Strawberries_HW

Jie Fei

2024-10-15

Data cleaning and organization

```
library(knitr)
library(kableExtra)
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::group_rows() masks kableExtra::group_rows()
## x 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 (15): Program, Period, Geo Level, State, State ANSI, Ag District, County...
## dbl  (2): Year, Ag District 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`        <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01~
## $ `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`       <chr> "011", "011", "011", "011", "011", "011", "101", "1~
## $ `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      <chr> "00000000", "00000000", "00000000", "00000000", "00~
## $ 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 == "STRAWBERRIES - PR
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)
```

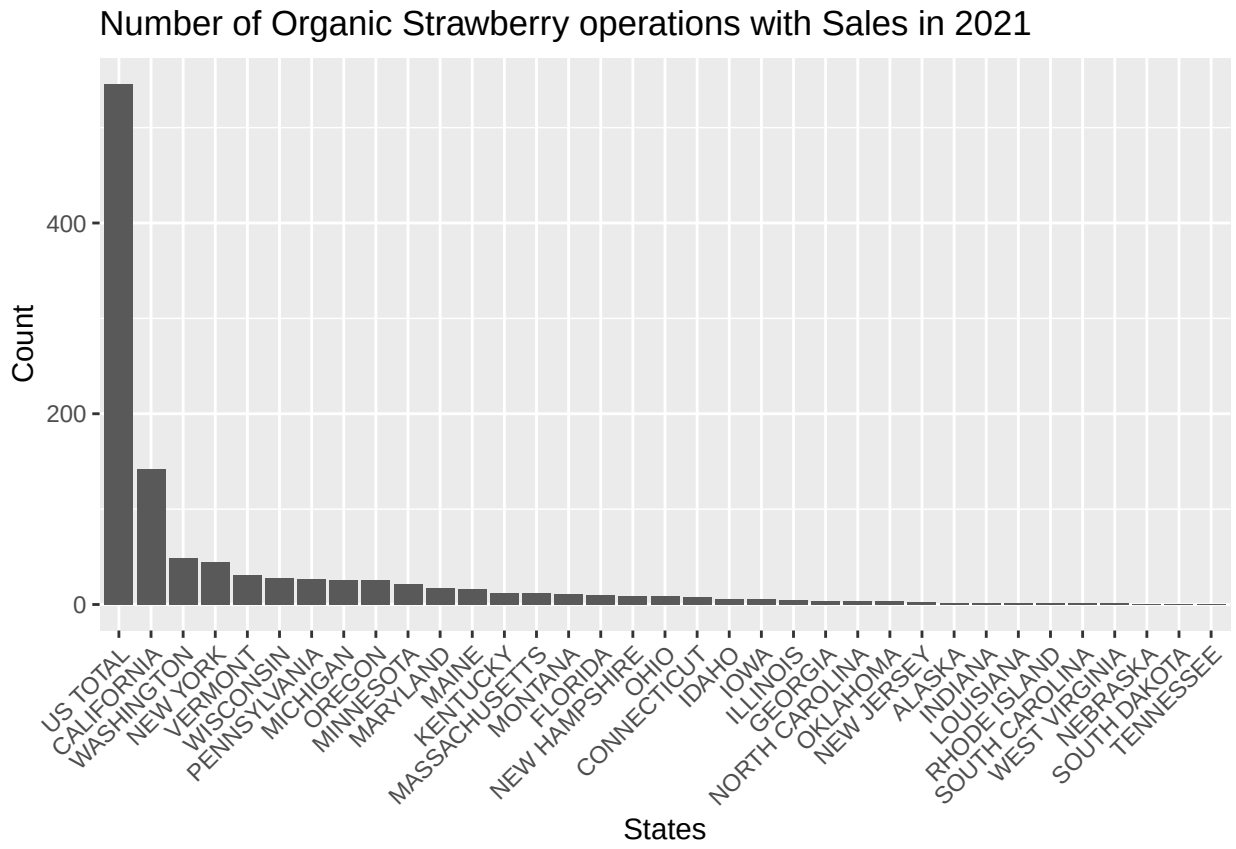
Data analysis and plots

```
# 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")
```

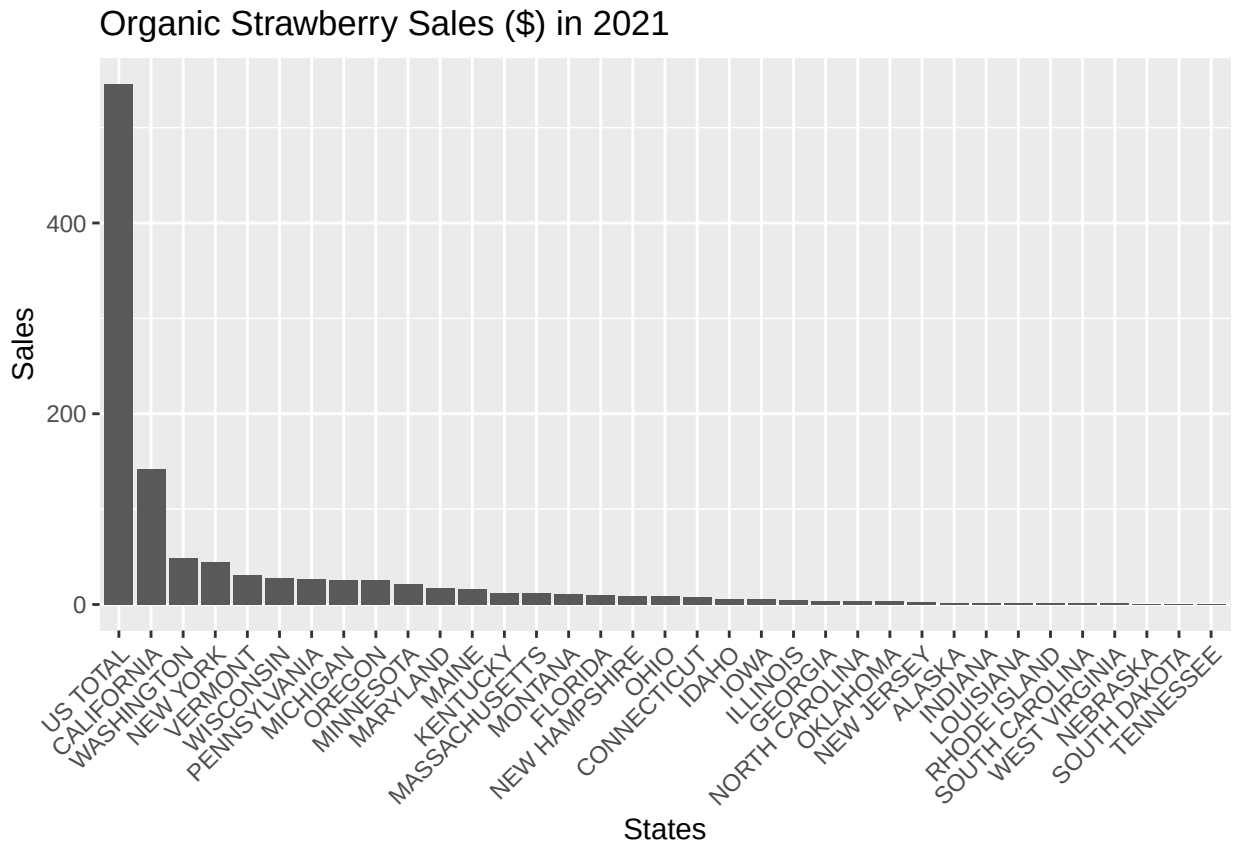



```
# Organic strawberry sales ($) in 2021
plot2_data <- strawberry |>
  select(c(Year, State, Category, Item, Value)) |>
  filter((Year == 2021) &
           (Category == "ORGANIC - SALES") &
           (Item == "MEASURED IN $") &
           (Value != "(D)"))

plot2_data$Value <- as.numeric(gsub(",", "", plot2_data$Value))

plot2_data <- plot1_data |> arrange(desc(Value))

ggplot(plot2_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 = "Sales",
title = "Organic Strawberry Sales ($) in 2021")
```




```
# Summary statistics by category and state
summary_data <- strawberry %>%
  group_by(State, Fruit) %>%
  summarise(total_value = sum(as.numeric(Value), na.rm = TRUE)) %>%
  arrange(desc(total_value))
```

```
## Warning: There were 218 warnings in ‘summarise()‘.
## The first warning was:
## i In argument: ‘total_value = sum(as.numeric(Value), na.rm = TRUE)‘.
## i In group 1: ‘State = "ALABAMA"‘ and ‘Fruit = "STRAWBERRIES"‘.
## Caused by warning:
## ! NAs introduced by coercion
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 217 remaining warnings.

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

```
print(summary_data)
```

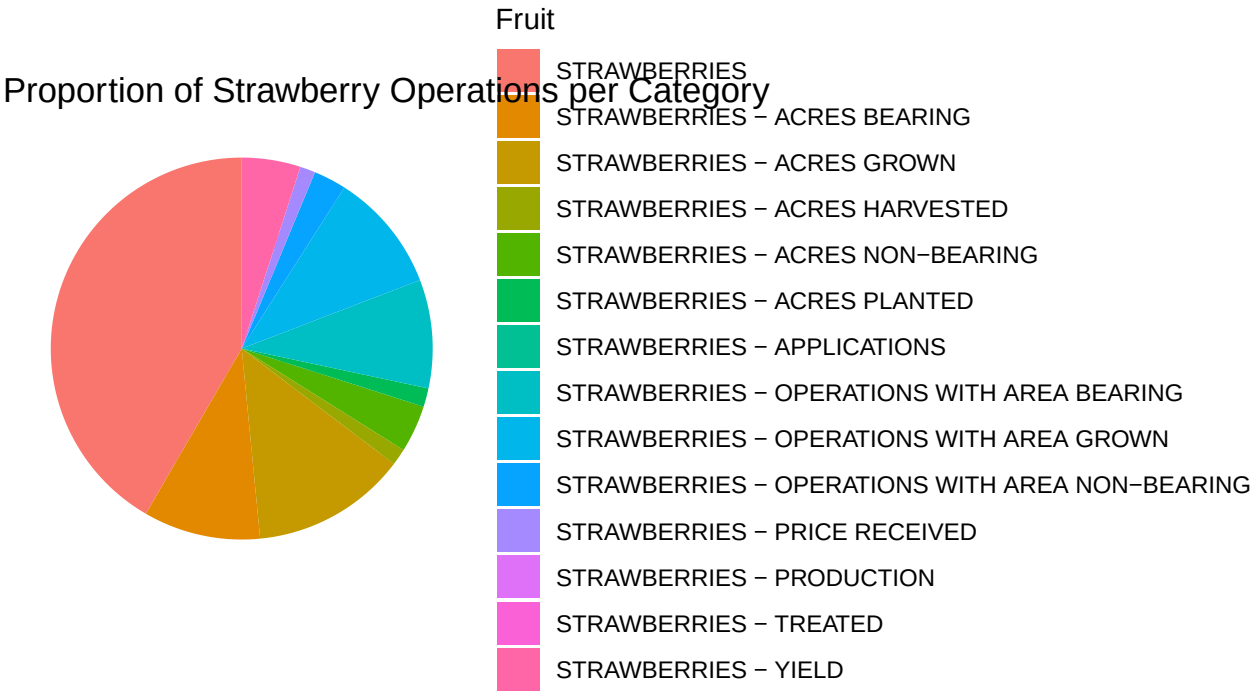
```
## # A tibble: 392 x 3
## # Groups:   State [52]
##    State          Fruit                          total_value
##    <chr>          <chr>                                <dbl>
##  1 CALIFORNIA     STRAWBERRIES                        37700.
```

```
##  2 FLORIDA        STRAWBERRIES                          10249.
##  3 US TOTAL       STRAWBERRIES                           6037.
##  4 CALIFORNIA     STRAWBERRIES - YIELD                   4505
##  5 US TOTAL       STRAWBERRIES - YIELD                   3897.
##  6 OREGON         STRAWBERRIES - ACRES GROWN             3457
##  7 NORTH CAROLINA STRAWBERRIES                           3316
##  8 MICHIGAN       STRAWBERRIES                           3191
##  9 US TOTAL       STRAWBERRIES - ACRES NON-BEARING       2753
## 10 PENNSYLVANIA   STRAWBERRIES                           2732
## # i 382 more rows
```

```r
# Pie chart: proportion of strawberry operations per fruit category
pie_data <- strawberry %>%
  group_by(Fruit) %>%
  summarise(total_value = sum(as.numeric(Value), na.rm = TRUE))
```

```
## Warning: There were 12 warnings in `summarise()`.
## The first warning was:
## i In argument: `total_value = sum(as.numeric(Value), na.rm = TRUE)`.
## i In group 1: `Fruit = "STRAWBERRIES"`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 11 remaining warnings.
```

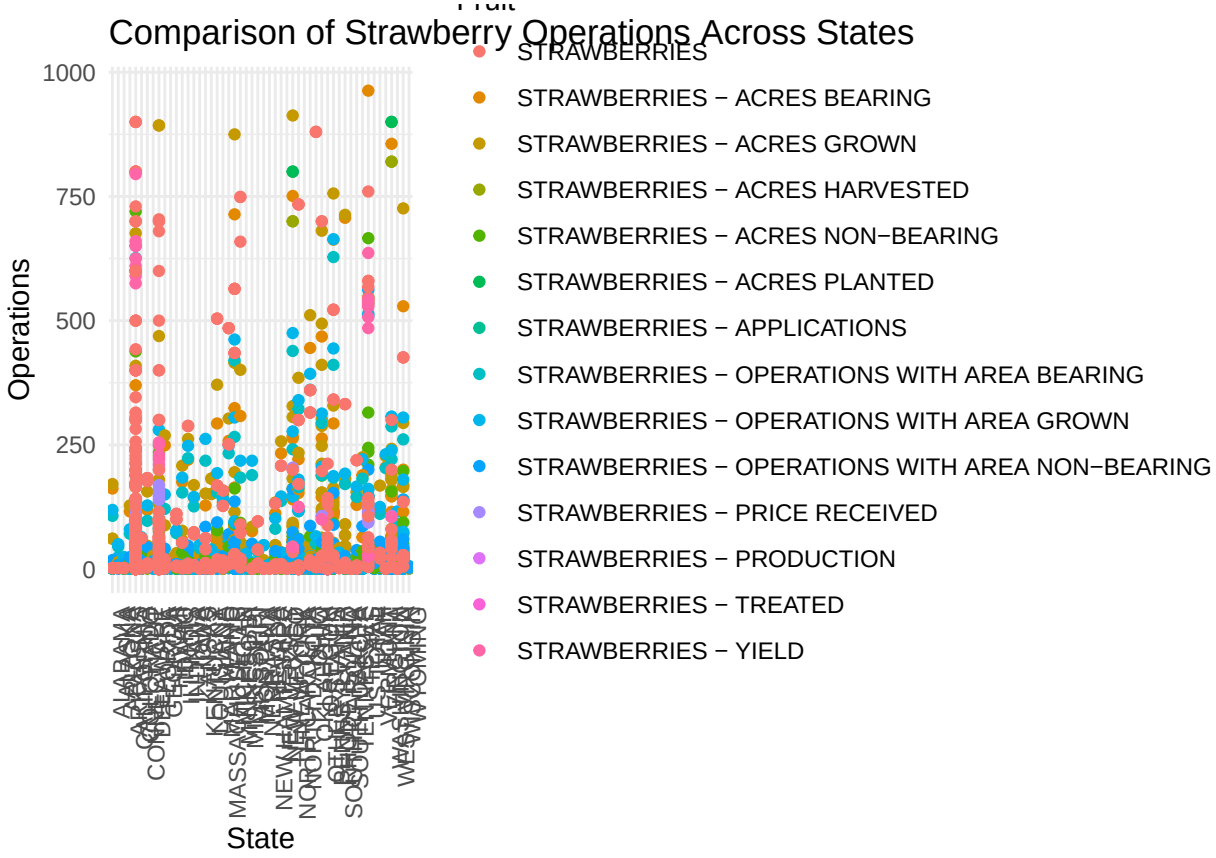
```r
ggplot(pie_data, aes(x = "", y = total_value, fill = Fruit)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y") +
  labs(title = "Proportion of Strawberry Operations per Category") +
  theme_void()
```




```
# Scatter plot: comparing operations across states
ggplot(strawberry, aes(x = State, y = as.numeric(Value), color = Fruit)) +
  geom_point() +
  labs(title = "Comparison of Strawberry Operations Across States", x = "State", y = "Operations") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
## Warning: Removed 5449 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Comparison of Strawberry Operations Across States
Operations
1000
750
500
250
0
State
Fruit
STRAWBERRIES
STRAWBERRIES − ACRES BEARING
STRAWBERRIES − ACRES GROWN
STRAWBERRIES − ACRES HARVESTED
STRAWBERRIES − ACRES NON−BEARING
STRAWBERRIES − ACRES PLANTED
STRAWBERRIES − APPLICATIONS
STRAWBERRIES − OPERATIONS WITH AREA BEARING
STRAWBERRIES − OPERATIONS WITH AREA GROWN
STRAWBERRIES − OPERATIONS WITH AREA NON−BEARING
STRAWBERRIES − PRICE RECEIVED
STRAWBERRIES − PRODUCTION
STRAWBERRIES − TREATED
STRAWBERRIES − YIELD