chem

Guangyu Ding

2024-10-23

Consider the survey_d_chem file.

```
library(dplyr)
```

```
##
## 载入程序包：'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(tidyr)
library(ggplot2)

chem_data <- read_csv("survey_d_chem.csv")
```

```
## Rows: 3359 Columns: 8
```

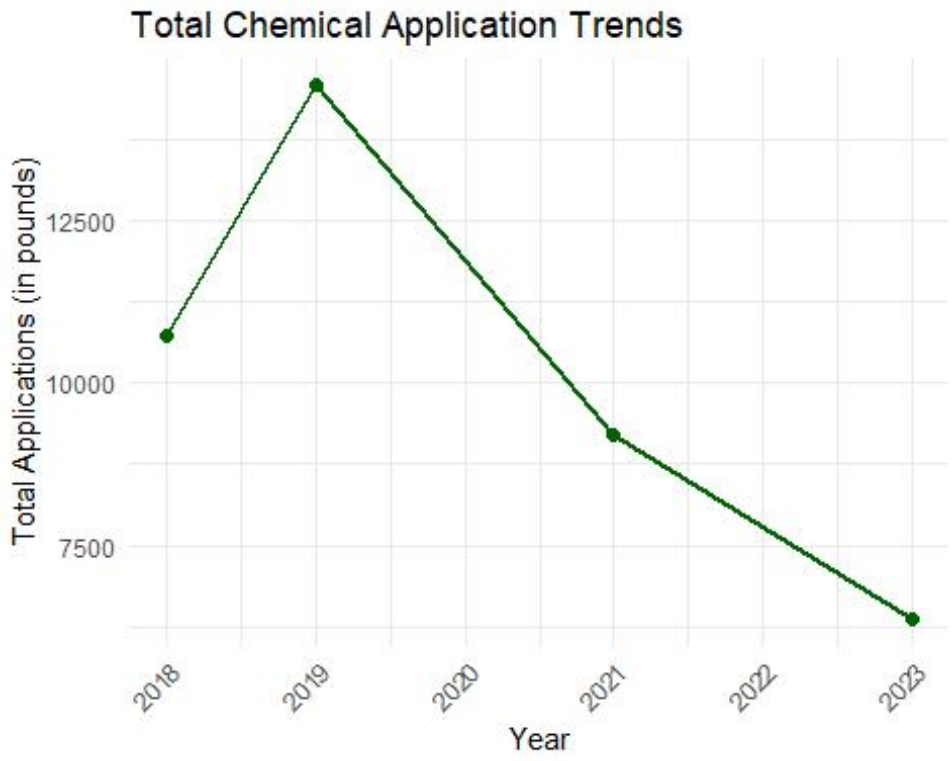
```
## ── Column specification ────────────────────────────────────────
────────────
## Delimiter: ","
## chr (7): State, mkt, measure, other, col2, Domain Category, Value
## dbl (1): Year
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```
## Filter data for California, because California is one of the major s
trawberry
## producing regions in the United States. Observing the data here is m
ore
## convincing.
chem_data_ca <- chem_data %>%
  filter(State == "CALIFORNIA")

# Clean the data by removing non-numeric values and converting to numer
ic
```

```r
chem_data_ca <- chem_data_ca %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))

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

# Summarize total applications per year for each chemical category
chem_summary <- chem_data_ca %>%
  group_by(Year, col2) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  spread(key = col2, value = Total_Value)

## `summarise()` has grouped output by 'Year'. You can override using t
he
## `.groups` argument.

# Print the summary
print(chem_summary)

## # A tibble: 4 × 5
## # Groups:   Year [4]
##    Year FUNGICIDE HERBICIDE INSECTICIDE OTHER
##   <dbl>     <dbl>     <dbl>       <dbl> <dbl>
## 1  2018     2349.      929.       5900. 1188.
## 2  2019     3512.      269.       4676. 2674.
## 3  2021     2523.      577.       3384. 1373.
## 4  2023     1397.      104.       1373. 1079.

## Visualize the data by making line chart
chem_data <- chem_data %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))

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

# Summarize total applications per year across all chemical types
chem_summary <- chem_data %>%
  group_by(Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))

ggplot(chem_summary, aes(x = Year, y = Total_Value)) +
  geom_line(color = "darkgreen", size = 1) +
  geom_point(color = "darkgreen", size = 2) +
  labs(title = "Total Chemical Application Trends",
       x = "Year",
```

```
       y = "Total Applications (in pounds)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## 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 warnin
g was
## generated.
```



Now separate the values by the categories listed in col2 column. Through this we can see which chemicals are being used more every year.

```
# Summarize total applications per year for each chemical category (col
2)
chem_summary1 <- chem_data %>%
  group_by(Year, col2) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Year'. You can override using t
he
## `.groups` argument.
```

```
# Spread the data to have categories as columns (if needed)
chem_summary_wide <- chem_summary1 %>%
```

```r
  spread(key = col2, value = Total_Value)

# Print the summary data for each chemical category
print(chem_summary1)
```

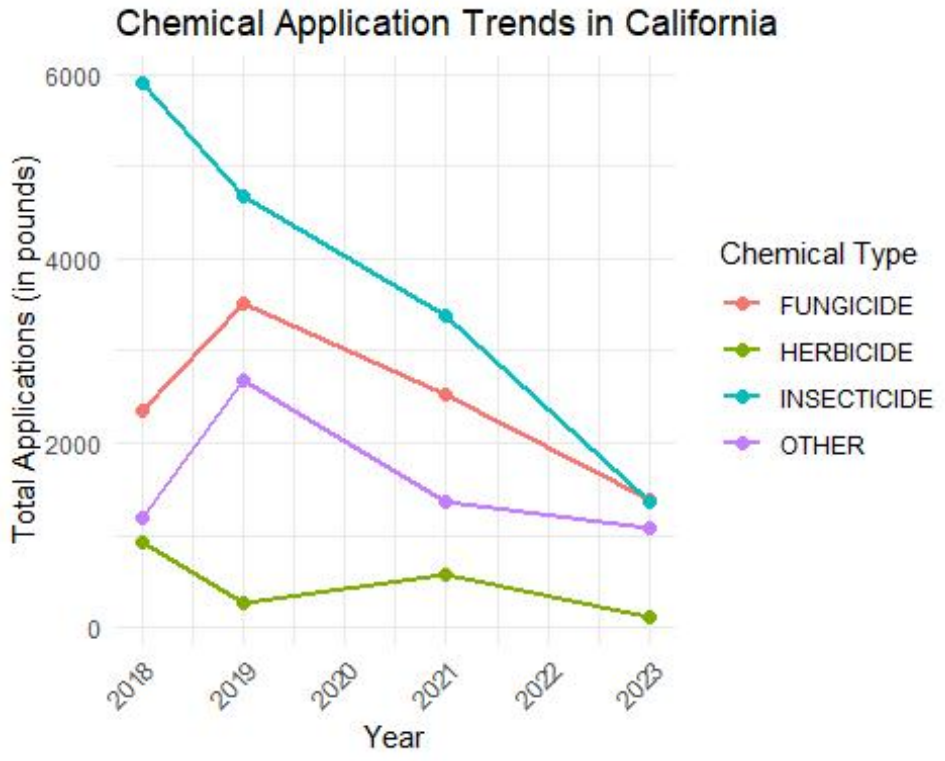
```
## # A tibble: 16 × 3
## # Groups:   Year [4]
##     Year col2        Total_Value
##    <dbl> <chr>             <dbl>
##  1  2018 FUNGICIDE         2682.
##  2  2018 HERBICIDE          929.
##  3  2018 INSECTICIDE       5937.
##  4  2018 OTHER             1188.
##  5  2019 FUNGICIDE         3997.
##  6  2019 HERBICIDE          584.
##  7  2019 INSECTICIDE       7305.
##  8  2019 OTHER             2691.
##  9  2021 FUNGICIDE         2919.
## 10  2021 HERBICIDE          586.
## 11  2021 INSECTICIDE       4275.
## 12  2021 OTHER             1422.
## 13  2023 FUNGICIDE         1989.
## 14  2023 HERBICIDE          130.
## 15  2023 INSECTICIDE       3130.
## 16  2023 OTHER             1101.
```

```r
## Obviously, Insecticides are the most commonly used chemical substanc
es for
## each year.

## We can see the trend of usage of each types of chemical substances b
y making
## a line chart to show all types of chemical substances used from 2018
 to 2023.
chem_data_ca <- chem_data_ca %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))

chem_summary <- chem_data_ca %>%
  group_by(Year, col2) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE)) %>%
  spread(key = col2, value = Total_Value)
```

```
## `summarise()` has grouped output by 'Year'. You can override using t
he
## `.groups` argument.
```

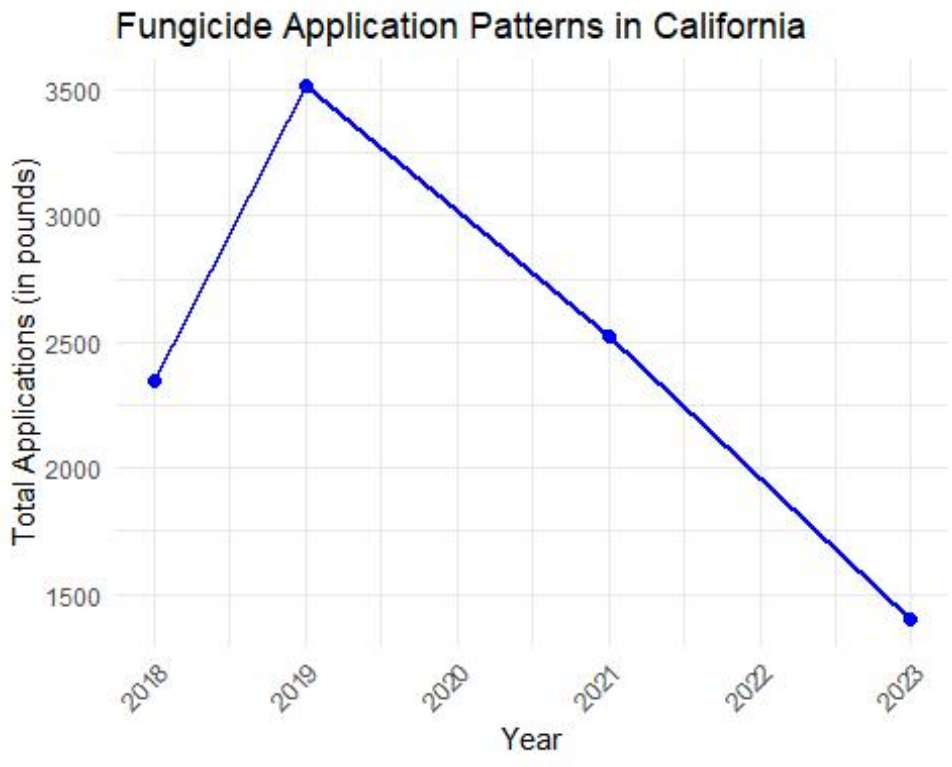
```r
chem_long <- chem_summary %>%
  gather(key = "Chemical_Type", value = "Total_Value", -Year)
```

```r
ggplot(chem_long, aes(x = Year, y = Total_Value, color = Chemical_Type,
 group = Chemical_Type)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Chemical Application Trends in California",
       x = "Year",
       y = "Total Applications (in pounds)",
       color = "Chemical Type") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
## Based on the plot, the usage of insecticide decreases from 2018 to 2
023.
## The fungicide had an increase in 2019, but decreased since then.
## The herbicide has an overall trend of decreasing, but experienced a
rebound in 2021.
## All other chemical substances has a similar trend with fungicide, bu
t decreased more from 2019 to 2021, less from 2021 to 2023.


## We can make more specific observation on each types of chemical subs
tances. i.e. Fungicide
# Filter data for California and fungicides
fungicide_data_ca <- chem_data %>%
  filter(State == "CALIFORNIA", col2 == "FUNGICIDE")
```

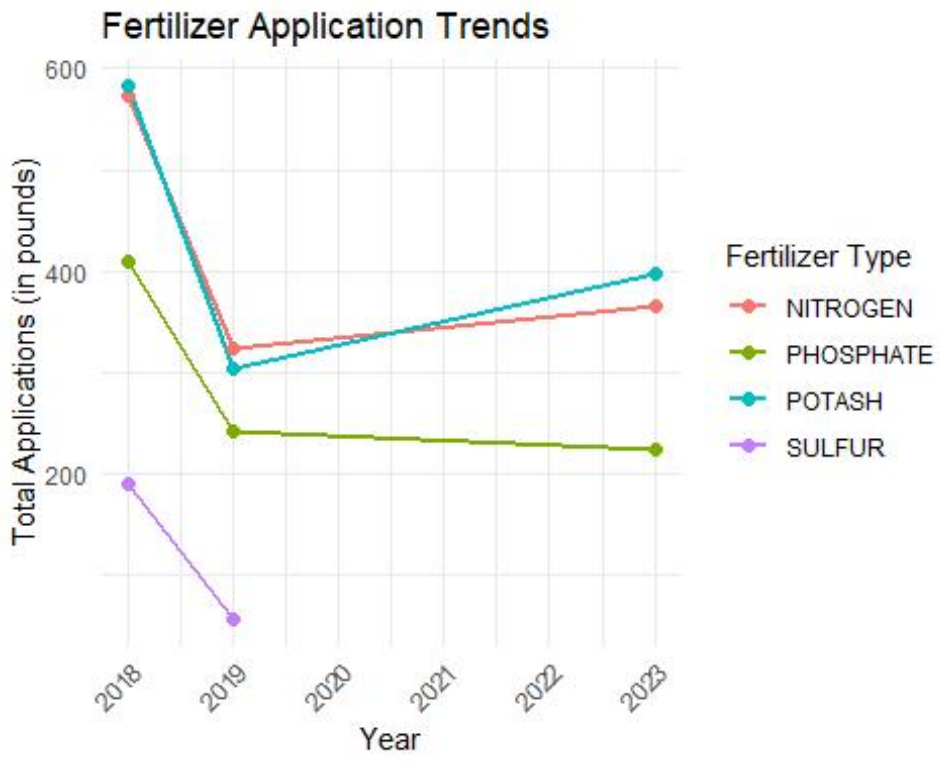
```r
# Clean the data by removing non-numeric values and converting to numer
ic
fungicide_data_ca <- fungicide_data_ca %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))

# Summarize total applications per year for fungicides
fungicide_summary <- fungicide_data_ca %>%
  group_by(Year) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))

# Plot the trends using ggplot2
ggplot(fungicide_summary, aes(x = Year, y = Total_Value)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "blue", size = 2) +
  labs(title = "Fungicide Application Patterns in California",
       x = "Year",
       y = "Total Applications (in pounds)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Now consider the suevey_d_fert file.By observing this file, we can get the usage of the fertilizer

```r
library(dplyr)
library(readr)
library(ggplot2)
```

```r
library(tidyr)

fert_data <- read_csv("survey_d_fert.csv")

## Rows: 115 Columns: 7
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (6): State, mk2, measure, other, chem_name, Value
## dbl (1): Year
##
## 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.

fert_data <- fert_data %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))

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

# Summarize total applications per year for each fertilizer type
fert_summary <- fert_data %>%
  group_by(Year, chem_name) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))

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

# Convert data to long format for easier plotting
fert_long <- fert_summary %>%
  spread(key = chem_name, value = Total_Value)

# Plot the trends
ggplot(fert_summary, aes(x = Year, y = Total_Value, color = chem_name, group = chem_name)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Fertilizer Application Trends",
       x = "Year",
       y = "Total Applications (in pounds)",
       color = "Fertilizer Type") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



By observation, from March 2018 to June 2020, nitrogen was the one being used

most. For the rest of time period from 2018 to 2023, Potash was at the first place.

Now visualize the usage of fertilizer in California. Compare the overall data

in the United States with the data in California.

```
library(dplyr)
library(readr)
library(ggplot2)
library(tidyr)

fert_data_ca <- read_csv("fert_data_ca.csv")
```

```
## Rows: 55 Columns: 7
## ── Column specification ────────────────────────────────────────────────
──────────
## Delimiter: ","
## chr (5): State, mk2, measure, other, chem_name
## dbl (2): Year, Value
##
## ℹ Use `spec()` to retrieve the full column specification for this d
```

```
ata.
## ℹ  Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```
summary(fert_data_ca)
```

```
##       Year         State               mk2              measure
##  Min.   :2018   Length:55          Length:55          Length:55
##  1st Qu.:2018   Class :character   Class :character   Class :charact
er
##  Median :2019   Mode  :character   Mode  :character   Mode  :charact
er
##  Mean   :2020
##  3rd Qu.:2023
##  Max.   :2023
##
##     other            chem_name            Value
##  Length:55          Length:55          Min.   :  2.00
##  Class :character   Class :character   1st Qu.: 11.10
##  Mode  :character   Mode  :character   Median : 21.00
##                                        Mean   : 57.99
##                                        3rd Qu.: 86.50
##                                        Max.   :315.00
##                                        NA's   :11
```

```
str(fert_data_ca)
```

```
## spc_tbl_ [55 × 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Year     : num [1:55] 2023 2023 2023 2023 2023 ...
##  $ State    : chr [1:55] "CALIFORNIA" "CALIFORNIA" "CALIFORNIA" "CAL
IFORNIA" ...
##  $ mk2      : chr [1:55] "APPLICATIONS" "APPLICATIONS" "APPLICATIONS
" "APPLICATIONS" ...
##  $ measure  : chr [1:55] "LB" "LB" "LB" "LB / ACRE / APPLICATION" ...
##  $ other    : chr [1:55] NA NA NA "AVG" ...
##  $ chem_name: chr [1:55] "NITROGEN" "PHOSPHATE" "POTASH" "NITROGEN"
 ...
##  $ Value    : num [1:55] NA NA NA 13 10 13 165 88 141 12.6 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Year = col_double(),
##   ..   State = col_character(),
##   ..   mk2 = col_character(),
##   ..   measure = col_character(),
```

```
##   ..   other = col_character(),
##   ..   chem_name = col_character(),
##   ..   Value = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(fert_data_ca)
```

```
## # A tibble: 6 × 7
##    Year State      mk2          measure                 other chem_n
ame Value
##   <dbl> <chr>      <chr>        <chr>                   <chr> <chr>
    <dbl>
## 1  2023 CALIFORNIA APPLICATIONS LB                      <NA>  NITROG
EN     NA
## 2  2023 CALIFORNIA APPLICATIONS LB                      <NA>  PHOSPH
ATE    NA
## 3  2023 CALIFORNIA APPLICATIONS LB                      <NA>  POTASH
       NA
## 4  2023 CALIFORNIA APPLICATIONS LB / ACRE / APPLICATION AVG   NITROG
EN     13
## 5  2023 CALIFORNIA APPLICATIONS LB / ACRE / APPLICATION AVG   PHOSPH
ATE    10
## 6  2023 CALIFORNIA APPLICATIONS LB / ACRE / APPLICATION AVG   POTASH
       13
```

```
fert_data_ca <- fert_data_ca %>%
  filter(!Value %in% c("(D)", "(NA)")) %>%
  mutate(Value = as.numeric(Value))
fert_summary <- fert_data_ca %>%
  group_by(Year, chem_name) %>%
  summarise(Total_Value = sum(Value, na.rm = TRUE))
```

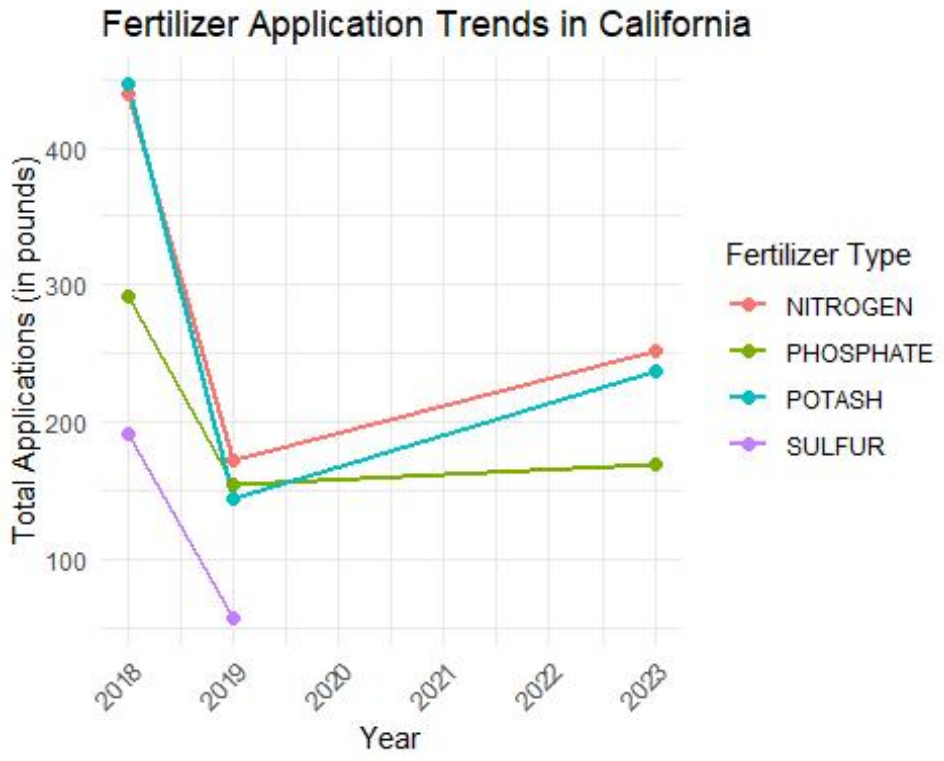
```
## `summarise()` has grouped output by 'Year'. You can override using t
he
## `.groups` argument.
```

```
print(fert_summary)
```

```
## # A tibble: 11 × 3
## # Groups:   Year [3]
##     Year chem_name Total_Value
##    <dbl> <chr>           <dbl>
##  1  2018 NITROGEN        439.
##  2  2018 PHOSPHATE       292.
##  3  2018 POTASH          446.
##  4  2018 SULFUR          191.
##  5  2019 NITROGEN        172.
##  6  2019 PHOSPHATE       154.
##  7  2019 POTASH          143.
##  8  2019 SULFUR           56.2
```

```
##  9  2023 NITROGEN        252.
## 10  2023 PHOSPHATE       170.
## 11  2023 POTASH          236.
```

```
# Visualize trends in fertilizer applications over the years
ggplot(fert_summary, aes(x = Year, y = Total_Value, color = chem_name,
group = chem_name)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Fertilizer Application Trends in California",
       x = "Year",
       y = "Total Applications (in pounds)",
       color = "Fertilizer Type") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We find that the usage of Nitrogen is ahead in California. Maybe other places

in United State can change from using potash to nitrogen.

Now consider the files fung, fung_ca_only and fung_fl_only to observe which chem is used most.

```
library(dplyr)
library(readr)
library(ggplot2)
```

```r
fung_fl <- read_csv("fung_fl_only.csv")
```

```
## Rows: 7 Columns: 2
## ── Column specification ────────────────────────────────────────────────────
────────────
## Delimiter: ","
## chr (1): chem_name
## dbl (1): chem_index
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```r
fung_data <- read_csv("fung.csv")
```

```
## Rows: 51 Columns: 2
## ── Column specification ────────────────────────────────────────────────────
────────────
## Delimiter: ","
## chr (1): chem_name
## dbl (1): chem_index
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```r
fung_ca <- read_csv("fung_ca_only.csv")
```

```
## Rows: 13 Columns: 2
## ── Column specification ────────────────────────────────────────────────────
────────────
## Delimiter: ","
## chr (1): chem_name
## dbl (1): chem_index
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

```r
summary(fung_fl)
```

```
##   chem_name           chem_index
##  Length:7           Min.   : 14504
##  Class :character   1st Qu.: 33901
##  Mode  :character   Median : 81901
##                     Mean   : 76499
```

```
##                    3rd Qu.:121089
##                    Max.   :129106

summary(fung_data)

##   chem_name           chem_index
##  Length:51          Min.   :  6327
##  Class :character   1st Qu.: 47091
##  Mode  :character   Median : 94655
##                     Mean   :105171
##                     3rd Qu.:128855
##                     Max.   :555550
##                     NA's   :1

summary(fung_ca)

##   chem_name           chem_index
##  Length:13          Min.   :  6327
##  Class :character   1st Qu.: 16482
##  Mode  :character   Median : 55459
##                     Mean   : 73429
##                     3rd Qu.:129068
##                     Max.   :230000

# Combine the data into one DataFrame for analysis
combined_fung <- bind_rows(fung_fl, fung_data, fung_ca)

# Summarize or clean the data if necessary (e.g., handle missing values)
combined_fung <- combined_fung %>%
  filter(!is.na(chem_index), !is.na(chem_name))

# Create a bar plot to visualize chem_name versus chem_index
ggplot(combined_fung, aes(x = chem_name, y = chem_index, fill = chem_na
me)) +
  geom_bar(stat = "identity") +
  labs(title = "Chem Index Values by Chem Name",
       x = "Chem Name",
       y = "Chem Index") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")

## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed ma
ximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```

ex Values by Chem Name

ILLUS SUBT. GB03	FENHEXAMID	PYR
ILLUS SUBTILIS	FLUDIOXONIL	PYR
D	FLUOPYRAM	QUI
AX DECAHYDRATE	FLUTOLANIL	STR
CALID	FLUXAPYROXAD	SUL
UBSP KURSTAKI EVB-113-19	FOSETYL-AL	TET
TAN	IPRODIONE	THI
OROTHALONIL	ISOFETAMID	THI
PER CHLORIDE HYD.	MANCOZEB	TRI
PER HYDROXIDE	MEFENOXAM	TRI
PER OCTANOATE	MONO-POTASSIUM SALT	TRI
LUFENAMID	MYCLOBUTANIL	

The plot produced seems not that useful. Here I think what we need to do is

to sort the data in the three fung tables in descending order to obtain the

usage quantities of each chemical substance nationwide, in California, and

in Florida.