searcher <- function(result_json_object) {
  hierarchies <- result_json_object[["result"]][["Hierarchies"]][["Hierarchy"]]

  for (i in seq_along(hierarchies)) {
    if (hierarchies[[i]][["SourceName"]] == "GHS Classification (UNECE)") {
      return(i)
    }
  }
  # Return NULL if GHS Classification is not found
  return(NULL)
}

hazards_retriever <- function(index, result_json_object) {
  if (is.null(index)) {
    return(NA)  # Return NA if GHS data is not available
  }

  hierarchy <- result_json_object[["result"]][["Hierarchies"]][["Hierarchy"]][[index]]
  nodes <- hierarchy[["Node"]]
  hazard_statements <- c()
  i <- 1

  while (i <= length(nodes) && str_detect(nodes[[i]][["Information"]][["Name"]], "^H")) {
    hazard_statements <- c(hazard_statements, nodes[[i]][["Information"]][["Name"]])
    i <- i + 1
  }
  if (length(hazard_statements) == 0) {
    return(NA)
  }
  return(hazard_statements)
}

# List of chemicals to process
chemical_vec <- c("reynoutria sachaline", "flutriafol", "chloropicrin")

# Initialize an empty list to store results
results_list <- list()

for (chemical in chemical_vec) {
  result <- get_pug_rest(
    identifier = chemical,
    namespace = "name",
    domain = "compound",
    operation = "classification",
    output = "JSON"
  )

  ghs_index <- GHS_searcher(result)
  hazards <- hazards_retriever(ghs_index, result)
```

```
  # Store the results in a list
  results_list[[chemical]] <- hazards
}

# Convert the results list into a data frame
results_df <- results_list %>%
  enframe(name = "Chemical", value = "Hazard_Statements") %>%
  unnest(cols = c(Hazard_Statements))

# Display the data frame
print(results_df)
```

```
## # A tibble: 25 x 2
##    Chemical             Hazard_Statements
##    <chr>                <chr>
##  1 reynoutria sachaline <NA>
##  2 flutriafol           H302: Harmful if swallowed [Warning Acute toxicity, ora~
##  3 flutriafol           H300: Health Hazards
##  4 flutriafol           Hazard Statement Codes
##  5 flutriafol           H312: Harmful in contact with skin [Warning Acute toxic~
##  6 flutriafol           H332: Harmful if inhaled [Warning Acute toxicity, inhal~
##  7 flutriafol           H411: Toxic to aquatic life with long lasting effects [~
##  8 flutriafol           H400: Environmental Hazards
##  9 flutriafol           H412: Harmful to aquatic life with long lasting effects~
## 10 chloropicrin         H301: Toxic if swallowed [Danger Acute toxicity, oral]
## # i 15 more rows
```

Cite: Yibing Wang