Project

EDA: Preparing Strawberry data for analysis

Due: March 21

As described in class, this document is a starter for the Midterm project.

Your assignment is to clean, organize, and explore the data. Turn in a report that describes how your work has set the stage for further analysis and model building.

The dataset contains strawberry farming data with details about conventional and organic cultivation. These data include information about chemicals used in strawberry farming, as well as sales, revenue and expense details.

While there is no "right answer" for this assignment, there are characteristics for the report that are essential. For example, sata visualization is critical. So is producing tabular presentations and document structure. Your target audience consists of analysts who may take the next steps with the data analysis and modeling.

Think of your report as a stage on which to showcase your ability to use R to work with data and produce professional reports. This is an opportunity to do some data storytelling.

Submit your report on or before March 21 using the Midterm portal on Blackboard.

Introduction: foundations

Before we begin to work with the strawberry data, let's talk about how we will approach the work.

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.

References In their handbook "An introduction to data cleaning with R" by Edwin de Jonge and Mark van der Loo, de Jonge and van der Loo go into detail about specific data cleaning isssues and how to handle them in R.

"Problems, Methods, and Challenges in Comprehensive Data Cleansing" by Heiko Müller and Johann-Christoph Freytag is a good companion to the de Jonge and van der Loo handbook, offering additional issues in their discussion.

Attitudes

Mechanistic descriptions of data cleaning methods are insufficient.

Data is the product (or by-product) of purposeful human activity Much of the data used in analysis accessed on local databases or online which may create the impression that the data have been carefully curated. Beware. Data are produced by people for a purpose, with a point-of-view, and at a time and location that may affect the data. The provenance and lineage of the data are meta data you should include when reporting analysis. Data collection is purposeful human activity with all of the risks and weaknesses that are part of any purposeful human activity.

Data is language Data has meaning. Data can be included in sentences related to the meaning of the data. Cleaning and organizing data should be informed by the meaning the data convey and how that meaning relates to the research you are doing do achieve this important result.

- Immerse yourself in the data. Put data into context.
- Visualize the data to find problems, confirm your understandings, and plan your data organization. People do a bad job of seeing meaningful patterns in data but a good job of seeing patterns of all kinds when data are rendered as plots. As you product and show visualizations, ask your self and those who view your presentations, "what do you see?" and "what do you wonder?"

Example: Strawberries

Public information

WHO says strawberries may not be so safe for you–2017March16

Pesticides + poison gases = cheap, year-round strawberries 2019March20

Multistate Outbreak of Hepatitis A Virus Infections Linked to Fresh Organic Strawberries-2022March5

Strawberry makes list of cancer-fighting foods-2023May31

What is the question?

- Where they are grown? By whom?
- Are they really loaded with carcinogenic poisons?
- Are they really good for your health? Bad for your health?
- Are organic strawberries carriers of deadly diseases?
- When I go to the market should I buy conventional or organic strawberries?

The data

The data set for this assignment has been selected from:

strawberries 2025march6

USDA NASS

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

Read the file

```
strawberry <- read_csv("strawb_mar6.csv",
                       col_names = TRUE,
                       show_col_types = FALSE)

source("my_functions.R")
```

Examine the data. How is it organized?

```
strawb <- strawberry |> drop_one_value_col()
```

```
# assume data is a tibble
# n_show is the number of rows to show



show_unique <- function(data, nrows=10 ){
  # make a tibble items to hold the data to show
  # browser()
  a <- nrows * dim(data)[2]  # number of cells in items
  items <- rep(" ", a) # items will coerce everything to char
  dim(items) <- c(nrows ,dim(data)[2]) # shape items
  items <- as_tibble(items)
  colnames(items) <- colnames(data)
  # browser()
  for(i in 1:dim(data)[2]){

    col_items <- unique(data[,i])
    # row_ex is the number of rows needed
    # to make the column length conformable with items
    row_ex <- nrows - dim(col_items)[1]
    if(row_ex >= 0){
      ex_rows <- tibble(rep(" ",row_ex))
      colnames(ex_rows) <- colnames(col_items)
      col_add <- rbind2(col_items, ex_rows)

    } else if(row_ex < 0){
      col_add <- col_items[1:10,]

    }

    items[,i] <- col_add

  }

  return(items)
}

## test <- show_unique(strawb, 10)
```

```
#|label: split strawb into census and survey pieces

strw_census <- strawb |> filter(Program == "CENSUS")

strw_survey <- strawb |> filter(Program == "SURVEY")

nrow(strawb) == (nrow(strw_census) + nrow(strw_survey))
```

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

```
s_census <- strw_census |> drop_one_value_col(prt_val = TRUE)
```

```
## [1] "Looking for single value columns in data frame:  strw_census"
## [1] "Columns dropped:"
##     Program      Period Week Ending
##    "CENSUS"      "YEAR"          NA
```

```
s_survey <- strw_survey |> drop_one_value_col(prt_val = TRUE)
```

```
## [1] "Looking for single value columns in data frame:  strw_survey"
## [1] "Columns dropped:"
##        Program      Commodity         CV (%)
##       "SURVEY" "STRAWBERRIES"             NA
```

```
unique_sur <- s_survey |> show_unique(nrows = 10)

unique_cen <- s_census |> show_unique(nrows = 10)


strw_census <- s_census |> select(-`State ANSI`)

strw_survey <- s_survey |> select(-`State ANSI`, -`Week Ending`, -Period)

rm(s_census, s_survey, strawberry, strawb, items)
```

```
commod <- strw_census$Commodity |> unique()

#### split Data Item

strw_census <- strw_census |>
  separate_wider_delim(  cols = Commodity,
                         delim = ",",
                         names = c("INCOME",
                                   "NET CASH FARM",
                                   "STRAW"
                                              ),
                         too_many = "error",
                         names_sep = " ",
                         too_few = "align_start"
  )
```

```r
inc <- strw_census$Fruit |> unique()

strw_census <- strw_census |>
  separate_wider_delim(  cols = Fruit,
                         delim = ",",
                         names = c("INCOME",
                                   "STRAWB"
                                              ),
                         too_many = "error",
                         too_few = "align_start"
  )
```

```r
straw_cen_f <- strw_census |> filter(State == "FLORIDA")

straw_sur_f <- strw_survey |> filter(State == "FLORIDA")
straw_cen_c <- strw_census |> filter(State == "CALIFORNIA")
straw_sur_c <- strw_survey |> filter(State == "CALIFORNIA")

straw_sur_c<-straw_sur_c %>% mutate(Value=gsub(',',"",Value))
straw_sur_f<-straw_sur_f %>% mutate(Value=gsub(',',"",Value))
straw_cen_c<-straw_cen_c %>% mutate(Value=gsub(',',"",Value))
straw_cen_f<-straw_cen_f %>% mutate(Value=gsub(',',"",Value))
```

```r
analyze<-function(data,data_title){
  data %>%
    distinct(Domain) %>%
    pull()

  selected_chems <- data %>%
    filter(Domain %in% c("CHEMICAL, FUNGICIDE", "CHEMICAL, INSECTICIDE", "CHEMICAL, HERBICIDE"))

  selected_chems <- selected_chems %>%
    mutate(Value = ifelse(Value == "(D)"|Value == "(NA)"|is.na(Value), NA, Value))%>%
    mutate(Value = as.numeric(as.character(Value)))

  chem_summary <- selected_chems %>%
    group_by(Domain) %>%
    summarise(
      Avg_Application_Rate = mean(Value, na.rm = TRUE),
      Total_Usage = sum(Value, na.rm = TRUE)
    ) %>%
    arrange(Domain)


  avg_rate_plot<-ggplot(chem_summary, aes(x = Domain, y = Avg_Application_Rate, fill = Domain)) +
    geom_col(width = 0.6) +
    geom_text(aes(label = Avg_Application_Rate), vjust = -0.5, size = 4) +
    labs(
      title = paste(data_title, "Avg Comparison"),
      x = "Domain",
      y = "Avg_Application_Rate"
    ) +
```

```
    scale_fill_brewer(palette = "Pastel1") +
    theme_minimal() +
    theme(
      legend.position = "none",
      axis.text.x = element_text(angle = 0, hjust = 0.5)
    )

  total_usage_plot<-ggplot(chem_summary, aes(x = Domain, y = Total_Usage, fill = Domain)) +
    geom_col(width = 0.6) +
    geom_text(aes(label = scales::comma(Total_Usage)), vjust = -0.5, size = 4) + # 千位分隔格式
    labs(
      title = paste(data_title,"Total Comparison"),
      x = "Domain",
      y = "Total_Usage"
    ) +
    scale_fill_brewer(palette = "Set2") +
    theme_minimal() +
    theme(
      legend.position = "none",
      panel.grid.major.x = element_blank()
    )



  multidim_plot<-ggplot(chem_summary, aes(x = Domain)) +
    geom_point(aes(y = "Usage", size = Total_Usage, color = Domain),
               alpha = 0.8, show.legend = FALSE) +
    geom_point(aes(y = "Rate", size = Avg_Application_Rate, color = Domain),
               alpha = 0.8, show.legend = FALSE) +
    scale_size_continuous(range = c(10, 30)) +
    scale_color_manual(values = c("#1B9E77", "#D95F02", "#7570B3")) +
    labs(
      title = paste(data_title,"Multidimensional Comparison"),
      x = NULL,
      y = NULL,
      caption = "Bubble size represents magnitude"
    ) +
    theme_linedraw() +
    theme(axis.text.y = element_text(face = "bold"))

  list(
      chem_summary = chem_summary,
      avg_rate_plot = avg_rate_plot,
      total_usage_plot = total_usage_plot,
      multidim_plot = multidim_plot
    )
}

result_ca <- analyze(straw_sur_c, "CALIFORNIA")
result_fl <- analyze(straw_sur_f, "FLORIDA")
```

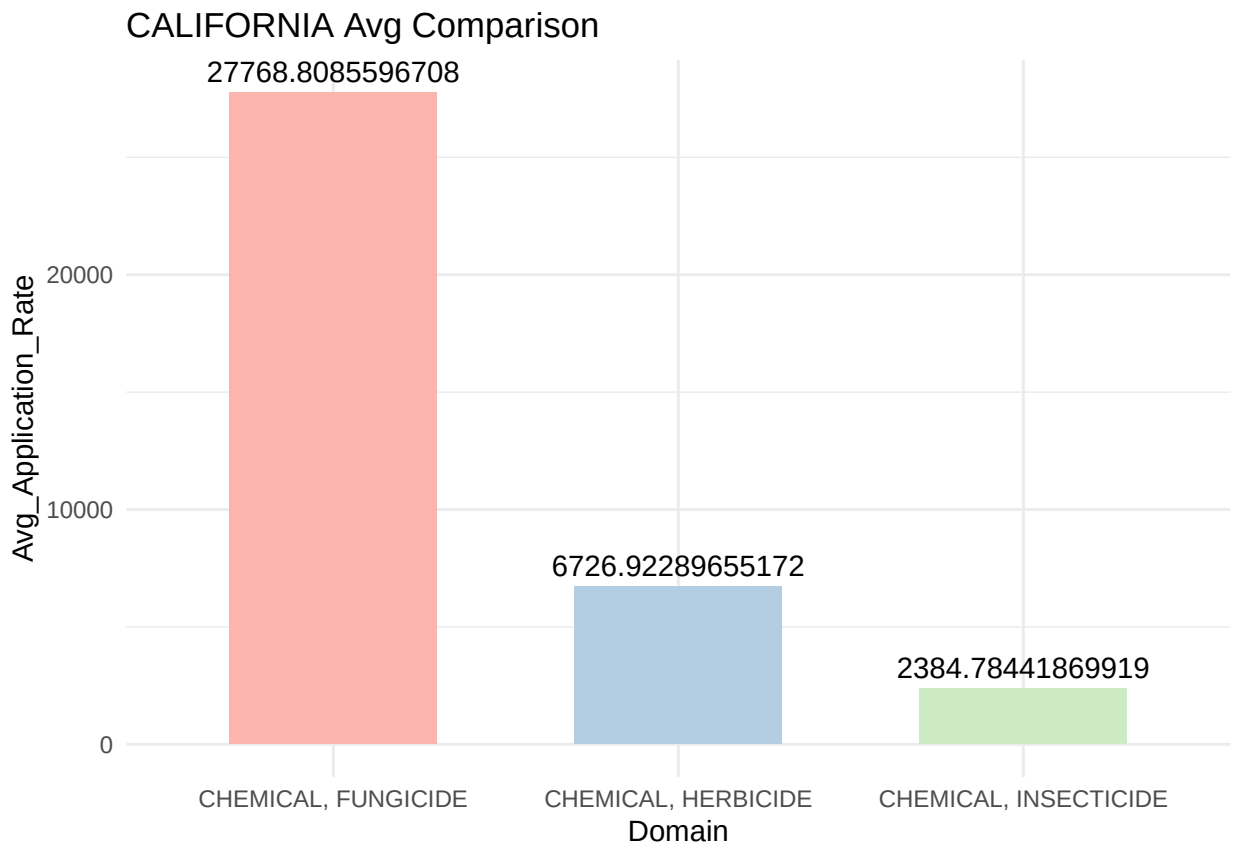
```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Value = as.numeric(as.character(Value))`.
```

```
## Caused by warning:
## ! 强制改变过程中产生了NA
```

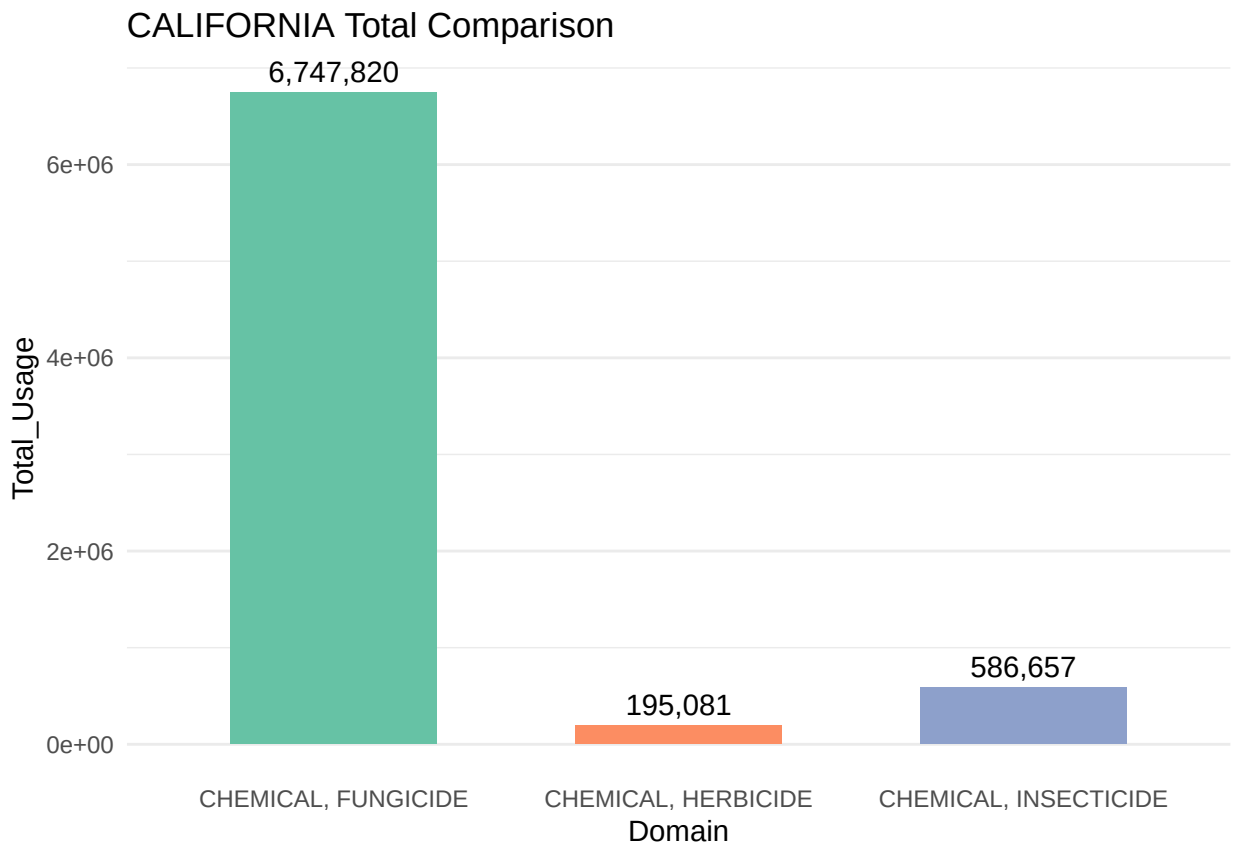
```
print(result_ca$chem_summary)
```

```
## # A tibble: 3 x 3
##   Domain                Avg_Application_Rate Total_Usage
##   <chr>                                <dbl>       <dbl>
## 1 CHEMICAL, FUNGICIDE                 27769.    6747820.
## 2 CHEMICAL, HERBICIDE                  6727.     195081.
## 3 CHEMICAL, INSECTICIDE                2385.     586657.
```

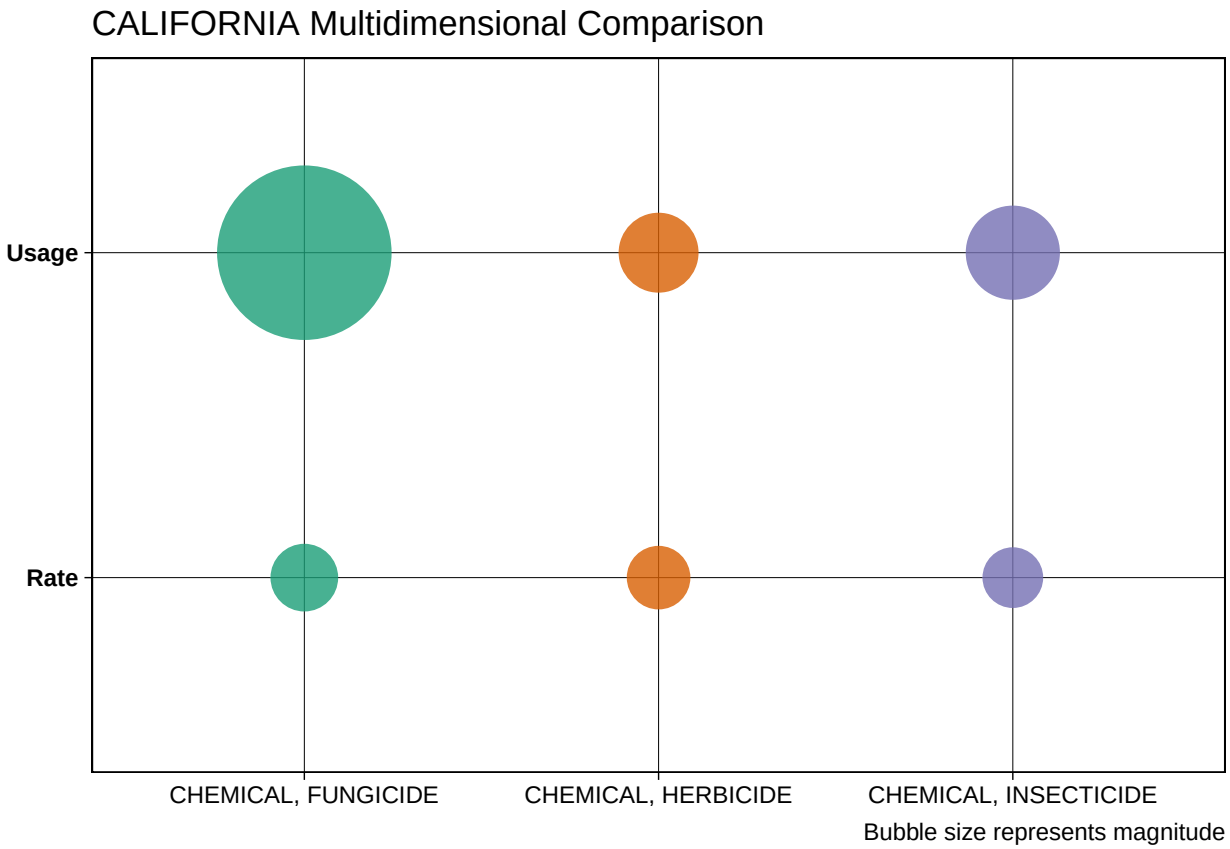
```
print(result_ca$avg_rate_plot)
```




```
print(result_ca$total_usage_plot)
```



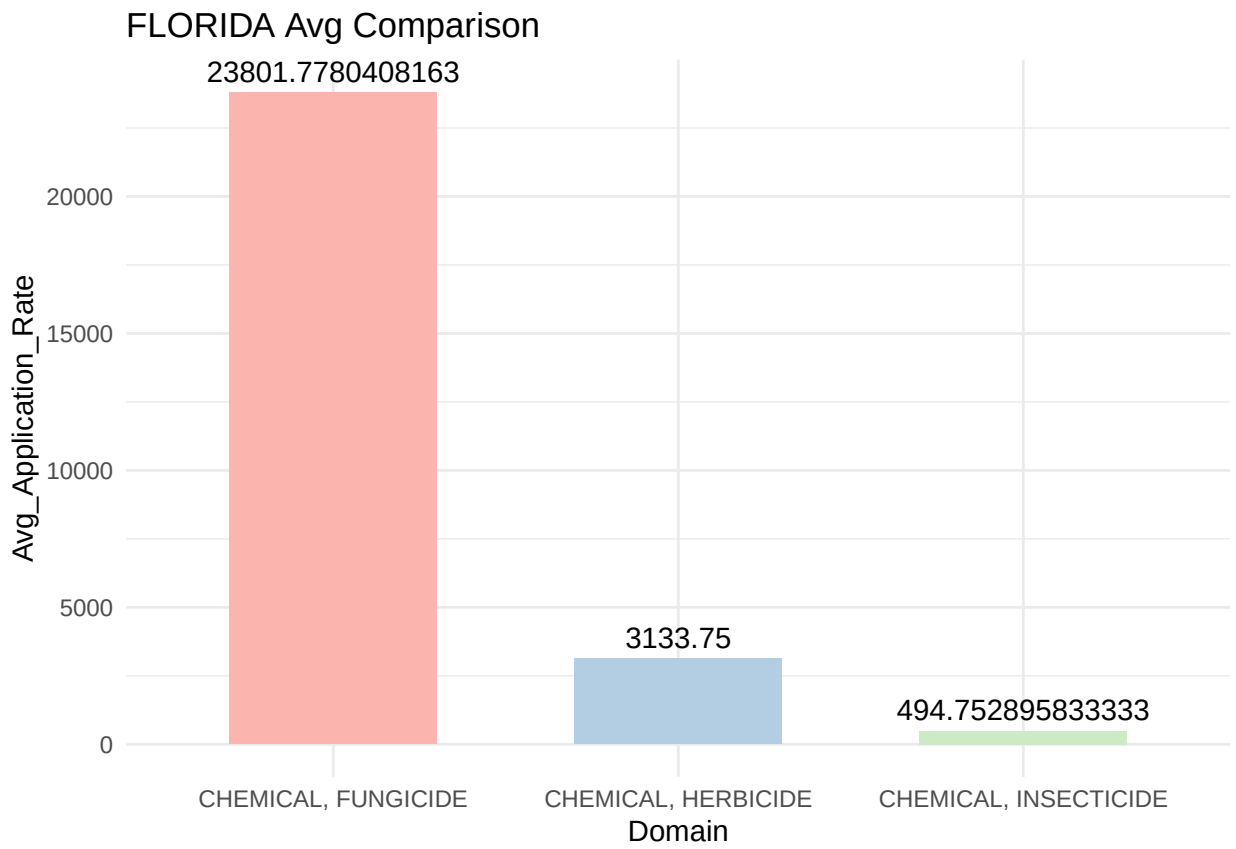

```
print(result_ca$multidim_plot)
```



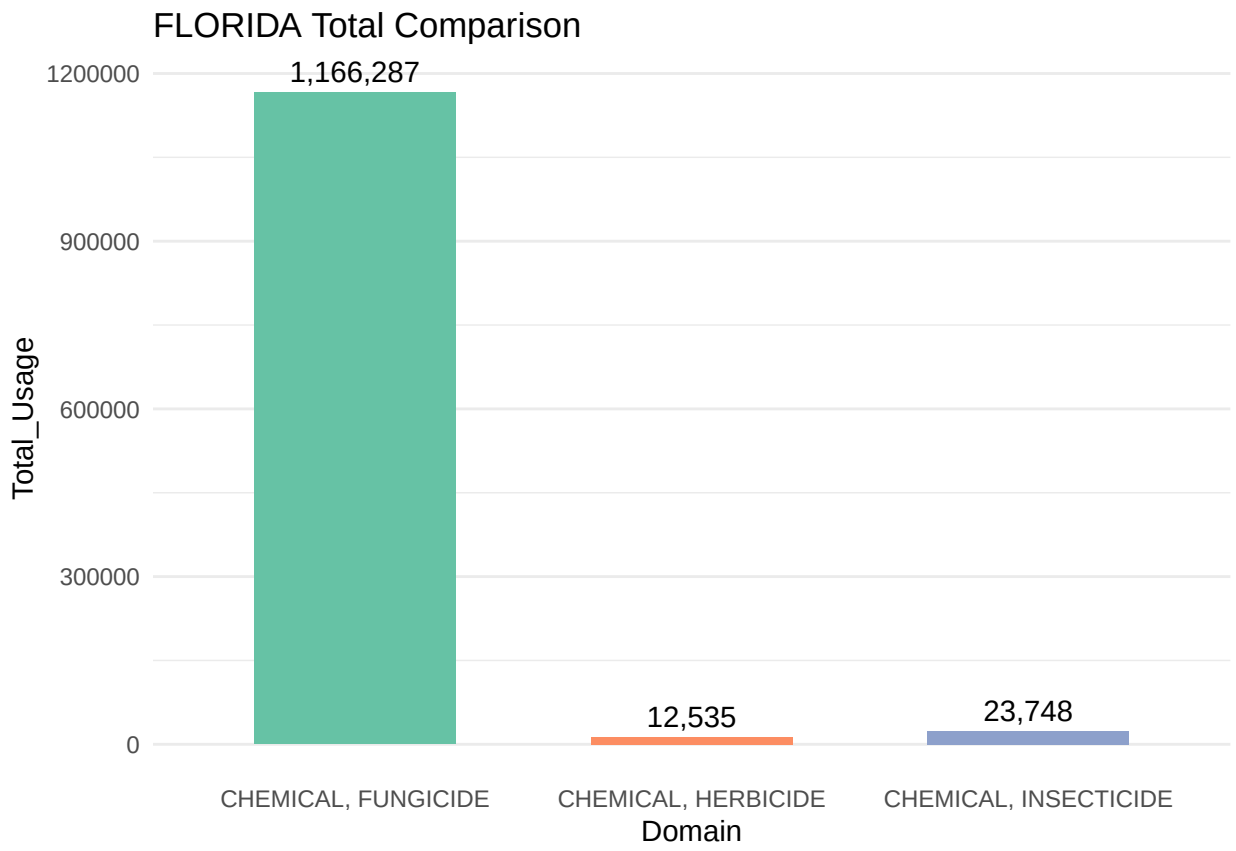

```
print(result_fl$chem_summary)
```

```
## # A tibble: 3 x 3
##   Domain                Avg_Application_Rate Total_Usage
##   <chr>                                <dbl>       <dbl>
## 1 CHEMICAL, FUNGICIDE                 23802.    1166287.
## 2 CHEMICAL, HERBICIDE                  3134.      12535
## 3 CHEMICAL, INSECTICIDE                 495.      23748.
```

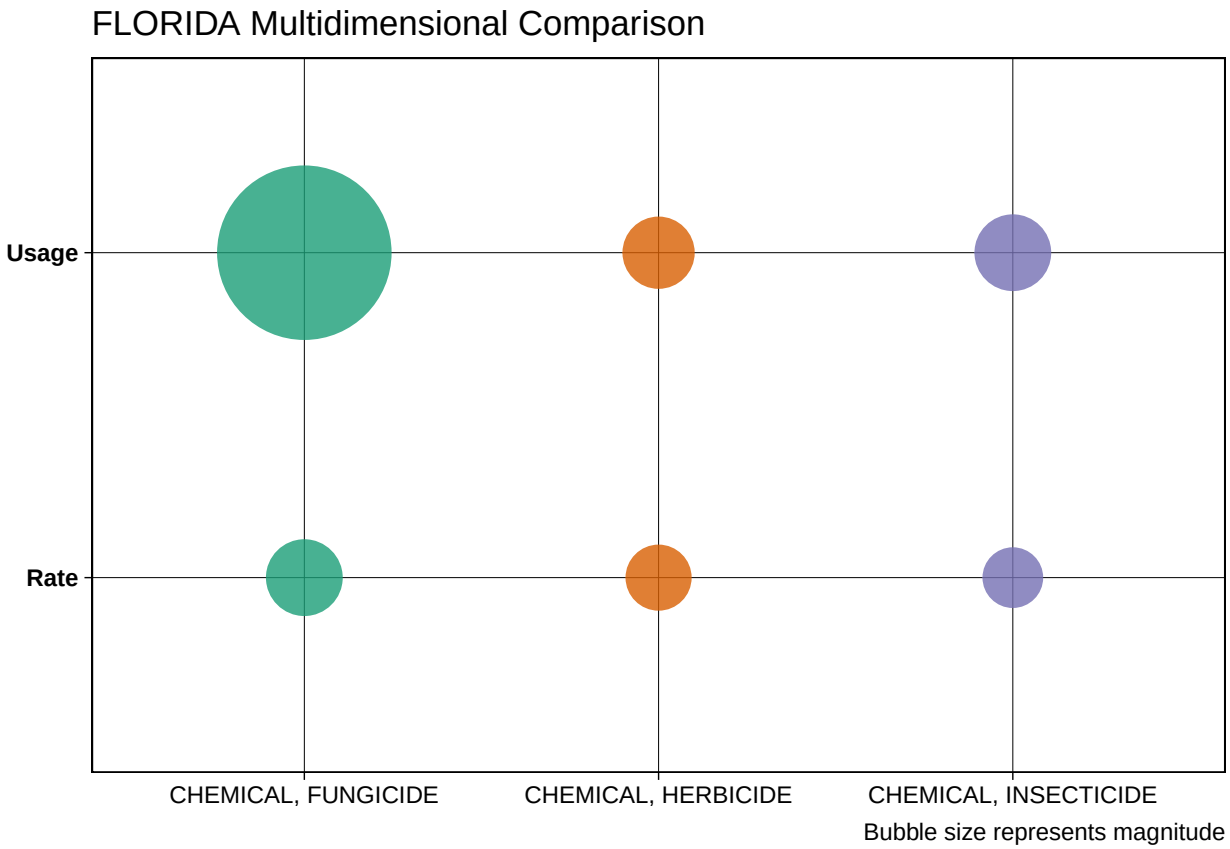
```
print(result_fl$avg_rate_plot)
```




```
print(result_fl$total_usage_plot)
```

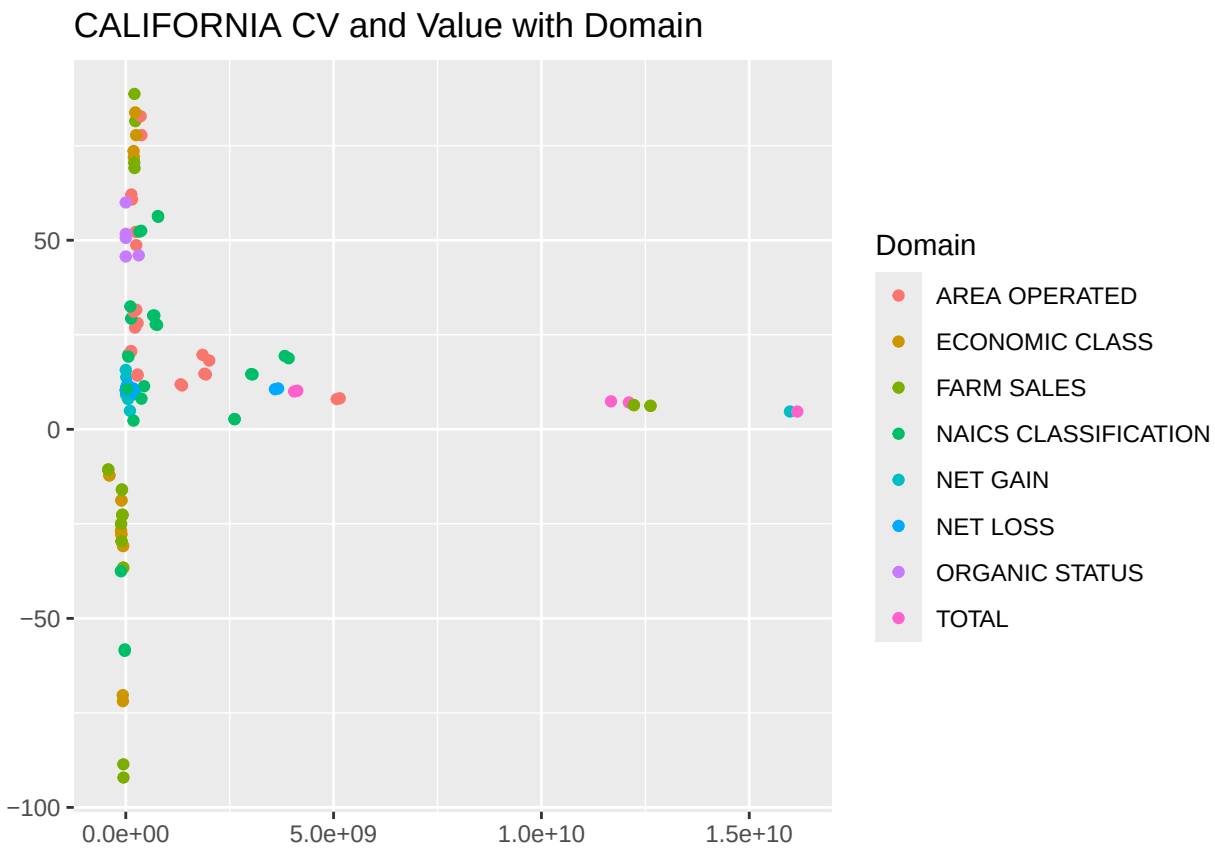



```
print(result_fl$multidim_plot)
```

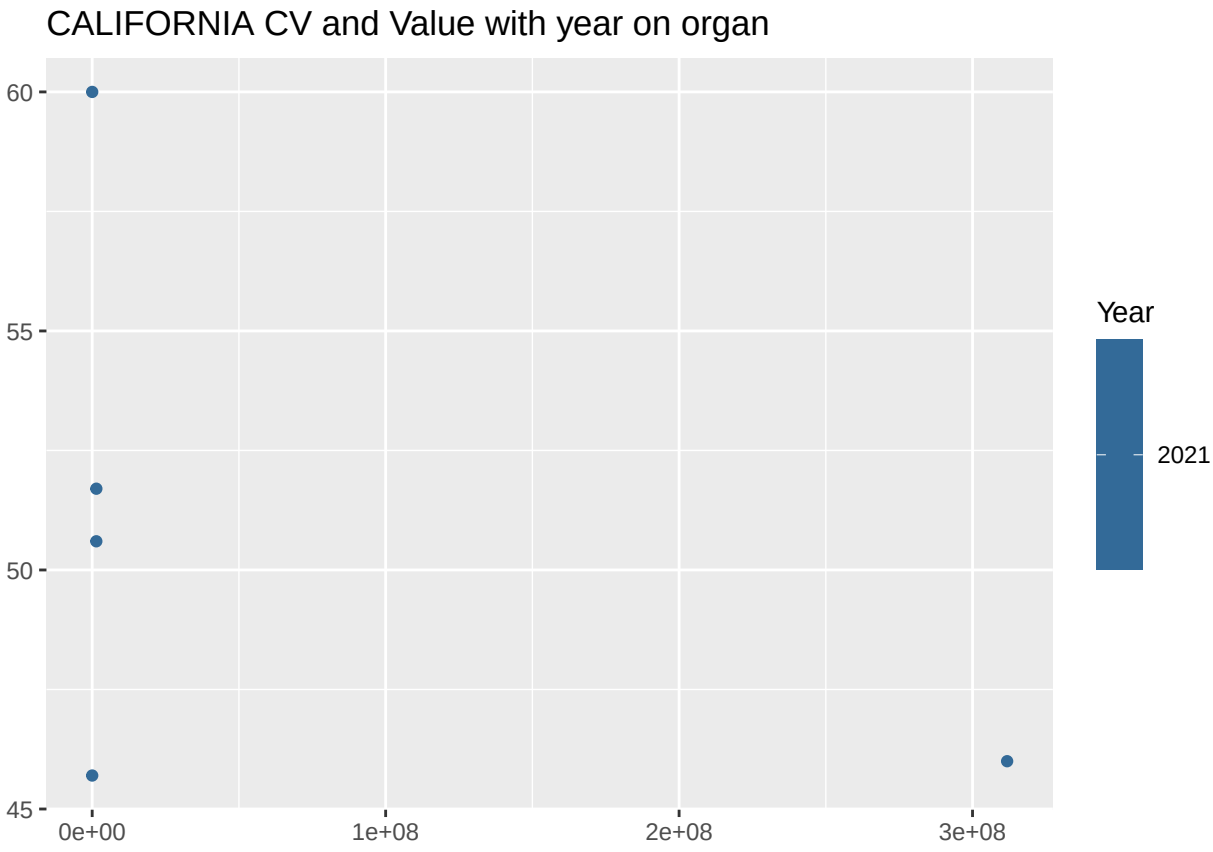



```
analyze2<-function(data,data_title){
  # 1. 处理 value 和 CV)
  strawb_columns <- data %>% select(Year,Domain,Value,`CV (%)`)

  strawb_columns <- strawb_columns %>%
    mutate(Value = ifelse(Value == "(D)"|Value == "(NA)"|is.na(Value), NA, Value))%>%
    mutate(Value = as.numeric(as.character(Value)))

  strawb_columns <- strawb_columns %>%
    mutate(`CV (%)` = ifelse(`CV (%)` == "(D)"|`CV (%)` == "(H)"|`CV (%)` == "(L)"|`CV (%)` == "(NA)"|is
    mutate(`CV (%)` = as.numeric(as.character(`CV (%)`)))

  strawb_columns <- strawb_columns %>% drop_na(Value,`CV (%)`)

  columns1<- strawb_columns %>% group_by(Year,Domain)  %>% summarise(
    Mean_Value = mean(Value,na.rm=TRUE),
    Mean_CV = mean(`CV (%)`,na.rm = TRUE),
    .groups = "keep"
  )

  all_plot <- strawb_columns %>% ggplot(aes(x=Value,y=`CV (%)`,color=Domain))+geom_point()+
    labs(
      title = paste(data_title,"CV and Value with Domain"),
      x = NULL,
      y = NULL,
    )
```

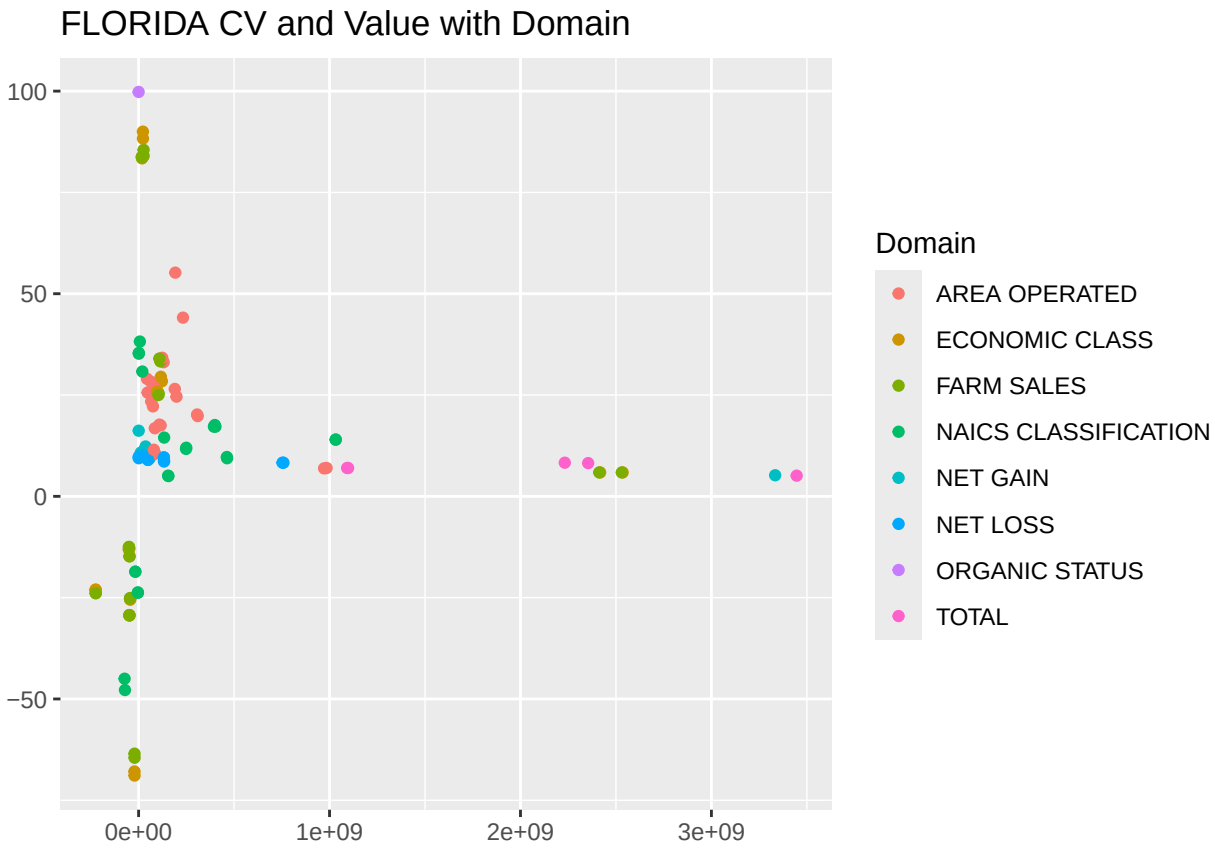
```r
  organ_columns <-  strawb_columns %>% filter(Domain == "ORGANIC STATUS")

  organ_plot <- organ_columns %>% ggplot(aes(x=Value,y=`CV (%)`,color=Year))+geom_point()+
    labs(
      title = paste(data_title,"CV and Value with year on organ"),
      x = NULL,
      y = NULL,
    )

  # line_plot <- ggplot(columns1, aes(x = Year)) +
  #   geom_line(aes(y = Mean_CV, color = Domain)) +
  #     geom_line(aes(y = Mean_Value, color = Domain)) +
  #     labs(
  #       title = "CV and Value Over Time",
  #       x = "Year",
  #       y = "Values",
  #       color = "Domain"
  #     ) +
  #     scale_color_brewer(palette = "Set1") +  # 使用预设的颜色方案
  #     theme_minimal()

  list(
      all_plot = all_plot,
      organ_plot = organ_plot
      # line_plot = line_plot
    )
}

result_ca<-analyze2(straw_cen_c,"CALIFORNIA")
result_fl<-analyze2(straw_cen_f,"FLORIDA")

print(result_ca$all_plot)
```



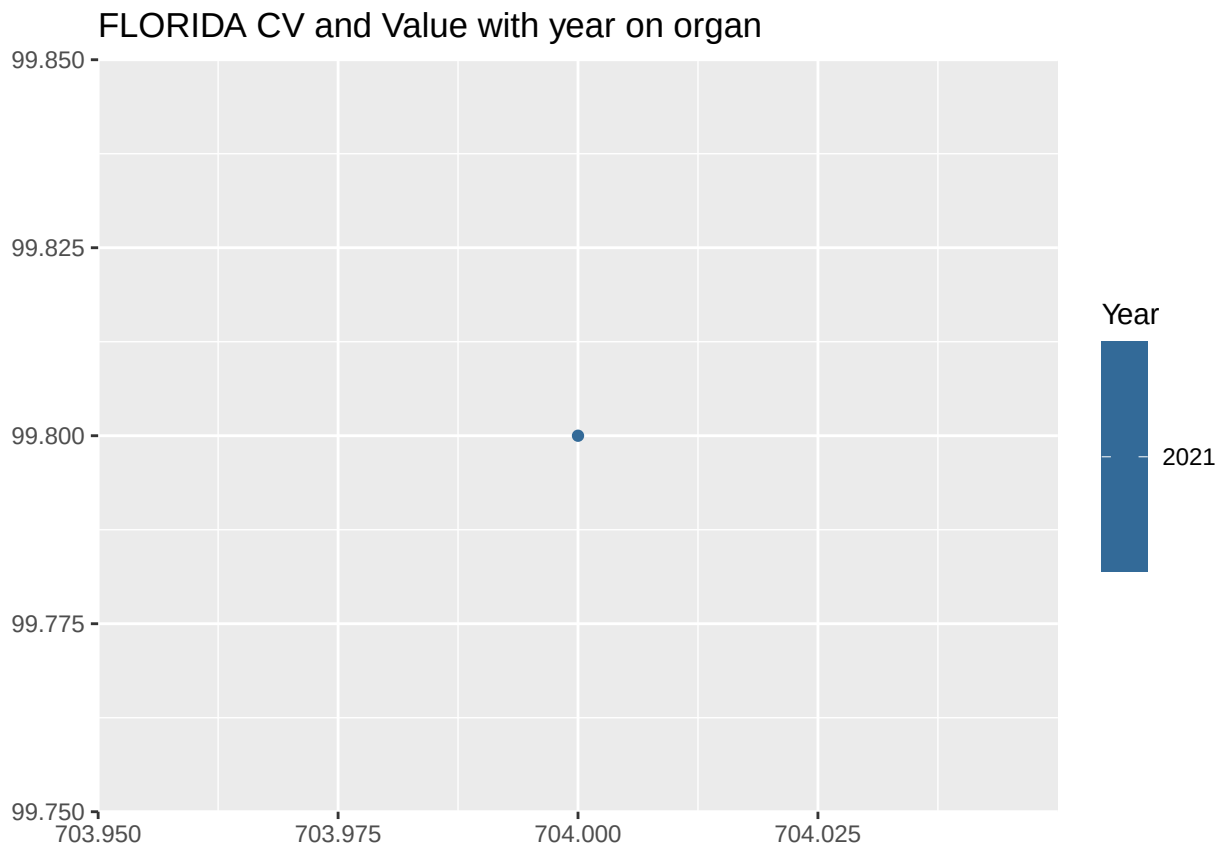

```
print(result_ca$organ_plot)
```




```
print(result_fl$all_plot)
```

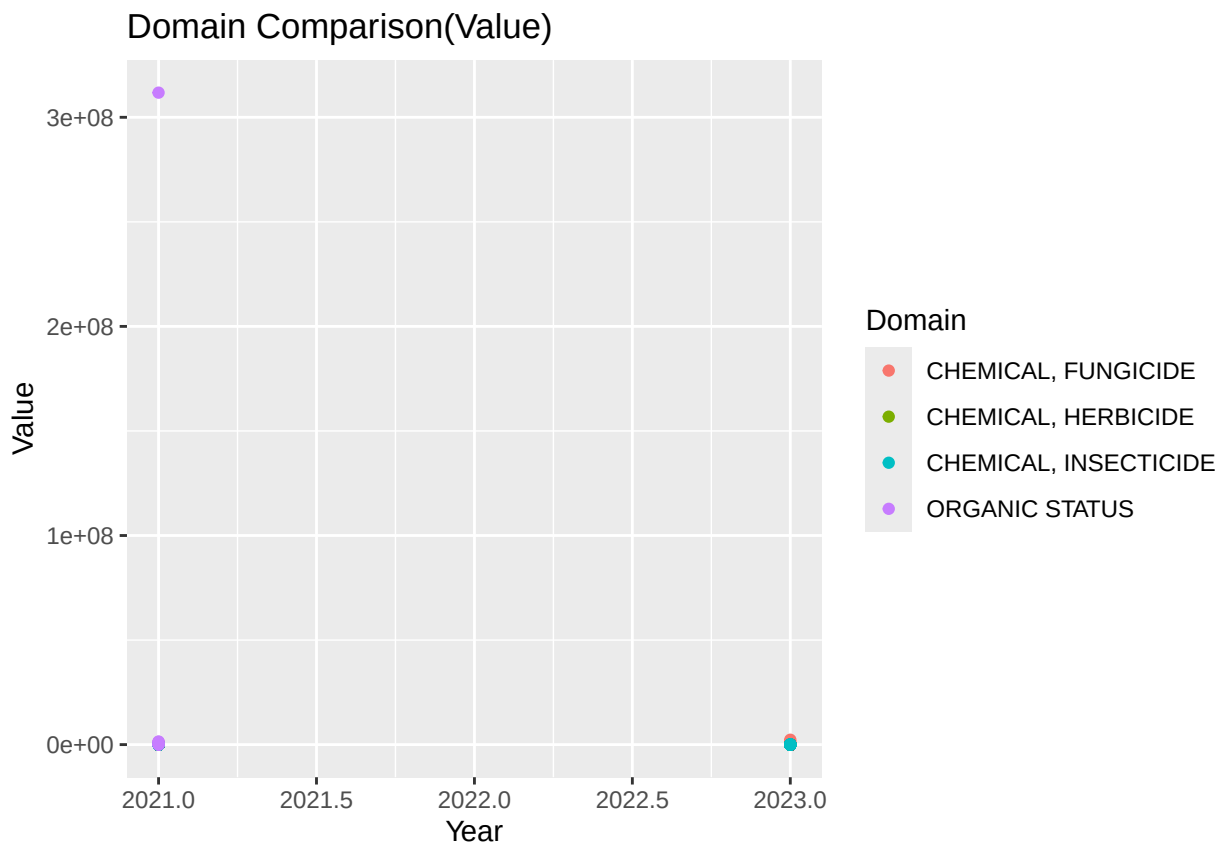



```
print(result_fl$organ_plot)
```



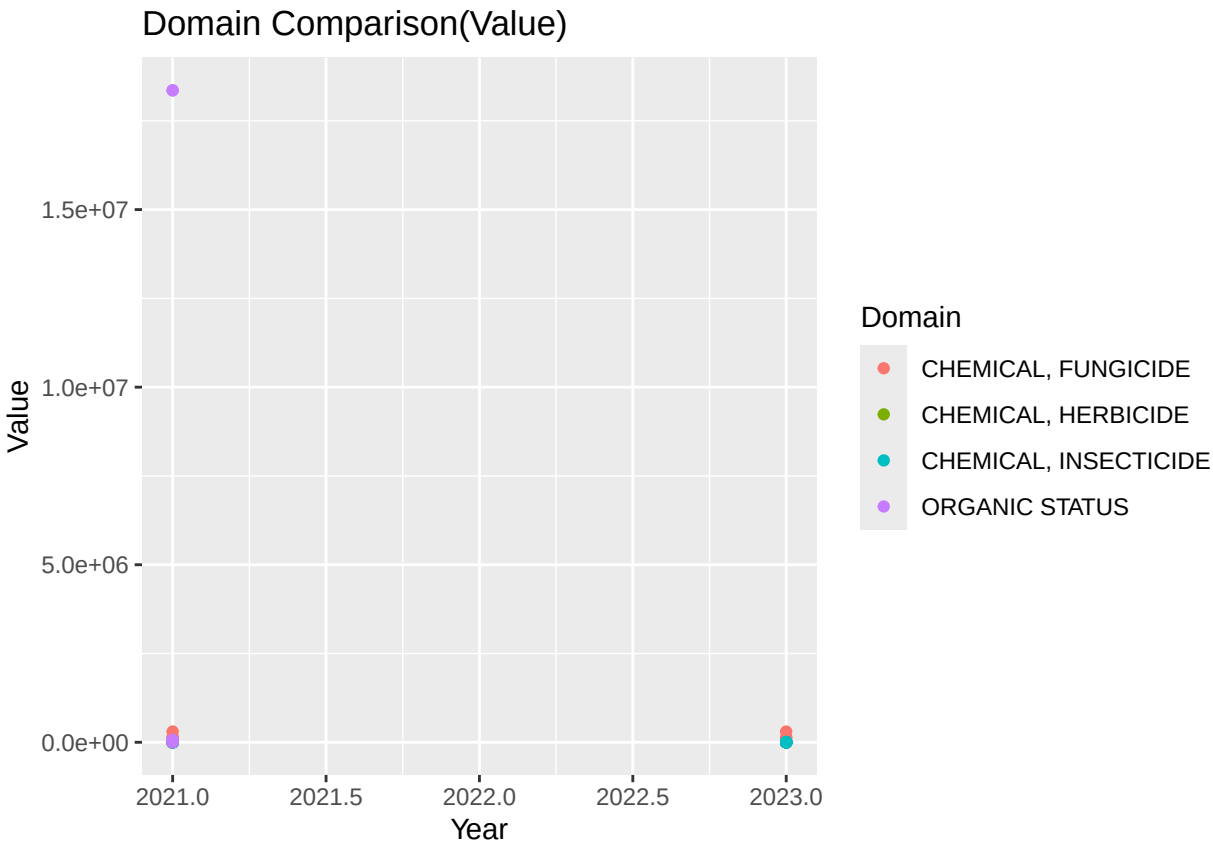

```
# 对比
analyze3<-function(data1,data2){
  strawb_columns <- data1 %>% select(Year,Domain,Value)

  strawb_columns <- strawb_columns %>% drop_na(Value)

  strawb_columns <- strawb_columns %>%
    mutate(Value = ifelse(Value == "(D)"|Value == "(NA)"|Value == "(H)"|is.na(Value), NA, Value))%>%
    mutate(Value = as.numeric(as.character(Value)))

  strawb_columns <- strawb_columns %>% drop_na(Value)

  organ_value <-  strawb_columns %>% filter(Domain == "ORGANIC STATUS")


  selected_chems <- data2 %>%
    filter(Domain %in% c("CHEMICAL, FUNGICIDE", "CHEMICAL, INSECTICIDE", "CHEMICAL, HERBICIDE"))

  selected_chems <- selected_chems %>%
    mutate(Value = ifelse(Value == "(D)"|Value == "(NA)"|Value == "(H)"|is.na(Value), NA, Value))%>%
    mutate(Value = as.numeric(as.character(Value)))

  selected_chems <- selected_chems %>% drop_na(Value)

  selected_chems <- selected_chems %>% select(Year,Domain,Value)
```

```
  combined_data <- bind_rows(selected_chems, organ_value)

  combined_data <- combined_data %>% drop_na(Value)

  combined_data %>% ggplot(aes(x=Year,y=Value,color=Domain))+geom_point()+
    labs(
      title = "Domain Comparison(Value)",
      x = "Year",
      y = "Value"
    )
}

analyze3(straw_cen_c,straw_sur_c)
```




```
analyze3(straw_cen_f,straw_sur_f)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Value = as.numeric(as.character(Value))`.
## Caused by warning:
## ! 强制改变过程中产生了NA
```


Domain Comparison(Value)
Value
1.5e+07
1.0e+07
5.0e+06
0.0e+00
2021.0
2021.5
2022.0
2022.5
2023.0
Year
Domain
CHEMICAL, FUNGICIDE
CHEMICAL, HERBICIDE
CHEMICAL, INSECTICIDE
ORGANIC STATUS