

Strawberry Data EDA

Zecheng Li

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
datacen <- read.csv("straw_cen_cleaned2.csv")
datasur <- read.csv("straw_sur_cleaned2.csv")
str(datasur)
```

```
## 'data.frame': 1432 obs. of 16 variables:
## $ Program      : chr "SURVEY" "SURVEY" "SURVEY" "SURVEY" ...
## $ Year         : int 2024 2024 2023 2023 2023 2023 2023 2023 2023 2023 ...
## $ Period       : chr "YEAR" "YEAR" "MARKETING YEAR" "MARKETING YEAR" ...
## $ Geo.Level    : chr "NATIONAL" "NATIONAL" "NATIONAL" "NATIONAL" ...
## $ State        : chr "US TOTAL" "US TOTAL" "US TOTAL" "US TOTAL" ...
## $ State.ANSI   : int -1 -1 -1 -1 -1 6 12 -1 -1 -1 ...
## $ Commodity    : chr "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" ...
## $ Market_Type  : chr "FRESH MARKET" "PROCESSING" "OTHER" "FRESH MARKET" ...
## $ Measure_Operation: chr "PRICE RECEIVED, ADJUSTED BASE" "PRICE RECEIVED, ADJUSTED BASE" "PRICE RE
## $ Unit_of_Measure : chr "$ / CWT" "$ / TON" "$ / CWT" "$ / CWT" ...
## $ Domain       : chr "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
## $ Chemical_Use  : chr "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" ...
## $ Chemical_Name : chr "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" ...
## $ Chemical_Code : logi NA NA NA NA NA NA ...
## $ Value        : num 10.9 4.04 123 142 43.8 121 147 142 43.8 485 ...
## $ CV....       : logi NA NA NA NA NA NA ...
```

```
unique1<- unique(datasur$Chemical_Name)
unique2<- unique(datasur$Chemical_Code)
ca_chemical <- subset(datasur, State != "California")
ca_chemical1 <- subset(ca_chemical, !(Chemical_Name %in% c("NOT SPECIFIED", "TOTAL")))
head(ca_chemical1)
```

```
##   Program Year Period Geo.Level      State State.ANSI      Commodity Market_Type
## 19 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
## 20 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
## 21 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
## 22 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
## 23 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
## 24 SURVEY 2023  YEAR      STATE CALIFORNIA          6 STRAWBERRIES  BEARING
##   Measure_Operation      Unit_of_Measure      Domain
## 19 APPLICATIONS          LB CHEMICAL, INSECTICIDE
## 20 APPLICATIONS LB / ACRE / APPLICATION, AVG  CHEMICAL, FUNGICIDE
```

```
## 21      APPLICATIONS LB / ACRE / APPLICATION, AVG    CHEMICAL, FUNGICIDE
## 22      APPLICATIONS LB / ACRE / APPLICATION, AVG    CHEMICAL, FUNGICIDE
## 23      APPLICATIONS LB / ACRE / APPLICATION, AVG    CHEMICAL, FUNGICIDE
## 24      APPLICATIONS LB / ACRE / APPLICATION, AVG    CHEMICAL, FUNGICIDE
##      Chemical_Use      Chemical_Name Chemical_Code  Value CV...
## 19  INSECTICIDE      (ABAMECTIN          NA 300.000    NA
## 20   FUNGICIDE      (AZOXYSTROBIN        NA  0.234    NA
## 21   FUNGICIDE (BORAX DECAHYDRATE        NA  0.042    NA
## 22   FUNGICIDE      (BOSCALID          NA  0.354    NA
## 23   FUNGICIDE      (CAPTAN             NA  1.693    NA
## 24   FUNGICIDE      (CYPRODINIL        NA  0.316    NA
```

```
ca_chemical2 <- ca_chemical1[ca_chemical1$Year %in% 2018:2023, ]
```

```
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
```

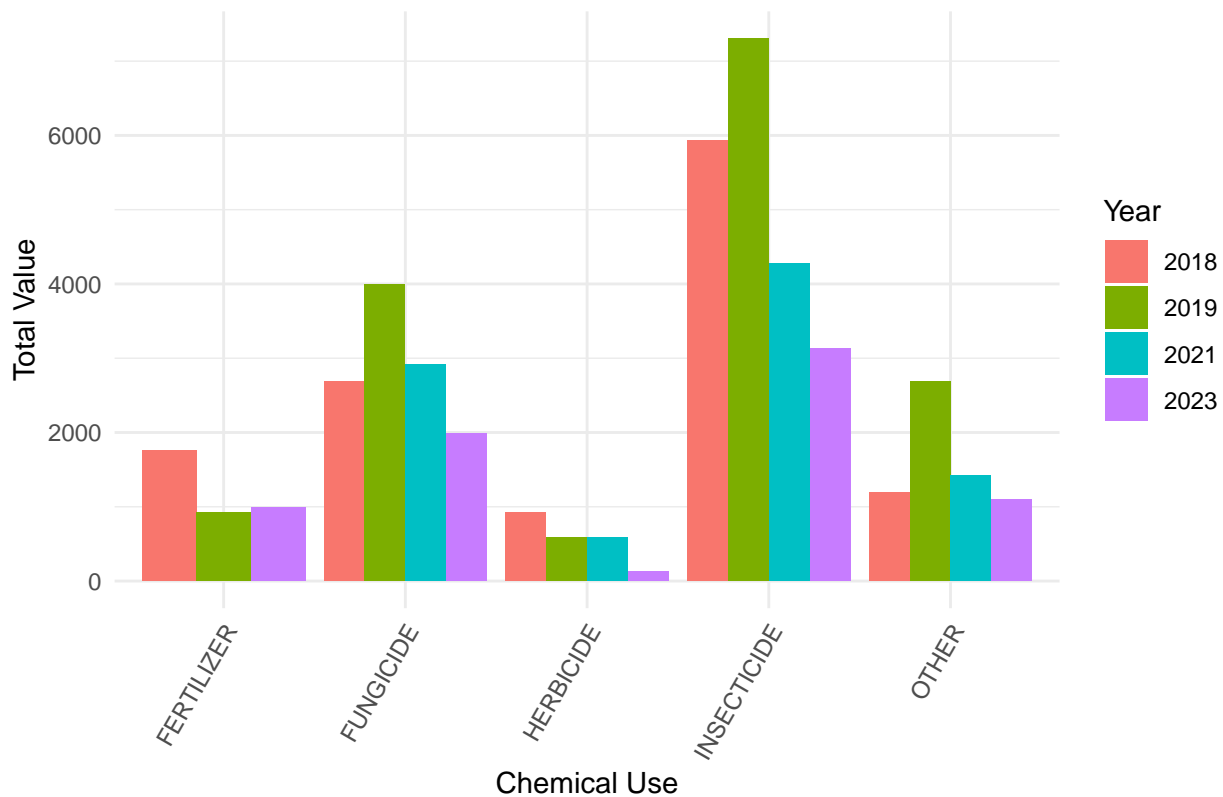
```
ca_chemical3 <- ca_chemical2 %>%
  group_by(Chemical_Use, Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
## `.groups` argument.
```

```
ggplot(ca_chemical3, aes(x = Chemical_Use, y = Total_Value, fill = as.factor(Year))) +
  geom_col(position = "dodge") + # geom_col geom_bar(stat = "identity")
  labs(title = "Usage of Chemicals (CA, 2018-2023)",
       x = "Chemical Use",
       y = "Total Value",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

Usage of Chemicals (CA, 2018–2023)



```
library(dplyr)
library(ggplot2)
ca_chemical_agg <- ca_chemical2 %>%
  group_by(Chemical_Name, Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
tenchemicals <- function(data, year) {
  data %>%
    filter(Year == year) %>%
    arrange(desc(Total_Value)) %>%
    slice_head(n = 10)
}

top_10_2023 <- tenchemicals(ca_chemical_agg, 2023)
top_10_2021 <- tenchemicals(ca_chemical_agg, 2021)

print(top_10_2023)
```

```
## # A tibble: 10 x 3
##   Chemical_Name      Year Total_Value
##   <chr>            <int>     <dbl>
## 1 (CHLOROPICRIN      2023       692.
## 2 (ACETAMIPRID       2023       566.
## 3 (TOTAL)            2023       498
```

```
## 4 (THIAMETHOXAM      2023      478.
## 5 (CHLORANTRANILIPROLE 2023      475.
## 6 (ABAMECTIN          2023      447.
## 7 (POTASH)            2023      398.
## 8 (NITROGEN)          2023      366.
## 9 (DICHLOROPROPENE    2023      273.
## 10 (CAPTAN            2023      229.
```

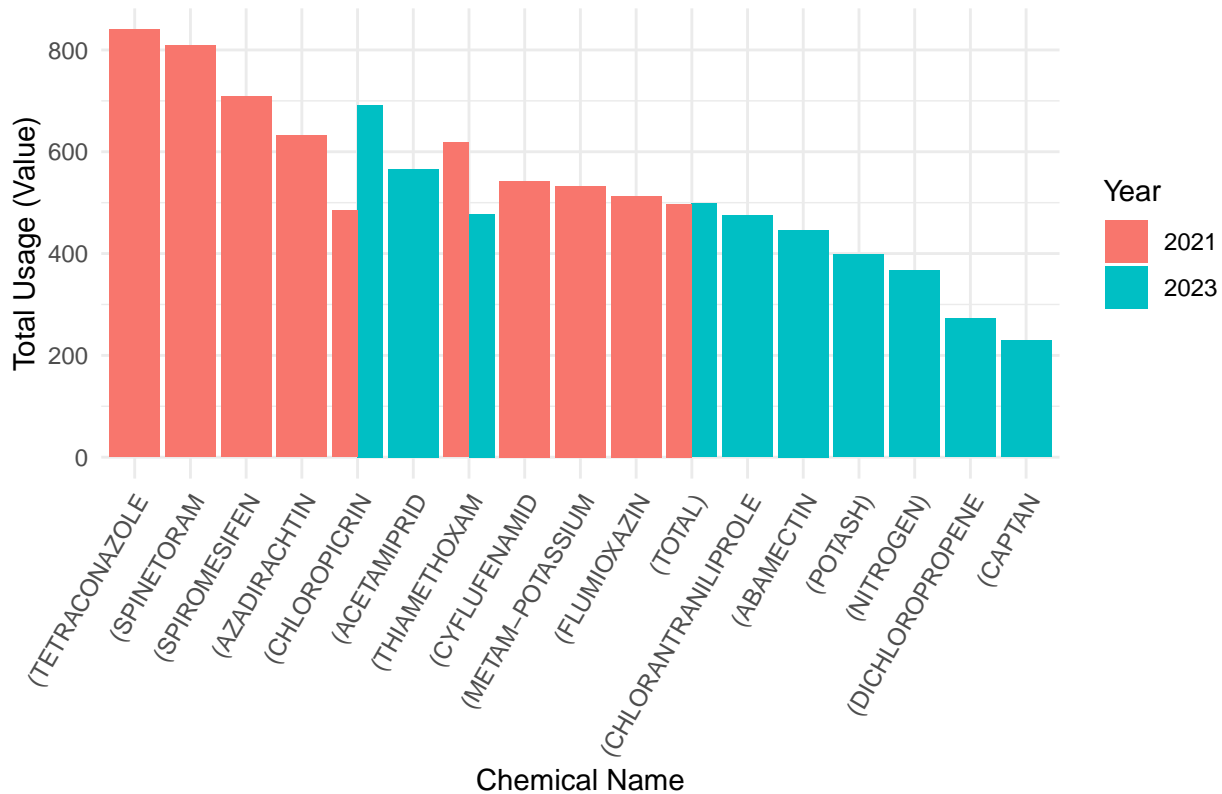
```
print(top_10_2021)
```

```
## # A tibble: 10 x 3
##   Chemical_Name   Year Total_Value
##   <chr>          <int>     <dbl>
## 1 (TETRACONAZOLE  2021      840.
## 2 (SPINETORAM    2021      809.
## 3 (SPIROMESIFEN  2021      709.
## 4 (AZADIRACHTIN  2021      632.
## 5 (THIAMETHOXAM  2021      619.
## 6 (CYFLUFENAMID  2021      542.
## 7 (METAM-POTASSIUM 2021      533.
## 8 (FLUMIOXAZIN   2021      513.
## 9 (TOTAL)        2021      497.
## 10 (CHLOROPICRIN  2021      485.
```

```
top_10_all <- bind_rows(
  top_10_2023 %>% mutate(Year = 2023),
  top_10_2021 %>% mutate(Year = 2021)
)
```

```
ggplot(top_10_all, aes(x = reorder(Chemical_Name, -Total_Value), y = Total_Value, fill = as.factor(Year))) +
  geom_col(position = "dodge") +
  labs(title = "Top 10 Chemicals by Total Usage for 2021 and 2023",
       x = "Chemical Name",
       y = "Total Usage (Value)",
       fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

Top 10 Chemicals by Total Usage for 2021 and 2023



```
library(dplyr)
library(ggplot2)

ca_chemical2_filtered <- ca_chemical2 %>%
  filter(Chemical_Use == "INSECTICIDE")

ca_chemical_agg1 <- ca_chemical2_filtered %>%
  group_by(Chemical_Name, Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  ungroup()
```

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

```
get_top_10 <- function(data, year) {
  data %>%
    filter(Year == year) %>%
    arrange(desc(Total_Value)) %>%
    slice_head(n = 10)
}

top_10_2023_new <- get_top_10(ca_chemical_agg1, 2023)
top_10_2021_new <- get_top_10(ca_chemical_agg1, 2021)

print(top_10_2023_new)
```

```
## # A tibble: 10 x 3
##   Chemical_Name      Year Total_Value
```

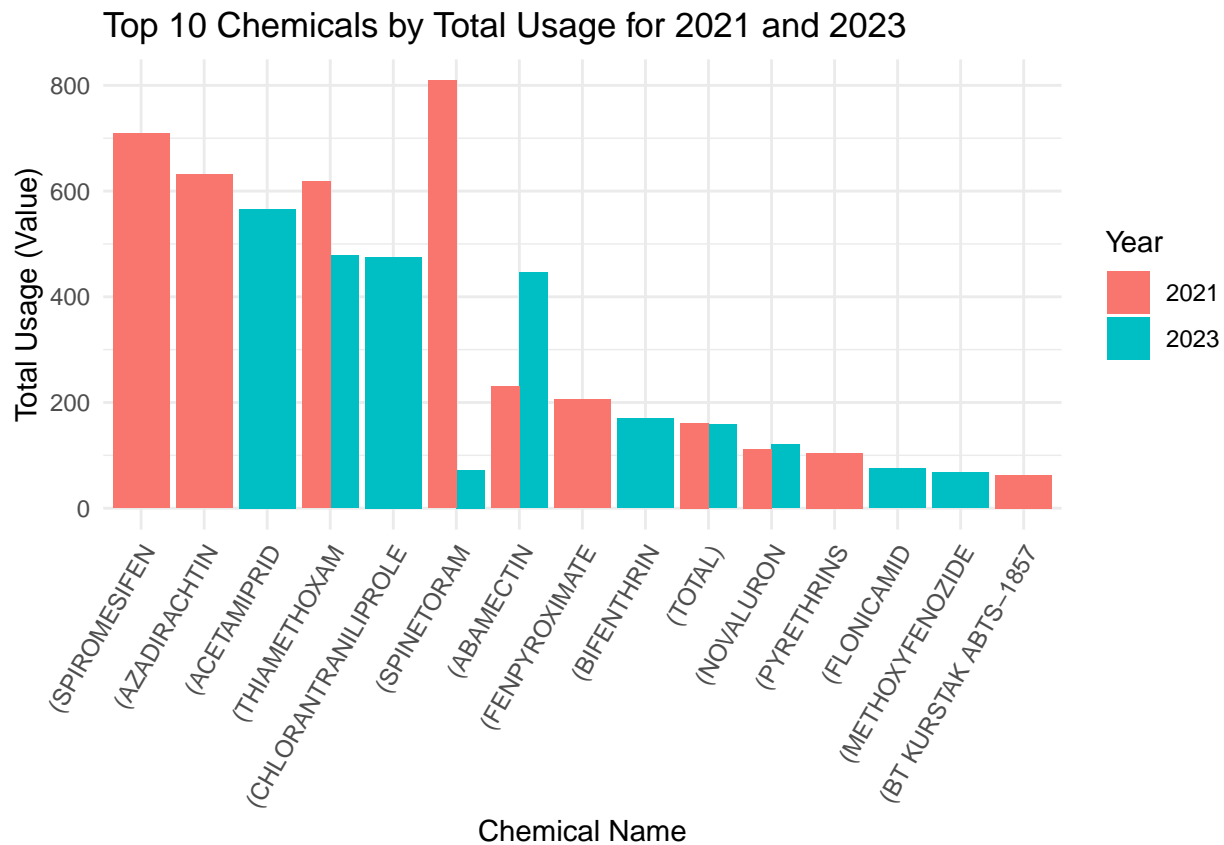
```
##      <chr>                <int>      <dbl>
##  1 (ACETAMIPRID           2023      566.
##  2 (THIAMETHOXAM          2023      478.
##  3 (CHLORANTRANILIPROLE   2023      475.
##  4 (ABAMECTIN             2023      447.
##  5 (BIFENTHRIN            2023      169.
##  6 (TOTAL)                2023      158
##  7 (NOVALURON             2023      121.
##  8 (FLONICAMID            2023       75.6
##  9 (SPINETORAM            2023       72.0
## 10 (METHOXYFENOZIDE        2023       68.0
```

```
print(top_10_2021_new)
```

```
## # A tibble: 10 x 3
##   Chemical_Name      Year Total_Value
##   <chr>             <int>      <dbl>
##  1 (SPINETORAM       2021      809.
##  2 (SPIROMESIFEN     2021      709.
##  3 (AZADIRACHTIN     2021      632.
##  4 (THIAMETHOXAM     2021      619.
##  5 (ABAMECTIN        2021      231.
##  6 (FENPYROXIMATE    2021      205.
##  7 (TOTAL)           2021      161
##  8 (NOVALURON         2021      112.
##  9 (PYRETHRINS       2021      105.
## 10 (BT KURSTAK ABTS-1857 2021       62.5
```

```
top_10_all_new <- bind_rows(
  top_10_2023_new %>% mutate(Year = 2023),
  top_10_2021_new %>% mutate(Year = 2021)
)
```

```
ggplot(top_10_all_new, aes(x = reorder(Chemical_Name, -Total_Value), y = Total_Value, fill = as.factor(
  geom_col(position = "dodge") +
  labs(title = "Top 10 Chemicals by Total Usage for 2021 and 2023",
    x = "Chemical Name",
    y = "Total Usage (Value)",
    fill = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

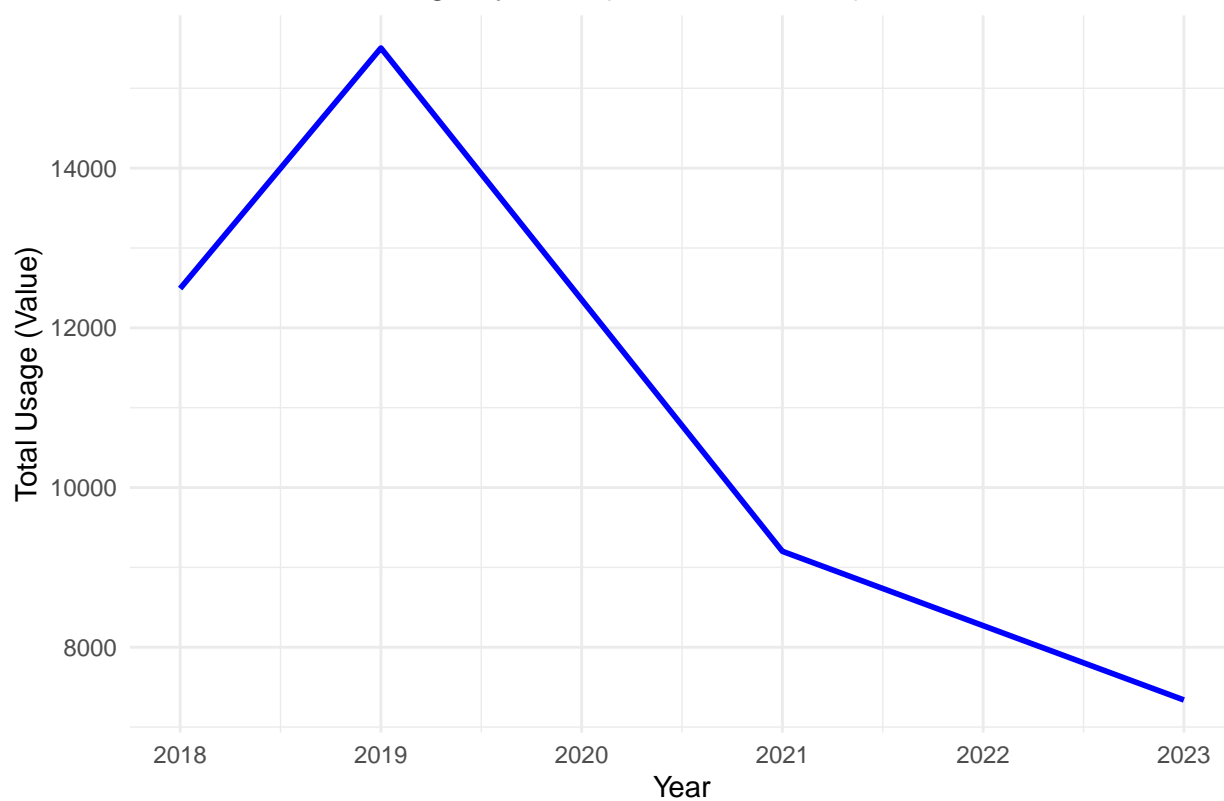


```
ca_chemical_total <- ca_chemical2 %>%
  group_by(Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))

ggplot(ca_chemical_total, aes(x = Year, y = Total_Value)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "Total Chemical Usage by Year (CA, 2018-2023)",
       x = "Year",
       y = "Total Usage (Value)") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

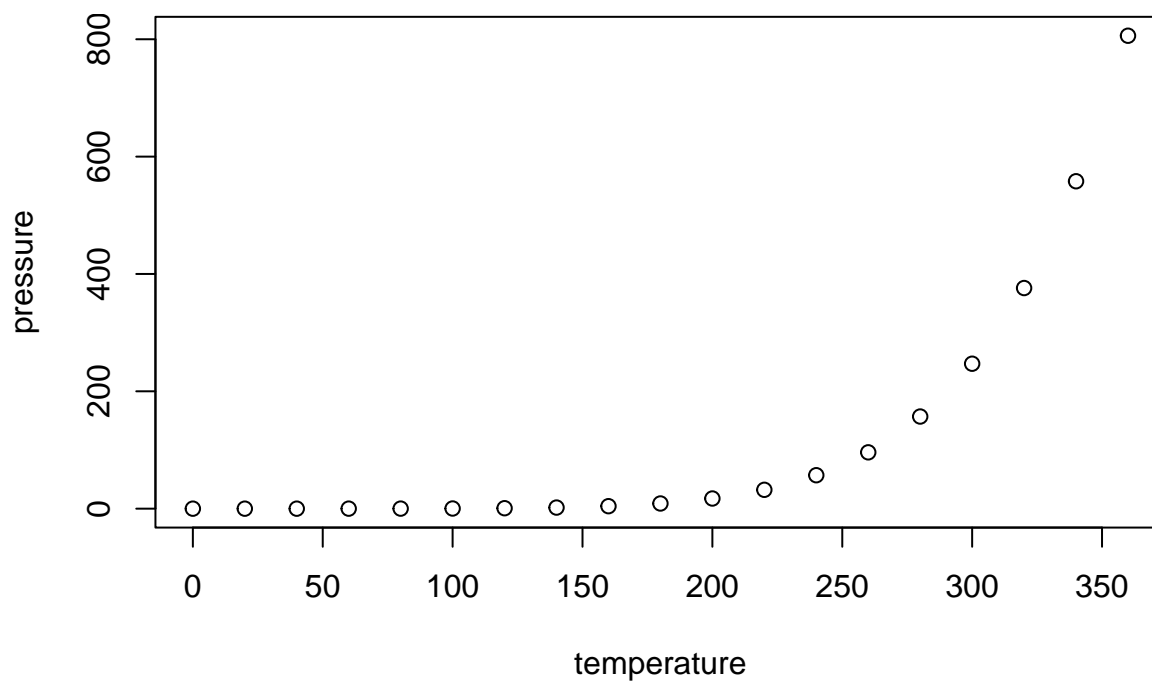
Total Chemical Usage by Year (CA, 2018–2023)



““

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.