Strawberries_HW3

Jie Fei

```
# data cleaning and organization

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

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

```
library(stringr)

# read the strawberry data
strawberry <- read_csv("strawberries25_v3.csv", col_names = TRUE)
```

```
## Rows: 12669 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (12): Program, Period, Geo Level, State, Ag District, County, Commodity,...
## dbl  (5): Year, State ANSI, Ag District Code, County ANSI, watershed_code
## lgl  (4): Week Ending, Zip Code, Region, Watershed
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(strawberry)
```

```
## 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`        <dbl> 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 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`       <dbl> 11, 11, 11, 11, 11, 11, 101, 101, 101, 101, 119, 11…
## $ `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      <dbl> 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,…
## $ 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)",…
```

```
# examine the data. How is it organized?

# 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
```

```
# to get an idea of the data -- looking at california only
calif_census <- strawberry |> filter((State == "CALIFORNIA") & (Program == "CENSUS"))
calif_census <- calif_census |> select(Year, `Data Item`, Value)

calif_survey <- strawberry |> filter((State == "CALIFORNIA") & (Program == "SURVEY"))
calif_survey <- strawberry |> select(Year, Period, `Data Item`, Value)
```

```
# remove columns with a single value in all columns

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"
```

```
drop_one_value_col(strawberry)
```

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

```
# separate composite columns

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

```
# fix the leading space problem

strawberry$Category[1]
```

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

```
# 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")
```

```
# exam the fruit column and find hidden sub-columns

unique(strawberry$Fruit)
```

```
##  [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"
##  [8] "STRAWBERRIES - PRICE RECEIVED"
##  [9] "STRAWBERRIES - ACRES HARVESTED"
## [10] "STRAWBERRIES - ACRES PLANTED"
## [11] "STRAWBERRIES - PRODUCTION"
## [12] "STRAWBERRIES - YIELD"
## [13] "STRAWBERRIES - APPLICATIONS"
## [14] "STRAWBERRIES - TREATED"
```

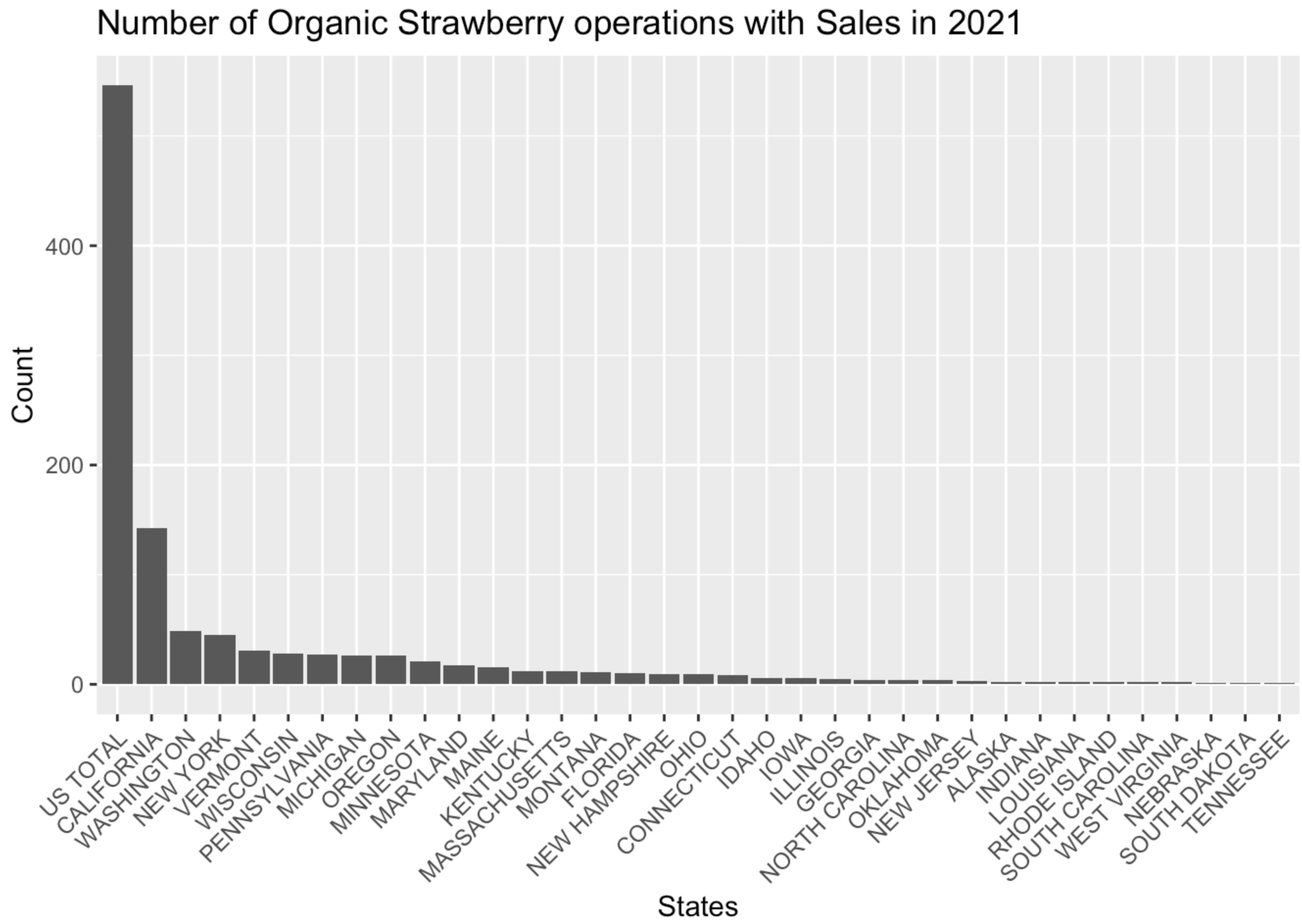
```
# generate a list of rows with the production and price information
spr <- which((strawberry$Fruit == "STRAWBERRIES - PRODUCTION") | (strawberry$Fruit == "S
TRAWBERRIES - PRICE RECEIVED"))
strw_prod_price <- strawberry |> slice(spr)

# this has the census data, too
strw_chem <- strawberry |> slice(-1*spr)  ## too soon
```

```
# exam the rest of columns and split sales and chemicals into two dataframes

strw_b_sales <- strawberry |> filter(Program == "CENSUS")
strw_b_chem <- strawberry |> filter(Program == "SURVEY")
nrow(strawberry) == (nrow(strw_b_chem) + nrow(strw_b_sales))
```

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

```
# export the cleaned strawberry data

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

```
# data analysis and plots

# number of organic strawberry operations with sales in 2021
plot1_data <- strawberry |>
  select(c(Year, State, Category, Value)) |>
  filter((Year == 2021) & (Category == "ORGANIC - OPERATIONS WITH SALES"))

plot1_data$Value <- as.numeric(plot1_data$Value)

plot1_data <- plot1_data |> arrange(desc(Value))

ggplot(plot1_data, aes(x = reorder(State, -Value), y = Value)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45,hjust = 1)) +
  labs(x = "States", y = "Count",
title = "Number of Organic Strawberry operations with Sales in 2021")
```



```
# read the cleaned strawberry and chemical data
strawberry <- read.csv("/Users/jie/Library/CloudStorage/OneDrive-BostonUniversity/Main Folder/02 Courses/PhD (BU)/2024 Fall/MA615 Data Science in R/HW/Strawberry HW/cleaned_strawberry.csv")

# load required package
library(dplyr)
library(tidyr)
library(reader)
```

```
## Loading required package: NCmisc
```

```
##
## Attaching package: 'reader'
```

```
## The following objects are masked from 'package:NCmisc':
##
##     cat.path, get.ext, rmv.ext
```

```
library(ggplot2)
library(tidyverse)
# install.packages('PubChemR')
library(PubChemR)
```

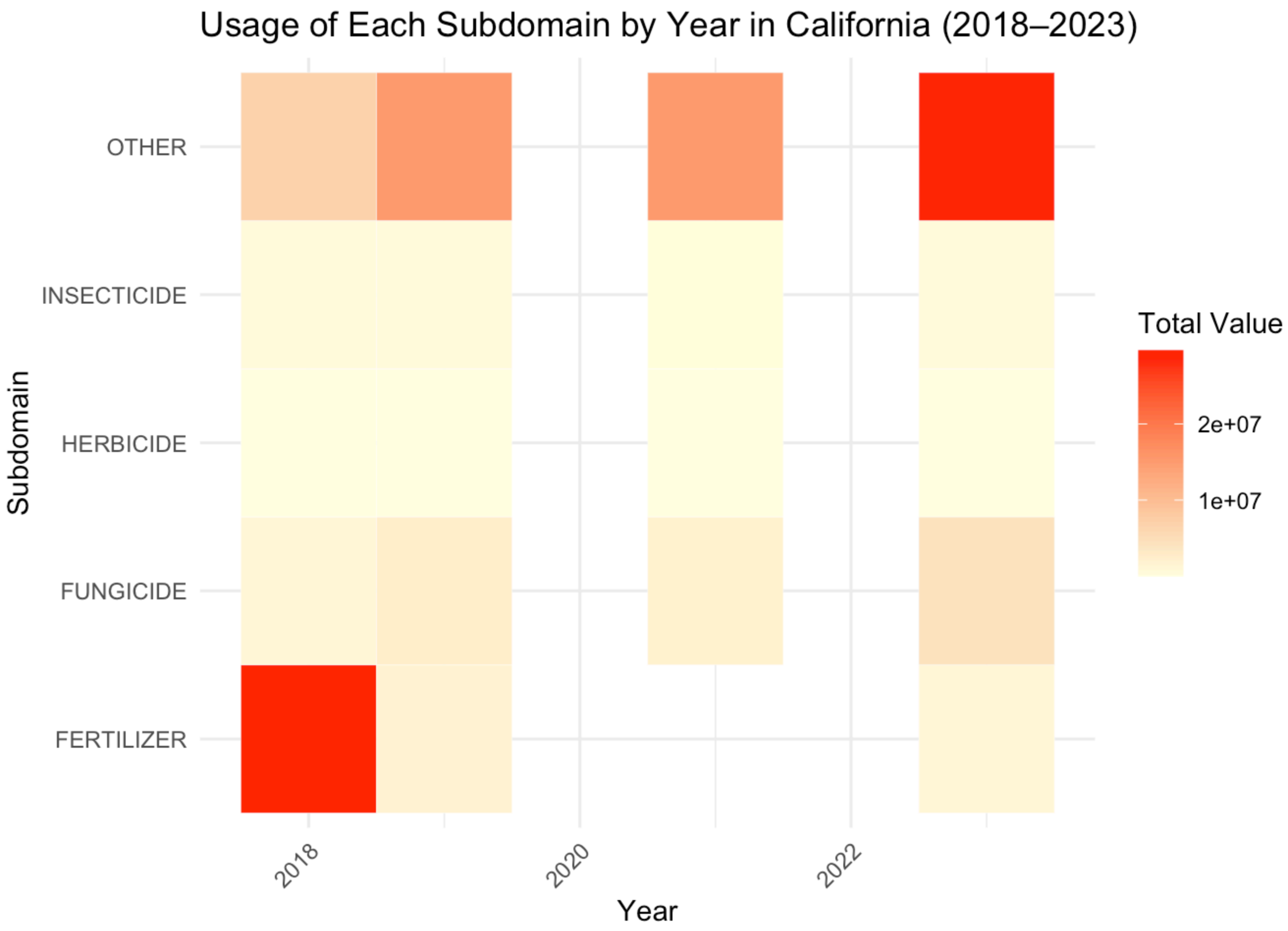
```r
### clean and organize data set as needed

# remove null value where the value column is (D) and (NA)
strawberry <- strawberry[strawberry$Value != "(D)" & strawberry$Value != "(NA)", ]

# keep data only from California and Florida
# the analysis will focus on these two states
strawberry2 <- subset(strawberry, State %in% c("CALIFORNIA", "FLORIDA"))

# create subset for specific domains
strawberry_total <- subset(strawberry2, Domain == "TOTAL")
strawberry_area <- subset(strawberry2, Domain == "AREA GROWN")
strawberry_organic <- subset(strawberry2, Domain == "ORGANIC STATUS")
strawberry_chemical <- subset(strawberry2, !(Domain %in% c("TOTAL", "AREA GROWN", "ORGANIC STATUS")))

# in chemical data set, split the domain column and create two new columns (new domain and sub domain)
strawberry_chemical2 <- strawberry_chemical %>%
  mutate(
    New_Domain = ifelse(grepl(",", Domain), trimws(sapply(strsplit(as.character(Domain), ","), `[`, 1)), NA),
    Sub_Domain = ifelse(grepl(",", Domain), trimws(sapply(strsplit(as.character(Domain), ","), `[`, 2)), Domain)
  )

# split the domain category column and create two new columns (domain category and code)
strawberry_chemical2 <- strawberry_chemical2 %>%
  mutate(
    Domain_Category = ifelse(grepl("=", `Domain.Category`),
                                    trimws(sub(".*\\(([^=]+)=.*\\)", "\\1", `Domain.Category`)),
                                    trimws(sub(".*\\(([^)]+)\\).*", "\\1", `Domain.Category`))),

    Code = ifelse(grepl("=", `Domain.Category`),
                                    trimws(sub(".*=\\s*([^)]+)\\).*", "\\1", `Domain.Category`)),
                                    NA)
  )

# remove old columns
strawberry_chemical2 <- strawberry_chemical2 %>% select(-Domain)
strawberry_chemical2 <- strawberry_chemical2 %>% select(-`Domain.Category`)

# display 5 sample rows to double check
strawberry_chemical2_display <- strawberry_chemical2 %>%
  head(5)

print(strawberry_chemical2_display)
```

```
##      Program Year Period Geo.Level      State State.ANSI Ag.District
## 8763  SURVEY 2023   YEAR     STATE CALIFORNIA          6        <NA>
## 8767  SURVEY 2023   YEAR     STATE CALIFORNIA          6        <NA>
## 8768  SURVEY 2023   YEAR     STATE CALIFORNIA          6        <NA>
## 8770  SURVEY 2023   YEAR     STATE CALIFORNIA          6        <NA>
## 8772  SURVEY 2023   YEAR     STATE CALIFORNIA          6        <NA>
##      Ag.District.Code County County.ANSI        Fruit               Category
## 8763               NA   <NA>          NA STRAWBERRIES BEARING - APPLICATIONS
## 8767               NA   <NA>          NA STRAWBERRIES BEARING - APPLICATIONS
## 8768               NA   <NA>          NA STRAWBERRIES BEARING - APPLICATIONS
## 8770               NA   <NA>          NA STRAWBERRIES BEARING - APPLICATIONS
## 8772               NA   <NA>          NA STRAWBERRIES BEARING - APPLICATIONS
##                Item Metric   Value CV.... New_Domain Sub_Domain
## 8763 MEASURED IN LB   <NA>   3,300   <NA>   CHEMICAL  FUNGICIDE
## 8767 MEASURED IN LB   <NA>   2,800   <NA>   CHEMICAL  FUNGICIDE
## 8768 MEASURED IN LB   <NA>   6,600   <NA>   CHEMICAL  FUNGICIDE
## 8770 MEASURED IN LB   <NA> 603,100   <NA>   CHEMICAL  FUNGICIDE
## 8772 MEASURED IN LB   <NA>  30,300   <NA>   CHEMICAL  FUNGICIDE
##        Domain_Category   Code
## 8763      AZOXYSTROBIN 128810
## 8767 BORAX DECAHYDRATE  11102
## 8768          BOSCALID 128008
## 8770            CAPTAN  81301
## 8772        CYPRODINIL 288202
```

```
### Analysis 1
### What are the total usage pattern of fertilizers and chemicals in each state across different years (2018-2023)?

# read data
strawberry_chemical2_item_mib <- read.csv("strawberry_chemical2_item_mib.csv")

# filter data only for California
california_usage <- strawberry_chemical2_item_mib %>%
  filter(State == "CALIFORNIA") %>%
  group_by(Year, Sub_Domain) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

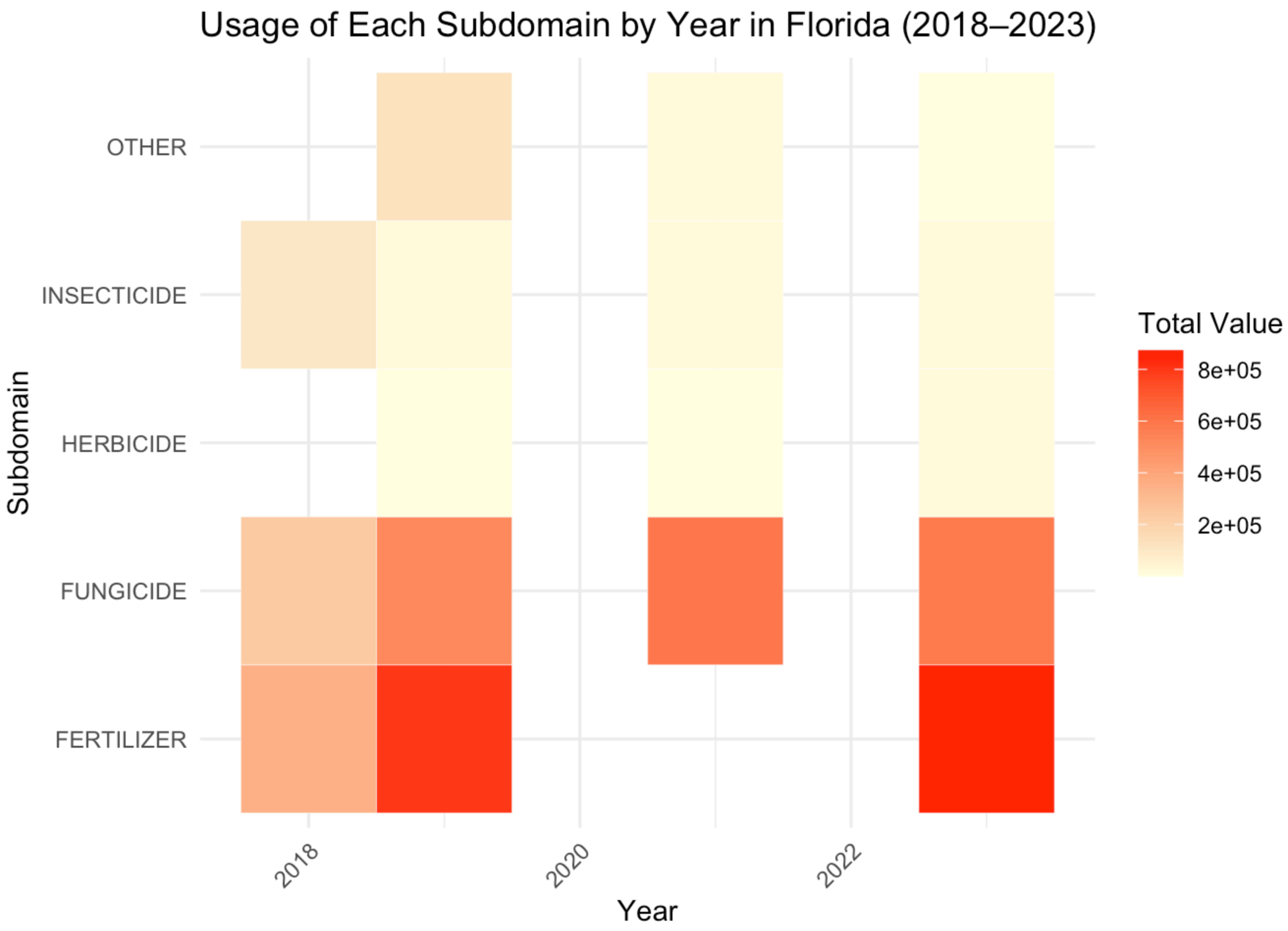
```
# plotting the heat map for California
ggplot(california_usage, aes(x = Year, y = Sub_Domain, fill = Total_Value)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow", high = "red") +
  theme_minimal() +
  labs(
    title = "Usage of Each Subdomain by Year in California (2018–2023)",
    x = "Year",
    y = "Subdomain",
    fill = "Total Value"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# filter data only for Florida
florida_usage <- strawberry_chemical2_item_mib %>%
  filter(State == "FLORIDA") %>%
  group_by(Year, Sub_Domain) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
# plotting the heat map for Florida
ggplot(florida_usage, aes(x = Year, y = Sub_Domain, fill = Total_Value)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow", high = "red") +
  theme_minimal() +
  labs(
    title = "Usage of Each Subdomain by Year in Florida (2018–2023)",
    x = "Year",
    y = "Subdomain",
    fill = "Total Value"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# The figures compare agricultural chemical usage by sub domain (fertilizers, fungicides, herbicides, insecticides, and others) between California and Florida from 2018 to 2023. In California, fertilizers and "other" chemicals show the highest usage, with a consistent trend over the years, while fungicides, herbicides, and insecticides have relatively low and stable usage. The color intensity indicates a broader range of total values, with California’s overall chemical usage exceeding Florida’s.

# In Florida, fertilizer usage is also prominent, especially in 2019 and 2023. Fungicides also show significant usage but at lower levels than fertilizers, while herbicides and insecticides remain minimal across the years. The color scale for Florida highlights a lower total usage range than California, indicating that California uses more agricultural chemicals overall. These differences underscore regional variations in agricultural practices and chemical dependency.
```

```
### Analysis 2
### What are the differences in the usage of each chemical or fertilizer between Florida and California each year?

# filter data for California and Florida, and aggregate by Year, State, and Domain_Category
domain_comparison <- strawberry_chemical2_item_mib %>%
  filter(State %in% c("CALIFORNIA", "FLORIDA")) %>%
  filter(Domain_Category != "TOTAL") %>%
  group_by(Year, State, Domain_Category) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  arrange(Year, Domain_Category, State)
```

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

```
# pivot the data to compare California and Florida side by side
domain_comparison_wide <- domain_comparison %>%
  pivot_wider(names_from = State, values_from = Total_Value, values_fill = 0) %>%
  rename(California_Usage = CALIFORNIA, Florida_Usage = FLORIDA)

print(domain_comparison_wide)
```

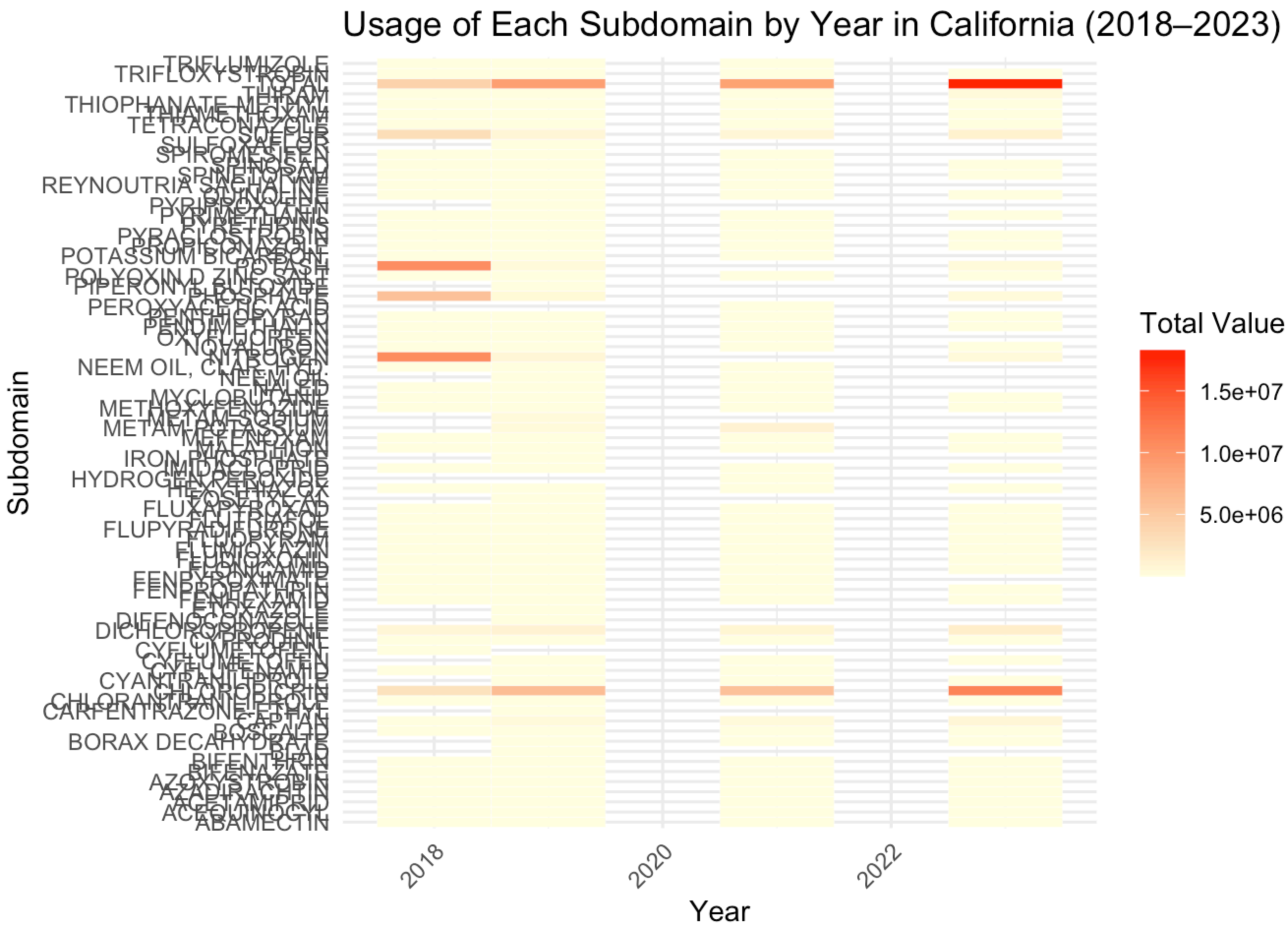
```
## # A tibble: 241 × 4
## # Groups:   Year [4]
##     Year Domain_Category     California_Usage Florida_Usage
##    <int> <chr>                          <int>         <int>
##  1  2018 ABAMECTIN                        200             0
##  2  2018 ACEQUINOCYL                     1400             0
##  3  2018 ACETAMIPRID                     1200             0
##  4  2018 AZADIRACHTIN                     600             0
##  5  2018 AZOXYSTROBIN                    1100             0
##  6  2018 BIFENAZATE                      5100             0
##  7  2018 BIFENTHRIN                      1100             0
##  8  2018 BOSCALID                        1800             0
##  9  2018 CAPTAN                         94800         87400
## 10  2018 CHLORANTRANILIPROLE              600             0
## # ℹ 231 more rows
```

```
# Our results were similar to those in Analysis 2, showing that each state has its own u
nique patterns and trends in chemical and fertilizer usage.
```

```
### Analysis 3
### What are the detailed total usage pattern of fertilizers and chemicals in California
across different years (2018-2023)?

# filter data only for California
california_usage <- strawberry_chemical2_item_mib %>%
  filter(State == "CALIFORNIA") %>%
  group_by(Year, Domain_Category) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
# plotting the heat map for California
ggplot(california_usage, aes(x = Year, y = Domain_Category, fill = Total_Value)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow", high = "red") +
  theme_minimal() +
  labs(
    title = "Usage of Each Subdomain by Year in California (2018-2023)",
    x = "Year",
    y = "Subdomain",
    fill = "Total Value"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# The heat map illustrates the usage of various chemicals and fertilizers sub domains in
California from 2018 to 2023, with each row representing a different sub domain and each
column corresponding to a specific year. The intensity of color (from light yellow to de
ep red) reflects the total usage amount, as indicated by the color scale on the right; d
arker shades represent higher usage levels, with the highest concentrations close to 15
million units shown in dark red. The chart highlights trends in chemicals and fertilizer
s usage, with some chemicals and fertilizers showing significant spikes in specific year
s, marked by intense red blocks, while others have consistently low or no usage across t
he years, represented by lighter yellow or white blocks. Gaps or lighter colors between
years for certain chemicals and fertilizers suggest either low, inconsistent, or non-con
tinuous application of these substances, indicating variable demand or regulatory change
s over time.
```

```
### Analysis 4
### What are the top 3 most used substances in each sub domain each year in California?

# filter out rows where Domain_Category is "TOTAL" and find top 3 substances in each Sub_Domain (chemicals and fertilizers)
top_substances <- strawberry_chemical2_item_mib %>%
  filter(Domain_Category != "TOTAL" & State == "CALIFORNIA") %>%
  group_by(Year, Sub_Domain, Domain_Category) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  arrange(Year, Sub_Domain, desc(Total_Value)) %>%
  group_by(Year, Sub_Domain) %>%
  slice_max(order_by = Total_Value, n = 3)
```

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

```
# display the result
print(top_substances)
```

```
## # A tibble: 56 × 4
## # Groups:   Year, Sub_Domain [19]
##     Year Sub_Domain  Domain_Category      Total_Value
##    <int> <chr>       <chr>                      <int>
##  1  2018 FERTILIZER  NITROGEN                10676000
##  2  2018 FERTILIZER  POTASH                  10583000
##  3  2018 FERTILIZER  PHOSPHATE                5745000
##  4  2018 FUNGICIDE   SULFUR                    298200
##  5  2018 FUNGICIDE   CAPTAN                     94800
##  6  2018 FUNGICIDE   THIRAM                     20100
##  7  2018 HERBICIDE   PENDIMETHALIN               2700
##  8  2018 HERBICIDE   OXYFLUORFEN                  800
##  9  2018 HERBICIDE   FLUMIOXAZIN                  100
## 10  2018 INSECTICIDE NEEM OIL, CLAR. HYD.       90500
## # ℹ 46 more rows
```

```
### Analysis 5
### What are structure/composition/function/potential hazards of the top 3 most used sub
stances in each sub domain for each year in California?

# function 1
GHS_searcher<-function(result_json_object){
  result<-result_json_object
  for (i in 1:length(result[["result"]][["Hierarchies"]][["Hierarchy"]])){
    if(result[["result"]][["Hierarchies"]][["Hierarchy"]][[i]][["SourceName"]]=="GHS Cla
ssification (UNECE)"){
      return(i)
    }
  }
}

# function 2
hazards_retriever<-function(index,result_json_object){
  result<-result_json_object
  hierarchy<-result[["result"]][["Hierarchies"]][["Hierarchy"]][[index]]
  i<-1
  output_list<-rep(NA,length(hierarchy[["Node"]]))
  while(str_detect(hierarchy[["Node"]][[i]][["Information"]][["Name"]],"H") & i<length(h
ierarchy[["Node"]])){
    output_list[i]<-hierarchy[["Node"]][[i]][["Information"]][["Name"]]
    i<-i+1
  }
  return(output_list[!is.na(output_list)])
}

# function to safely retrieve information if it exists
safe_get <- function(x, ...) {
  result <- tryCatch({
    Reduce(function(x, name) if (!is.null(x) && name %in% names(x)) x[[name]] else NULL,
list(x, ...))
  }, error = function(e) NULL)
  result
}

# extract unique Domain_Category values
unique_categories <- unique(top_substances$Domain_Category)

# initialize a list to store results
result_list <- list()

# loop through each unique category
for (category in unique_categories) {
  # retrieve data using get_pug_rest for each category
  result_d <- get_pug_rest(
    identifier = category,
    namespace = "name",
    domain = "compound",
    operation = "classification",
    output = "JSON"
  )

  # check if result_d contains expected data structure
  if (!is.null(safe_get(result_d, "result", "Hierarchies", "Hierarchy"))) {
    # process the retrieved data
    hazard_info <- hazards_retriever(GHS_searcher(result_d), result_d)

    # store the results in the list
    result_list[[category]] <- hazard_info
  } else {
    # handle cases where data is missing
    result_list[[category]] <- "Data not available"
  }
}
```

```
## Request failed [404]. Retrying in 2.7 seconds...
```

```
## Request failed [404]. Retrying in 3.7 seconds...
```

```
## Request failed [404]. Retrying in 1.8 seconds...
```

```
## Request failed [404]. Retrying in 1 seconds...
## Request failed [404]. Retrying in 1 seconds...
```

```
## Request failed [404]. Retrying in 3.4 seconds...
```

```
## Request failed [404]. Retrying in 1 seconds...
```

```
## Request failed [404]. Retrying in 3.6 seconds...
```

```
# view result_list
print(result_list)
```

```
## $NITROGEN
##  [1] "H280: Contains gas under pressure; may explode if heated [Warning Gases under p
ressure]"
##  [2] "H200: Physical Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H281: Contains refrigerated gas; may cause cryogenic burns or injury [Warning G
ases under pressure]"
##  [5] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [6] "H300: Health Hazards"
##  [7] "H319: Causes serious eye irritation [Warning Serious eye damage/eye irritatio
n]"
##  [8] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [9] "H400: Environmental Hazards"
## [10] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $POTASH
## [1] "Data not available"
##
## $PHOSPHATE
## logical(0)
##
## $SULFUR
##  [1] "H228: Flammable solid [Danger Flammable solids]"
##  [2] "H200: Physical Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H315: Causes skin irritation [Warning Skin corrosion/irritation]"
##  [5] "H300: Health Hazards"
##  [6] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [7] "H319: Causes serious eye irritation [Warning Serious eye damage/eye irritatio
n]"
##  [8] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
##  [9] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
## [10] "H413: May cause long lasting harmful effects to aquatic life [Hazardous to the
aquatic environment, long-term hazard]"
## [11] "H400: Environmental Hazards"
##
## $CAPTAN
##  [1] "H303: May be harmful if swallowed [Warning Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H315: Causes skin irritation [Warning Skin corrosion/irritation]"
##  [5] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [6] "H318: Causes serious eye damage [Danger Serious eye damage/eye irritation]"
##  [7] "H330: Fatal if inhaled [Danger Acute toxicity, inhalation]"
##  [8] "H331: Toxic if inhaled [Danger Acute toxicity, inhalation]"
##  [9] "H340: May cause genetic defects [Danger Germ cell mutagenicity]"
## [10] "H351: Suspected of causing cancer [Warning Carcinogenicity]"
## [11] "H361: Suspected of damaging fertility or the unborn child [Warning Reproductive
toxicity]"
## [12] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
## [13] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
## [14] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
## [15] "H400: Environmental Hazards"
## [16] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $THIRAM
##  [1] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H302+H332: Harmful if swallowed or if inhaled [Warning Acute toxicity, oral; ac
ute toxicity, inhalation]"
##  [5] "H315: Causes skin irritation [Warning Skin corrosion/irritation]"
##  [6] "H316: Causes mild skin irritation [Warning Skin corrosion/irritation]"
##  [7] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [8] "H319: Causes serious eye irritation [Warning Serious eye damage/eye irritatio
n]"
##  [9] "H320: Causes eye irritation [Warning Serious eye damage/eye irritation]"
## [10] "H330: Fatal if inhaled [Danger Acute toxicity, inhalation]"
## [11] "H332: Harmful if inhaled [Warning Acute toxicity, inhalation]"
## [12] "H340: May cause genetic defects [Danger Germ cell mutagenicity]"
## [13] "H341: Suspected of causing genetic defects [Warning Germ cell mutagenicity]"
## [14] "H361: Suspected of damaging fertility or the unborn child [Warning Reproductive
toxicity]"
## [15] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
## [16] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
## [17] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
## [18] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
## [19] "H400: Environmental Hazards"
## [20] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $PENDIMETHALIN
##  [1] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [5] "H351: Suspected of causing cancer [Warning Carcinogenicity]"
##  [6] "H361: Suspected of damaging fertility or the unborn child [Warning Reproductive
toxicity]"
##  [7] "H361d: Suspected of damaging the unborn child [Warning Reproductive toxicity]"
##  [8] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
```

```
##  [9] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
## [10] "H400: Environmental Hazards"
## [11] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $OXYFLUORFEN
##  [1] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [2] "H400: Environmental Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##  [5] "Hazardous to the aquatic environment, acute hazard"
##  [6] "Environmental Hazards"
##  [7] "Hazard Classes"
##  [8] "Hazardous to the aquatic environment, long-term hazard"
##  [9] "<img src=\"https://pubchem.ncbi.nlm.nih.gov/images/ghs/GHS09.svg\" style=\"widt
h:40px;height:40px\"/> GHS09"
## [10] "Hazard Pictograms"
##
## $FLUMIOXAZIN
##  [1] "H360: May damage fertility or the unborn child [Danger Reproductive toxicity]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H361d: Suspected of damaging the unborn child [Warning Reproductive toxicity]"
##  [5] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [6] "H400: Environmental Hazards"
##  [7] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##  [8] "Hazardous to the aquatic environment, acute hazard"
##  [9] "Environmental Hazards"
## [10] "Hazard Classes"
## [11] "Hazardous to the aquatic environment, long-term hazard"
##
## $`NEEM OIL, CLAR. HYD.`
## [1] "Data not available"
##
## $MALATHION
##  [1] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [5] "H320: Causes eye irritation [Warning Serious eye damage/eye irritation]"
##  [6] "H331: Toxic if inhaled [Danger Acute toxicity, inhalation]"
##  [7] "H341: Suspected of causing genetic defects [Warning Germ cell mutagenicity]"
##  [8] "H350: May cause cancer [Danger Carcinogenicity]"
##  [9] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
## [10] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
```

```
## [11] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
## [12] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
## [13] "H400: Environmental Hazards"
## [14] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $BIFENAZATE
##  [1] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H319: Causes serious eye irritation [Warning Serious eye damage/eye irritatio
n]"
##  [5] "H320: Causes eye irritation [Warning Serious eye damage/eye irritation]"
##  [6] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
##  [7] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
##  [8] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [9] "H400: Environmental Hazards"
## [10] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
## [11] "Hazardous to the aquatic environment, acute hazard"
## [12] "Environmental Hazards"
## [13] "Hazard Classes"
## [14] "Hazardous to the aquatic environment, long-term hazard"
##
## $CHLOROPICRIN
##  [1] "H301: Toxic if swallowed [Danger Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
##  [5] "H314: Causes severe skin burns and eye damage [Danger Skin corrosion/irritatio
n]"
##  [6] "H315: Causes skin irritation [Warning Skin corrosion/irritation]"
##  [7] "H318: Causes serious eye damage [Danger Serious eye damage/eye irritation]"
##  [8] "H319: Causes serious eye irritation [Warning Serious eye damage/eye irritatio
n]"
##  [9] "H330: Fatal if inhaled [Danger Acute toxicity, inhalation]"
## [10] "H335: May cause respiratory irritation [Warning Specific target organ toxicity,
single exposure; Respiratory tract irritation]"
## [11] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
## [12] "H372: Causes damage to organs through prolonged or repeated exposure [Danger Sp
ecific target organ toxicity, repeated exposure]"
## [13] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
## [14] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
## [15] "H400: Environmental Hazards"
## [16] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $DICHLOROPROPENE
## [1] "Data not available"
##
## $`REYNOUTRIA SACHALINE`
## [1] "Data not available"
##
## $`METAM-POTASSIUM`
##  [1] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H312: Harmful in contact with skin [Warning Acute toxicity, dermal]"
##  [5] "H314: Causes severe skin burns and eye damage [Danger Skin corrosion/irritatio
n]"
##  [6] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [7] "H332: Harmful if inhaled [Warning Acute toxicity, inhalation]"
##  [8] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [9] "H400: Environmental Hazards"
## [10] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##
## $ACEQUINOCYL
##  [1] "H317: May cause an allergic skin reaction [Warning Sensitization, Skin]"
##  [2] "H300: Health Hazards"
##  [3] "Hazard Statement Codes"
##  [4] "H370: Causes damage to organs [Danger Specific target organ toxicity, single ex
posure]"
##  [5] "H373: May causes damage to organs through prolonged or repeated exposure [Warni
ng Specific target organ toxicity, repeated exposure]"
##  [6] "H400: Very toxic to aquatic life [Warning Hazardous to the aquatic environment,
acute hazard]"
##  [7] "H400: Environmental Hazards"
##  [8] "H410: Very toxic to aquatic life with long lasting effects [Warning Hazardous t
o the aquatic environment, long-term hazard]"
##  [9] "Hazardous to the aquatic environment, acute hazard"
## [10] "Environmental Hazards"
## [11] "Hazard Classes"
## [12] "Hazardous to the aquatic environment, long-term hazard"
##
## $FLUTRIAFOL
## [1] "H302: Harmful if swallowed [Warning Acute toxicity, oral]"
## [2] "H300: Health Hazards"
## [3] "Hazard Statement Codes"
## [4] "H312: Harmful in contact with skin [Warning Acute toxicity, dermal]"
## [5] "H332: Harmful if inhaled [Warning Acute toxicity, inhalation]"
## [6] "H411: Toxic to aquatic life with long lasting effects [Hazardous to the aquatic
environment, long-term hazard]"
## [7] "H400: Environmental Hazards"
## [8] "H412: Harmful to aquatic life with long lasting effects [Hazardous to the aquati
c environment, long-term hazard]"
```

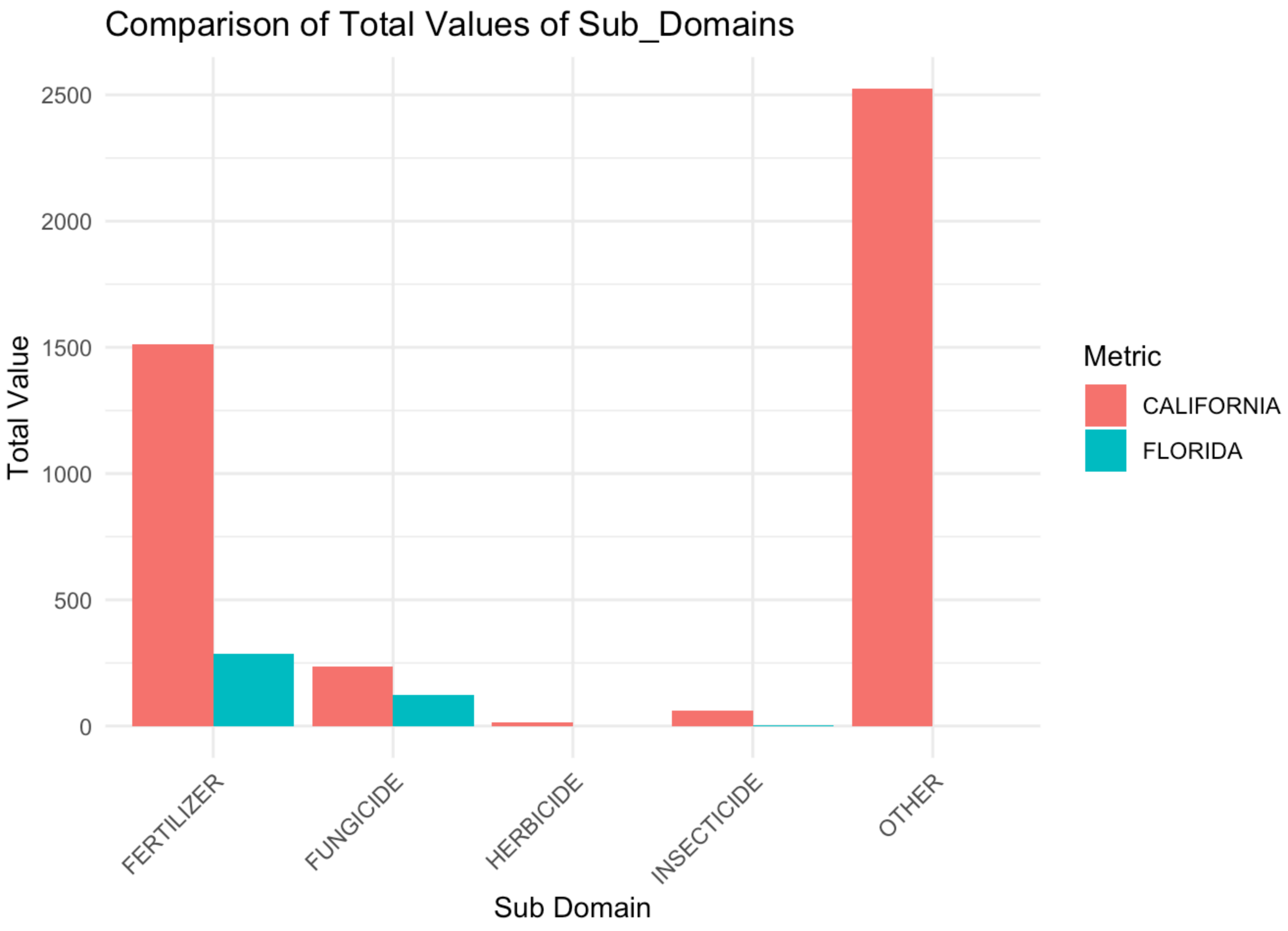
```
# The structure/composition/function/potential hazards of the top 3 most used substances
in each sub domain for each year in California were showed below to get more details abo
ut these substances.
```

```
### Analysis 6
### What is the difference in usage of each sub domain in unit of LB/ACRE/YEAR between C
alifornia and Florida?

# filter rows with "MEASURED IN LB / ACRE / YEAR"
strawberry_chemical2_item_mibay <- subset(strawberry_chemical2, Item == "MEASURED IN LB
/ ACRE / YEAR")

# ensure the "value" column is cleaned and converted to numeric
strawberry_chemical2_item_mibay$Value <- as.numeric(gsub(",", "", strawberry_chemical2_i
tem_mibay$Value))

# filter for California and Florida
strawberry_chemical2_item_mibay <- subset(strawberry_chemical2_item_mibay, State %in% c
("CALIFORNIA", "FLORIDA"))

# calculate the total "Value" for each "Sub_Domain" by state
sub_domain_comparison <- strawberry_chemical2_item_mibay %>%
  group_by(State, Sub_Domain) %>%
  summarize(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
# reshape the data for plotting
sub_domain_plot_data <- sub_domain_comparison %>%
  pivot_wider(names_from = State, values_from = Total_Value) %>%
  mutate(Difference_CA_FL = CALIFORNIA - FLORIDA)

# gather the data into long format for ggplot
plot_data <- sub_domain_plot_data %>%
  pivot_longer(cols = c("CALIFORNIA", "FLORIDA"),
               names_to = "Metric",
               values_to = "Total_Value")

# create the bar plot
ggplot(plot_data, aes(x = Sub_Domain, y = Total_Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Comparison of Total Values of Sub_Domains",
       x = "Sub Domain",
       y = "Total Value",
       fill = "Metric") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



```
# The bar plot compares the total values of various sub-domains (FERTILIZER, FUNGICIDE, HERBICIDE, INSECTICIDE, and OTHER) between California and Florida. California consistently shows higher totals across all sub-domains, with particularly large differences in "FERTILIZER" and "OTHER," where its values are significantly greater than Florida's. In sub-domains like "FUNGICIDE," "HERBICIDE," and "INSECTICIDE," the differences are less pronounced but still favor California. This disparity likely reflects differences in agricultural scale, practices, or crop requirements between the two states, with California exhibiting a much larger usage or application of the items measured.

# Unlike the analysis above, we used the unit 'MEASURED IN LB / ACRE / YEAR.' Below, I’ve shared my thoughts on the differences between using these two units in data analysis. The unit **"MEASURED IN LB"** represents the total quantity of a substance in pounds, without specifying how it is distributed across an area or over time, providing only a general measure of the total amount used or produced. In contrast, **"MEASURED IN LB / ACRE / YEAR"** normalizes the substance's application by area (per acre) and time (per year), offering a more specific and actionable metric. This normalized unit is particularly useful for comparing application rates across regions, assessing environmental impacts, or evaluating farming efficiency. While "LB" gives an overall quantity, "LB / ACRE / YEAR" provides context about application intensity, making it more relevant for detailed agricultural or environmental analysis.
```

```
### Analysis 7
### What is the relationship between the production in CWT of strawberries  and the usage of each sub domain each year in California?

# load the datasets
chemicals_data <- strawberry_chemical2_item_mibay
production_data <- strawberry_total

# clean the "Value" column in both datasets
chemicals_data$Value <- as.numeric(gsub(",", "", chemicals_data$Value))
production_data$Value <- as.numeric(gsub(",", "", production_data$Value))
```

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

```
# filter for California data
california_chemicals <- chemicals_data %>%
  filter(State == "CALIFORNIA")

california_production <- production_data %>%
  filter(State == "CALIFORNIA")

# aggregate total sub-domain usage by year
california_chemicals_aggregated <- california_chemicals %>%
  group_by(Year, Sub_Domain) %>%
  summarize(Total_Usage = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

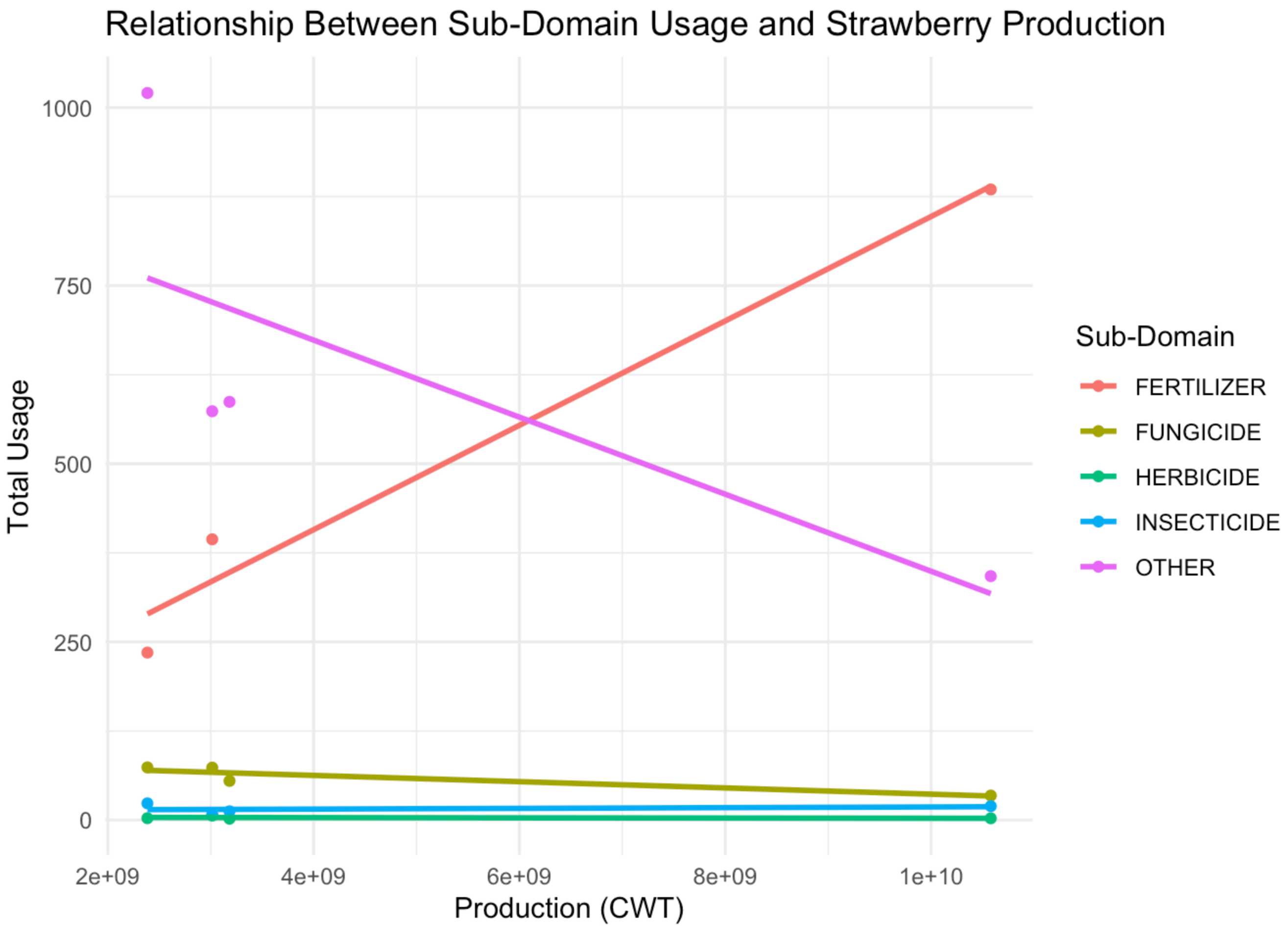
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
# aggregate production data by year
california_production_aggregated <- california_production %>%
  group_by(Year) %>%
  summarize(Production_CWT = sum(Value, na.rm = TRUE)) %>%
  ungroup()

# merge the datasets on the 'Year' column
merged_data <- california_chemicals_aggregated %>%
  left_join(california_production_aggregated, by = "Year")

# view the merged dataset
print(merged_data)
```

```
## # A tibble: 19 × 4
##     Year Sub_Domain  Total_Usage Production_CWT
##    <int> <chr>             <dbl>          <dbl>
##  1  2018 FERTILIZER       885      10579499738.
##  2  2018 FUNGICIDE         34.3    10579499738.
##  3  2018 HERBICIDE          2.40   10579499738.
##  4  2018 INSECTICIDE       19.4    10579499738.
##  5  2018 OTHER            342.     10579499738.
##  6  2019 FERTILIZER       235       2384818920
##  7  2019 FUNGICIDE         73.9     2384818920
##  8  2019 HERBICIDE          2.56    2384818920
##  9  2019 INSECTICIDE       23.4     2384818920
## 10  2019 OTHER           1020.      2384818920
## 11  2021 FUNGICIDE         54.9     3182560150
## 12  2021 HERBICIDE          1.84    3182560150
## 13  2021 INSECTICIDE       12.2     3182560150
## 14  2021 OTHER            587.      3182560150
## 15  2023 FERTILIZER       394       3014673496
## 16  2023 FUNGICIDE         73.5     3014673496
## 17  2023 HERBICIDE          6.21    3014673496
## 18  2023 INSECTICIDE        7.90    3014673496
## 19  2023 OTHER            574.      3014673496
```

```
# plot the relationship between usage in each sub-domain and production
ggplot(merged_data, aes(x = Production_CWT, y = Total_Usage, color = Sub_Domain)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Relationship Between Sub-Domain Usage and Strawberry Production",
    x = "Production (CWT)",
    y = "Total Usage",
    color = "Sub-Domain"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# The plot illustrates the relationship between the total usage of different chemical su
b-domains (e.g., FERTILIZER, FUNGICIDE, HERBICIDE, INSECTICIDE, OTHER) and strawberry pr
oduction (in CWT) in California. FERTILIZER usage shows a positive correlation with prod
uction, indicating that higher fertilizer use corresponds to increased strawberry yield
s. Conversely, the OTHER category demonstrates a negative correlation, where its usage d
ecreases as production rises. Sub-domains such as FUNGICIDE, HERBICIDE, and INSECTICIDE
appear to have weak or negligible correlations with production, suggesting their usage m
ight be relatively independent of yield levels. These trends highlight how specific chem
ical sub-domains may differently influence or relate to agricultural output.
```

Summarize:

We investigated the relationship between strawberry production and the usage of various chemical sub-domains in California. The dataset was cleaned and organized, focusing on California-specific data and differentiating between census and survey records. Columns with minimal variation were removed, and the data was aggregated by year and sub-domain for a more meaningful analysis. The goal was to explore how chemical usage patterns correlated with strawberry yields and to uncover trends across different sub-domains.

The analysis showed that fertilizer usage had a strong positive correlation with strawberry production. As production levels increased, the use of fertilizers rose consistently, suggesting its significant role in enhancing yields. This highlights the importance of fertilizers in supporting high-intensity strawberry farming in California. On the other hand, the OTHER sub-domain exhibited a negative correlation with production, indicating a potential reduction in its use as production becomes more efficient or as alternative practices are adopted.

For sub-domains like fungicides, herbicides, and insecticides, the correlations with production were either weak or negligible. This implies that these chemical categories may not directly influence overall production levels or may be used in a more consistent manner regardless of yield fluctuations. These findings suggest that their application might be more related to pest and disease management rather than yield optimization.

The visualizations provided additional insights into these relationships. Scatter plots with regression lines clearly demonstrated the trends for each sub-domain, allowing for easy identification of positive or negative correlations. Overall, this analysis underscores the critical role of fertilizers in driving strawberry production while revealing potential inefficiencies or evolving practices in other chemical sub-domains. These findings provide a foundation for further exploration of agricultural practices and their environmental impacts.