615midterm

JINGWU

2022-11-07

```
# midterm project
# Name: Jing Wu
# Date: Nov 5

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(readxl)

# read the data
raw_data <- read_xlsx("strawberries-2022oct30-a.xlsx", col_names = T)

# create a function to remove the columns which contains only 1 type content
col_remove <- function(df) {
  col1 <- colnames(df)
  col2 <- NULL
  for (i in 1:length(col1)) {
    if (dim(unique(df[col1[i]]))[1] != 1) {
      col2 <- append(col2, col1[i])
```

```
    }
  }
  data <- df %>%
    select(col2)
  return(data)
}
data1 <- col_remove(raw_data)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(col2)
##
##   # Now:
##   data %>% select(all_of(col2))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
# split the Item column into 4 columns
data2 <- data1 %>% separate(
  col = `Data Item`,
  into = c("Product", "Type", "Items", "Units"),
  sep = ",",
  fill = "right"
)

data2 <- unique(data2)

# remove the white spaces before the string after splitting
data2$Product <- sapply(data2$Product, str_trim)
data2$Type <- sapply(data2$Type, str_trim)
data2$Items <- sapply(data2$Items, str_trim)
data2$Units <- sapply(data2$Units, str_trim)

# split the data into organic and non-organic
organic_domain <- grep("organic", data2$Domain, ignore.case = T)
organic_type <- grep("organic", data2$Type, ignore.case = T)
organic_items <- grep("organic", data2$Items, ignore.case = T)
organic_domain_category <- grep("organic", data2$`Domain Category`, ignore.case = T)

# test if the column which contains "organic" is same
intersect(organic_domain, organic_domain_category) == organic_type
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [61] TRUE TRUE
```

```
# split the data into organic and non-organic
organic_data <- data2[organic_type, ]
non_organic_data <- data2[setdiff(1:2234, organic_type), ]
```

```
rm(data1, data2)

# organize the organic data
# first remove the columns only contain 1 kind of contend
organic_data <- col_remove(organic_data)

# observe other columns
print(unique(organic_data$Items))
```

```
## [1] "MEASURED IN $"        "MEASURED IN CWT"      "FRESH MARKET - SALES"
## [4] "PROCESSING - SALES"
```

```
print(unique(organic_data$Type))
```

```
## [1] "ORGANIC - SALES" "ORGANIC"
```

```
print(unique(organic_data$Units))
```

```
## [1] NA                "MEASURED IN $"   "MEASURED IN CWT"
```

```
# It seems like some units appear in the Items column.
# figure out whether rows where Units is NA and rows where Units values in Items are same
sum(is.na(organic_data$Units)) == length(grep("MEASURED IN", organic_data$Items))
```

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

```
as.numeric(which(is.na(organic_data$Units))) == grep("MEASURED IN", organic_data$Items)
```

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

```
# It is true.
# fill the units in Items into the NA of Units
na_row <- as.numeric(which(is.na(organic_data$Units)))
for (i in na_row) {
  organic_data$Units[i] <- organic_data$Items[i]
  organic_data$Items[i] <- NA
}

# we remove all "MEASURED IN" in the Units
organic_data$Units <- str_remove_all(organic_data$Units, "MEASURED IN ")

# Use same method to deal with NA in Items and values in Type
sum(is.na(organic_data$Items)) == length(grep("SALES", organic_data$Type, ignore.case = TRUE))
```

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

```
as.numeric(which(is.na(organic_data$Items))) == grep("SALES", organic_data$Type, ignore.case = TRUE)
```

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

```
# fill the Items in Type column into Type
na_row2 <- as.numeric(which(is.na(organic_data$Items)))
for (i in na_row2) {
  organic_data$Items[i] <- organic_data$Type[i]
}

# remove the Type column
organic_data %<>% select(!Type)

# normalize the digits in column CV
organic_data %>%
  filter(nchar(organic_data$`CV (%)`) > 10)
```

```
## # A tibble: 4 x 7
##    Year State      `State ANSI` Items                Units Value   `CV (%)`
##   <dbl> <chr>             <dbl> <chr>                <chr> <chr>   <chr>
## 1  2019 CALIFORNIA            6 FRESH MARKET - SALES CWT   1177214 33.700000000~
## 2  2019 NEW JERSEY           34 FRESH MARKET - SALES CWT   208     33.799999999~
## 3  2016 NEW YORK             36 ORGANIC - SALES      CWT   1012    19.600000000~
## 4  2016 OREGON               41 PROCESSING - SALES   CWT   3852    39.299999999~
```

```
# It seems like some values is in strange precision
# round them up
row_round <- which(nchar(organic_data$`CV (%)`) > 10)
for (i in row_round) {
  organic_data$`CV (%)`[i] <- as.character(as.numeric(organic_data$`CV (%)`[i]))
}

# deal with the NA and non-numeric values in Value and CV
organic_data$Value <- as.numeric(organic_data$Value)
```

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

```
organic_data$`CV (%)` <- as.numeric(organic_data$`CV (%)`)
```

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

```
# convert the table by pivot_wider
organic_data %<>% pivot_wider(names_from = Items, values_from = c(Value, `CV (%)`))

# remove the rows when 6 values are all NA
row_number <- NULL
for (i in 1:24) {
  if (all(is.na(organic_data[i, 5:10]))) {
    row_number <- append(row_number, i)
```

```
  }
}
organic_data <- organic_data[-row_number, ]

# It is done!
# remove unnecessary variables
rm(organic_domain_category, organic_domain, organic_items, organic_type, na_row, na_row2, row_round)

# clean the non-organic data
# find out data about chemistry
unique(non_organic_data$Domain)
```

```
## [1] "TOTAL"                 "CHEMICAL, FUNGICIDE"   "CHEMICAL, HERBICIDE"
## [4] "CHEMICAL, INSECTICIDE" "CHEMICAL, OTHER"       "FERTILIZER"
```

```
unique(non_organic_data$Type)
```

```
## [1] "MEASURED IN $ / CWT"           "FRESH MARKET - PRICE RECEIVED"
## [3] "PROCESSING - PRICE RECEIVED"   "BEARING - APPLICATIONS"
```

```
domain_chem <- as.numeric(which(non_organic_data$Domain != "TOTAL"))
type_chem <- as.numeric(which(non_organic_data$Type == "BEARING - APPLICATIONS"))
same <- intersect(domain_chem, type_chem)
length(same) == length(type_chem)
```

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

```
# split the data as chemical data and non-organic sale data
chem_data <- non_organic_data %>% slice(type_chem, preserve = FALSE)
non_organic_sale <- non_organic_data %>% slice(setdiff(1:2172, type_chem), preserve = FALSE)

# clean the chemical data
# split the Domain Category into Chemical domain and Chemical type
chem_data %<>% separate(
  col = `Domain Category`, into = c("Chemical domain", "Chemical type"),
  sep = ":",
  fill = "right"
)

# remove the white space and '(' ')' of Chemical Type
chem_data$`Chemical type` <- gsub("[()]", "", chem_data$`Chemical type`)
chem_data$`Chemical type` <- sapply(chem_data$`Chemical type`, str_trim)

# continue to split the Chemical type into name and code
chem_data %<>% separate(
  col = `Chemical type`, into = c("Name", "Code"),
  sep = "=",
  fill = "right"
)

# test if the Domain and Chemical Domain have save values
sum(chem_data$Domain == chem_data$`Chemical domain`) == length(chem_data$Domain)
```

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

```
# remove Chemical Domain as it is totally same with Domain
chem_data %<>% select(!`Chemical domain`)

# remove columns which only contain 1 kind of values
chem_data <- col_remove(chem_data)

# remove all "measured in"
chem_data$Items <- str_remove_all(chem_data$Items, "MEASURED IN ")

# test if when Units is null, Items is "LB"
t1 <- which(chem_data$Items == "LB")
t2 <- which(chem_data$Items == "LB") == as.numeric(which(is.na(chem_data$Units)))
length(t1) == length(t2)
```

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

```
# fill the lb into NA
chem_data$Units[t1] <- chem_data$Items[t1]

# remove the CHEMICAL string and rename Items into Type
chem_data$Domain <- str_remove_all(chem_data$Domain, "CHEMICAL, ")
chem_data %<>% rename(Type = Items)

# take unique values from the table
chem_data <- unique(chem_data)

# normalize the digits in column Value
chem_data %>%
  filter(nchar(chem_data$Value) > 10)
```

```
## # A tibble: 345 x 9
##     Year State      `State ANSI` Type             Units Domain Name  Code  Value
##    <dbl> <chr>             <dbl> <chr>            <chr> <chr>  <chr> <chr> <chr>
##  1  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "AZO~ " 12~ 0.23~
##  2  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "BOR~ " 11~ 1.70~
##  3  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "CAP~ " 81~ 1.66~
##  4  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "CYP~ " 28~ 0.33~
##  5  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "FLU~ " 13~ 0.17~
##  6  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "MEF~ " 11~ 0.48~
##  7  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "PEN~ " 90~ 0.27~
##  8  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "POL~ " 23~ 9.80~
##  9  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "POT~ " 73~ 2.15~
## 10  2021 CALIFORNIA            6 LB / ACRE / APP~ AVG   FUNGI~ "PYR~ " 99~ 0.17~
## # ... with 335 more rows
```

```
# It seems like some values is in strange precision
# round them up
row_round <- which(nchar(chem_data$Value) > 10)
for (i in row_round) {
  chem_data$Value[i] <- as.character(as.numeric(chem_data$Value[i]))
```

```
}

# convert Value into double type
chem_data$Value <- as.numeric(chem_data$Value)
```

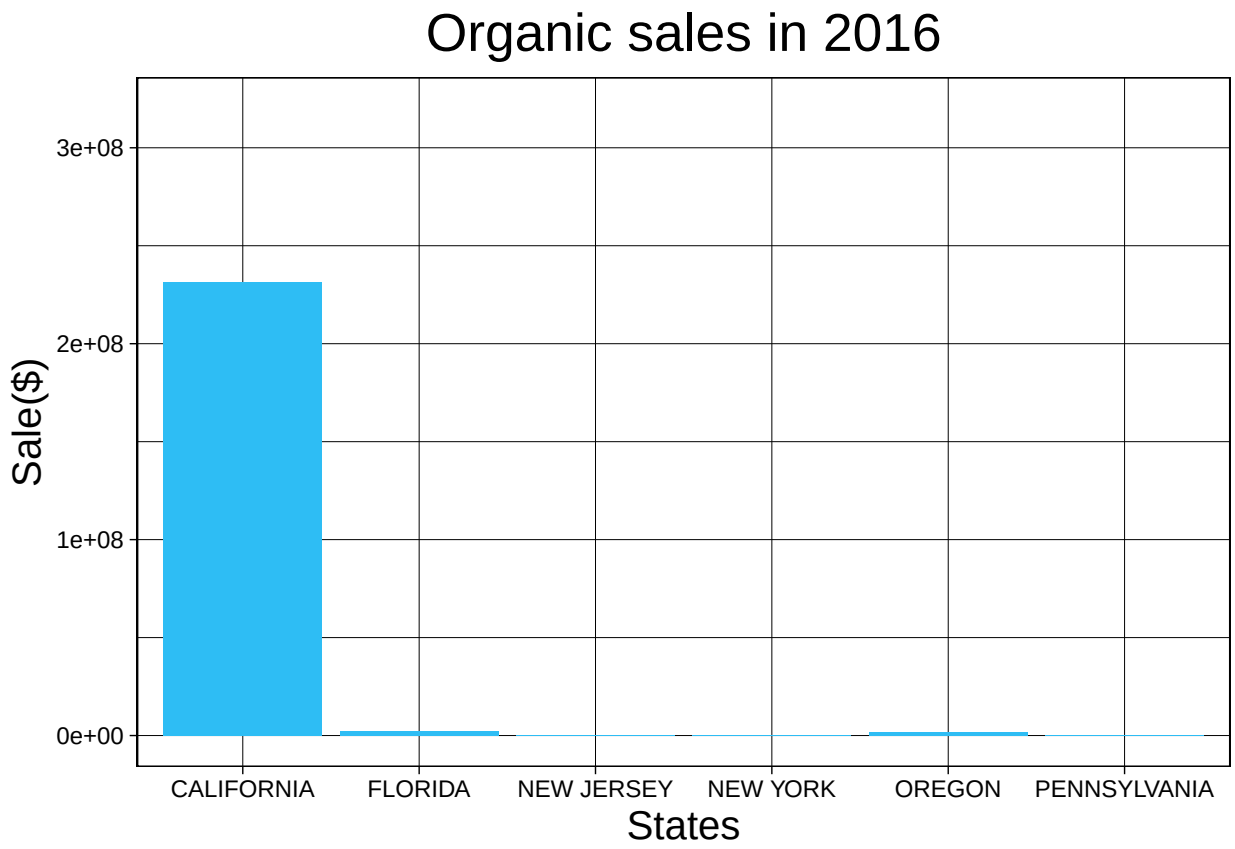
```
## Warning: NAs introduced by coercion
```

```
# use pivot_wider to organize data
chem_data %<>% pivot_wider(names_from = Year, values_from = Value, names_sort = TRUE)

# clean the data where values are NA in all year
chem_data %<>% filter(!(is.na(`2016`) & is.na(`2018`) & is.na(`2019`) & is.na(`2021`)))

# remove white spaces
chem_data$Name <- sapply(chem_data$Name, str_trim)
chem_data$Code <- sapply(chem_data$Code, str_trim)


# It is done!

rm(domain_chem, row_round, same, t1, t2, type_chem)

# Finally, clean the non-organic sale data
# clean the Type Items and Units with same method as organic data
na_row <- as.numeric(which(is.na(non_organic_sale$Items)))
for (i in na_row) {
  non_organic_sale$Items[i] <- non_organic_sale$Type[i]
  non_organic_sale$Type[i] <- non_organic_sale$Product[i]
}

# delete the Units and Product column, and rename the columns
non_organic_sale %<>% select(!c(Units, Product))
non_organic_sale %<>% rename(Units = Items, Items = Type)

# remove "MEASURED IN"
non_organic_sale$Units <- str_remove_all(non_organic_sale$Units, "MEASURED IN ")

# remove the columns which contain only 1 kind of content
non_organic_sale <- col_remove(non_organic_sale)

# convert the values into numeric type
non_organic_sale$Value <- as.numeric(non_organic_sale$Value)
```

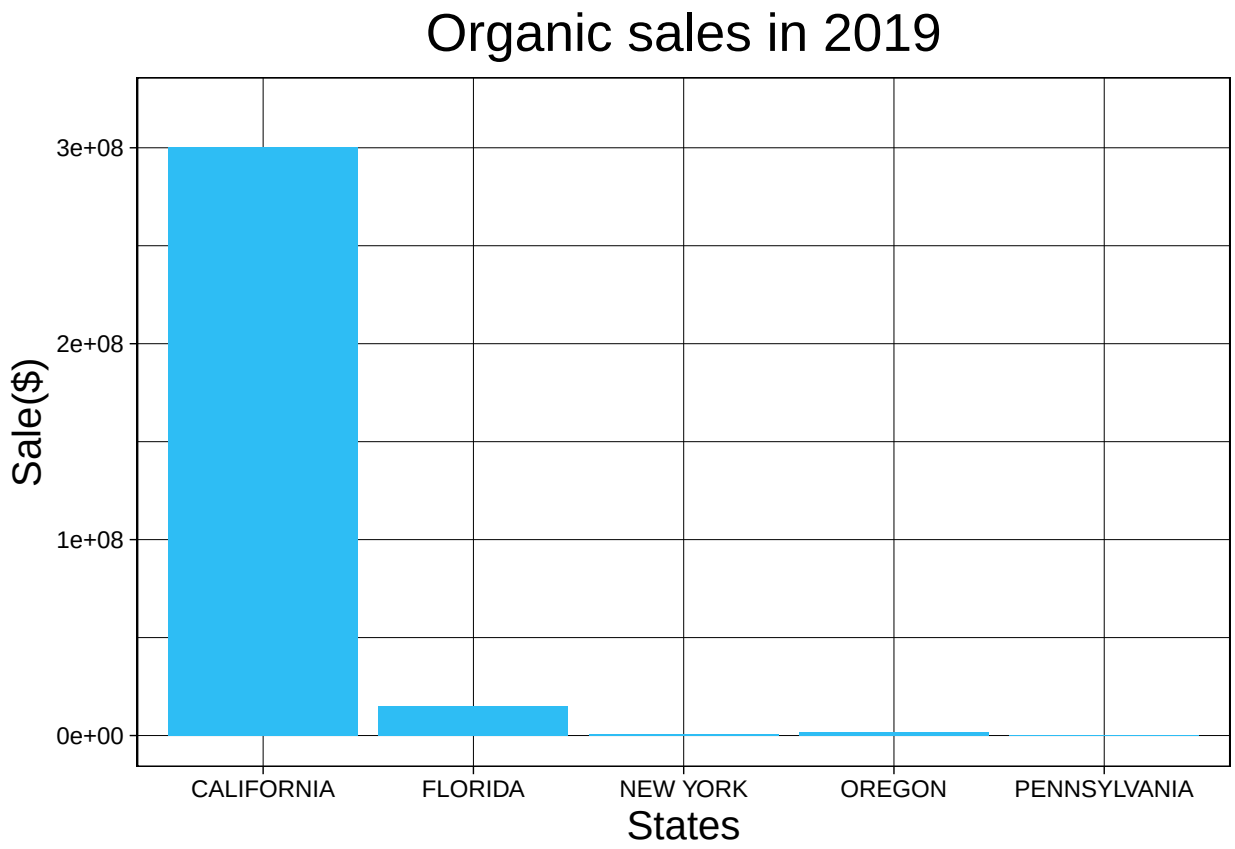
```
## Warning: NAs introduced by coercion
```

```
# convert the table by pivot_wider
non_organic_sale %<>% pivot_wider(names_from = Items, values_from = Value, names_sort = TRUE)

# remove the rows when 3 values are all NA
row_number <- NULL
for (i in 1:24) {
  if (all(is.na(non_organic_sale[i, 5:7]))) {
```

```
    row_number <- append(row_number, i)
  }
}
non_organic_sale <- non_organic_sale[-row_number, ]

# It is done!

rm(na_row, non_organic_data, row_number)


# data cleaning finished
# EDA
# explore the sales values in different state in 2016,2019
data1 <- organic_data %>%
  filter(Units == "$") %>%
  select(State, Year, `Value_ORGANIC - SALES`, `Value_FRESH MARKET - SALES`, `Value_PROCESSING - SALES`)

data1_2016 <- data1 %>%
  filter(Year == "2016")

# plot organic sales VS states in 2016
ggplot(data1_2016) +
  aes(x = State, y = `Value_ORGANIC - SALES`) +
  geom_col(fill = "#2EBDF4") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic sales in 2016"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 320000000)
```

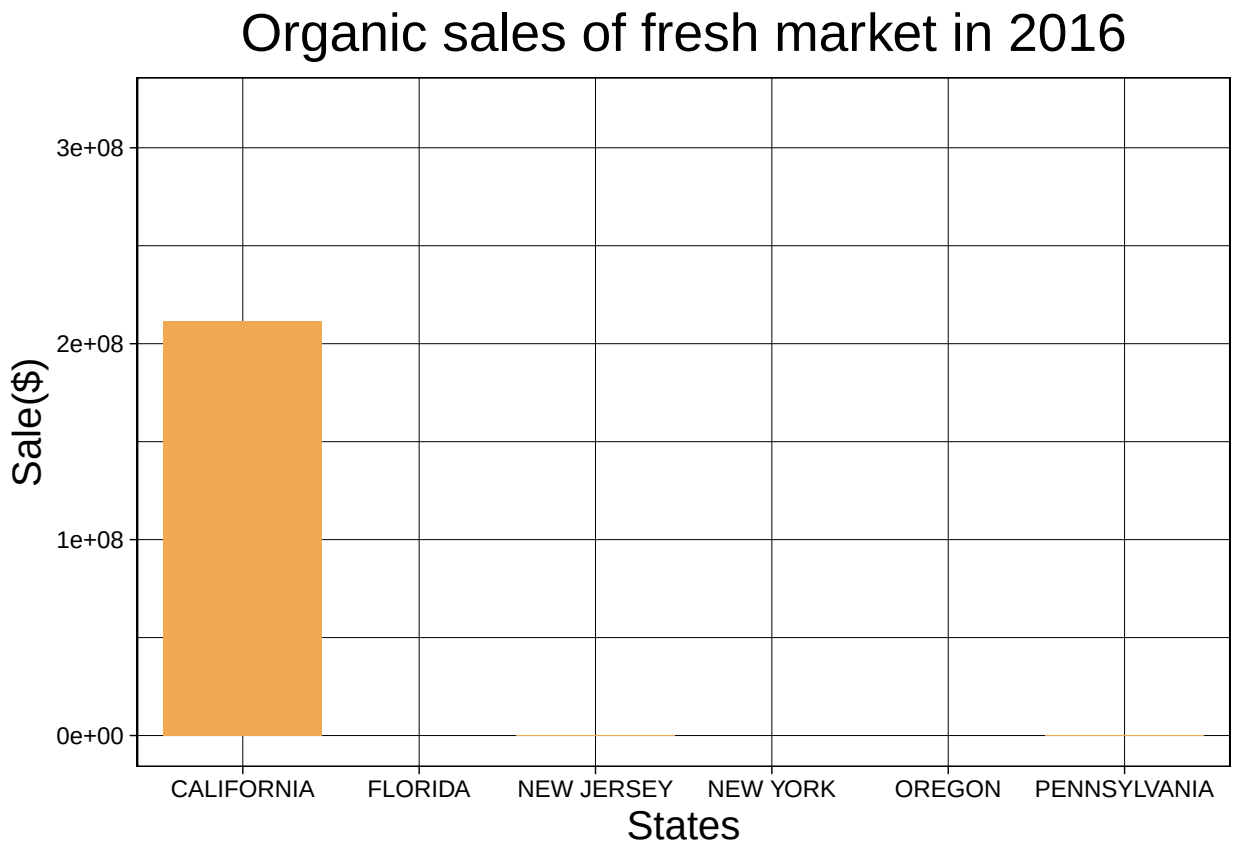



```
# plot organic sales VS states in 2019
data1_2019 <- data1 %>%
  filter(Year == "2019")

ggplot(data1_2019) +
  aes(x = State, y = `Value_ORGANIC - SALES`) +
  geom_col(fill = "#2EBDF4") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic sales in 2019"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 320000000)
```



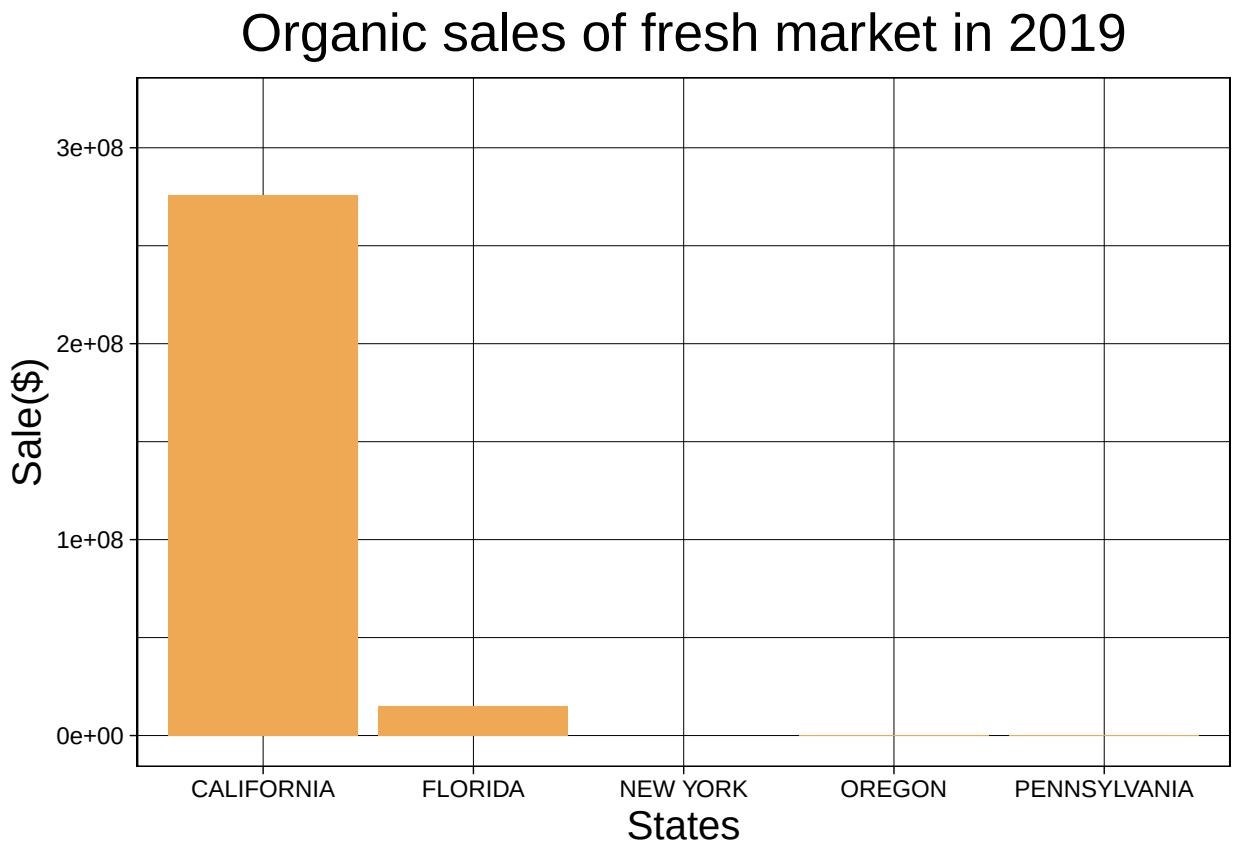

```
# plot organic sales of fresh market VS states in 2016 and 2019
ggplot(data1_2016) +
  aes(x = State, y = `Value_FRESH MARKET - SALES`) +
  geom_col(fill = "#EFA954") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic sales of fresh market in 2016"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 320000000)
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```



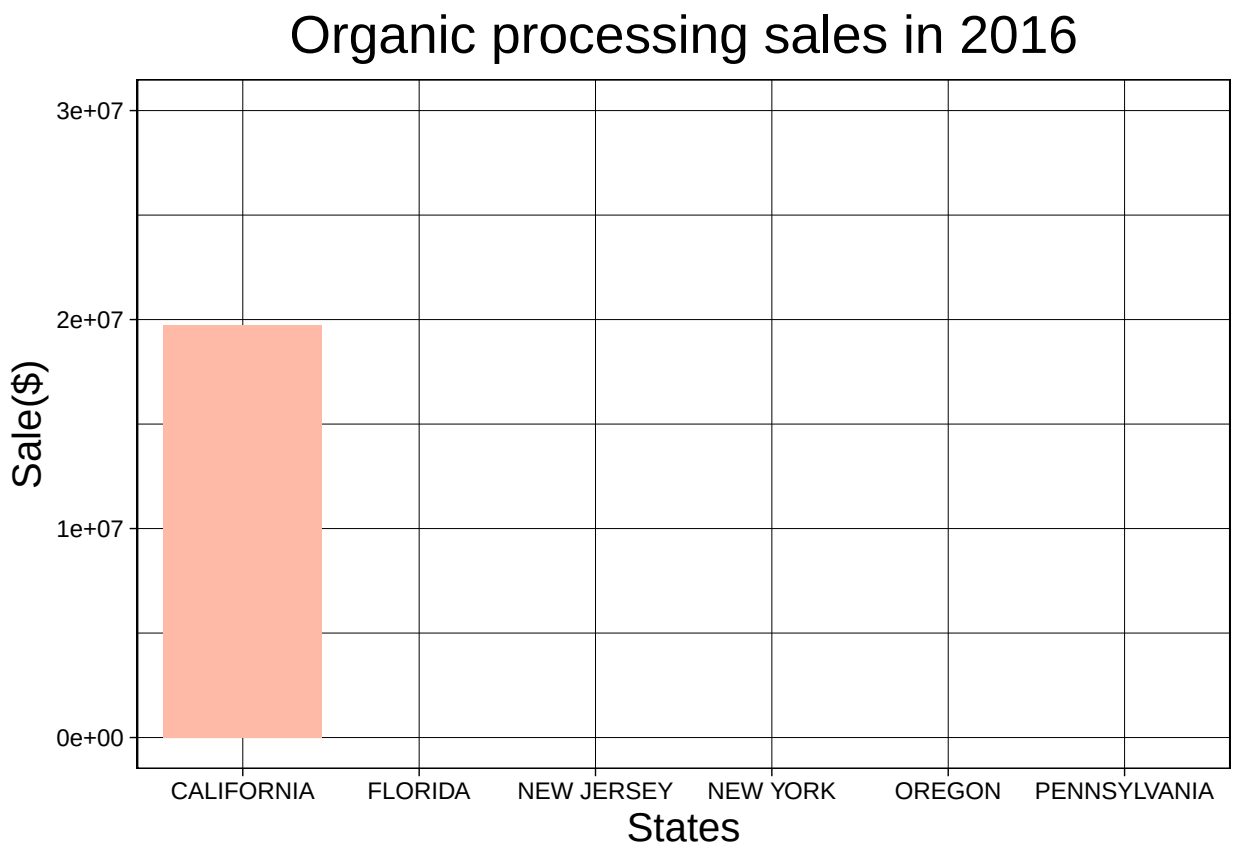

```
ggplot(data1_2019) +
  aes(x = State, y = `Value_FRESH MARKET - SALES`) +
  geom_col(fill = "#EFA954") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic sales of fresh market in 2019"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 320000000)
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



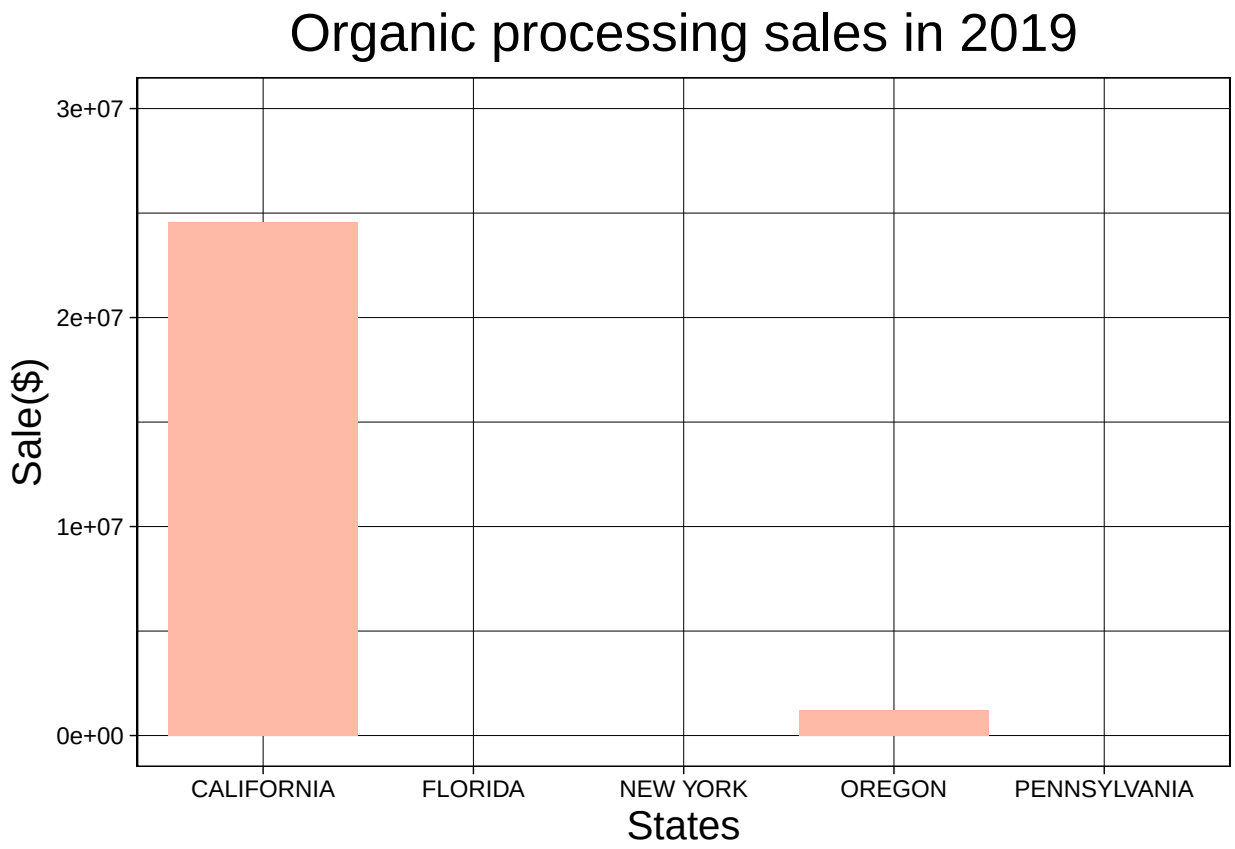

```
# plot processing organic sales VS states in 2016 and 2019
ggplot(data1_2016) +
  aes(x = State, y = `Value_PROCESSING - SALES`) +
  geom_col(fill = "#FFB9A7") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic processing sales in 2016"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 30000000)
```

```
## Warning: Removed 5 rows containing missing values (position_stack).
```



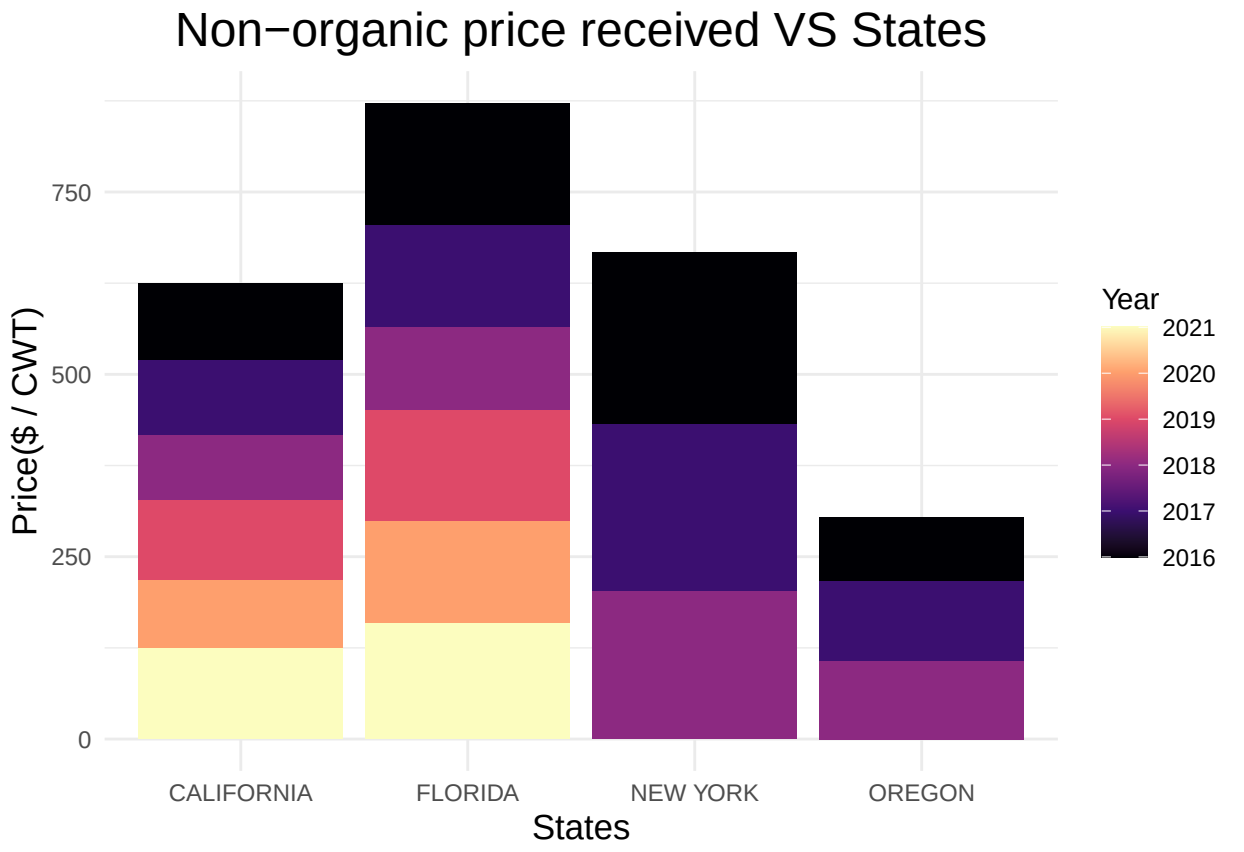

```
ggplot(data1_2019) +
  aes(x = State, y = `Value_PROCESSING - SALES`) +
  geom_col(fill = "#FFB9A7") +
  labs(
    x = "States",
    y = "Sale($)",
    title = "Organic processing sales in 2019"
  ) +
  theme_linedraw() +
  theme(
    plot.title = element_text(
      size = 20L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 15L),
    axis.title.x = element_text(size = 15L)
  ) +
  ylim(0, 30000000)
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```




```
# Then we explore the non_organic sale data
data2_price <- non_organic_sale %>%
  filter(Units == "$ / CWT") %>%
  select(!c(`State ANSI`, Units))

# First we plot the price receved in 2016-2021
ggplot(data2_price) +
  aes(
    x = State,
    y = `STRAWBERRIES - PRICE RECEIVED`,
    fill = Year
  ) +
  geom_col() +
  scale_fill_viridis_c(option = "magma", direction = 1) +
  labs(
    x = "States",
    y = "Price($ / CWT)",
    title = "Non-organic price received VS States"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(
      size = 18L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 13L),
```

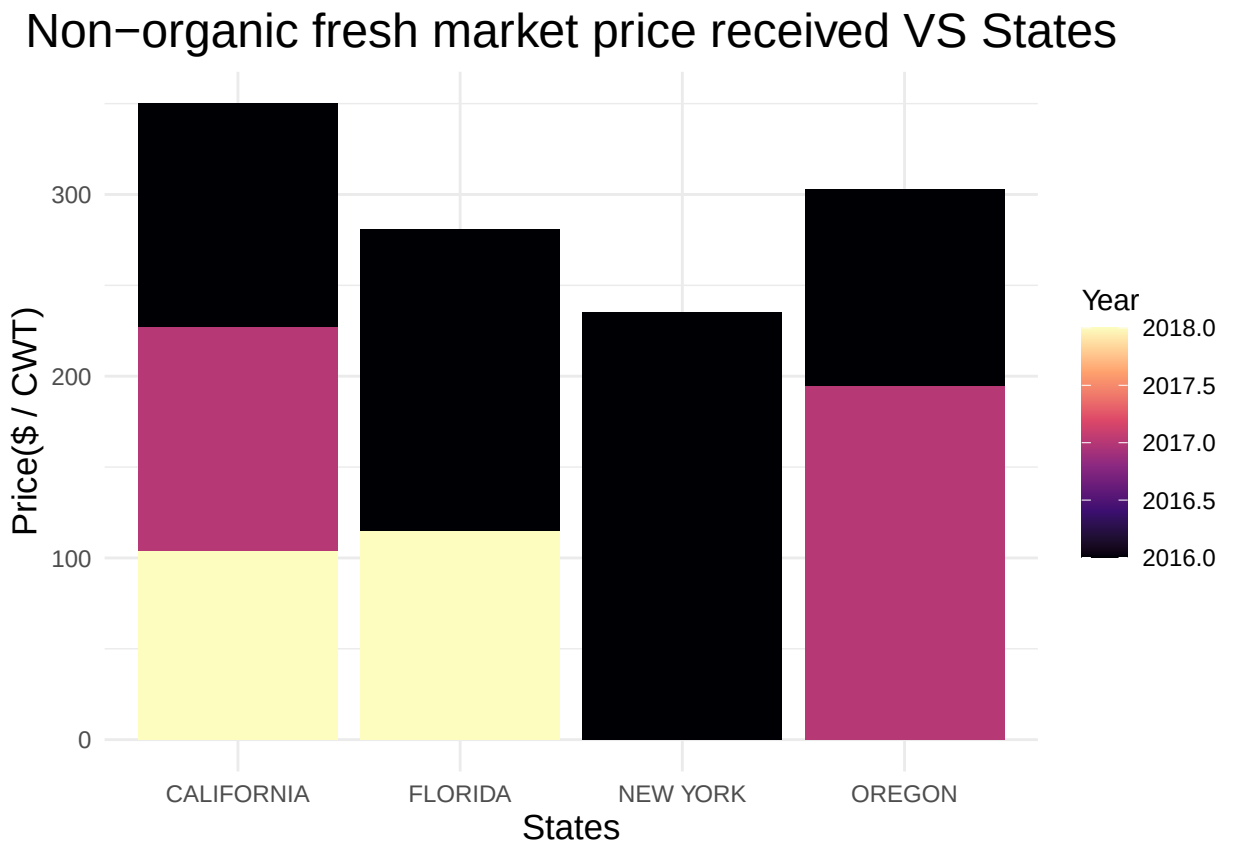
```
    axis.title.x = element_text(size = 13L)
  )
```




```
# then we can explore fresh market and processing

ggplot(data2_price) +
  aes(
    x = State,
    y = `FRESH MARKET - PRICE RECEIVED`,
    fill = Year
  ) +
  geom_col() +
  scale_fill_viridis_c(option = "magma", direction = 1) +
  labs(
    x = "States",
    y = "Price($ / CWT)",
    title = "Non-organic fresh market price received VS States"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(
      size = 18L,
      hjust = 0.5
    ),
    axis.title.y = element_text(size = 13L),
```

```
    axis.title.x = element_text(size = 13L)
  )
```

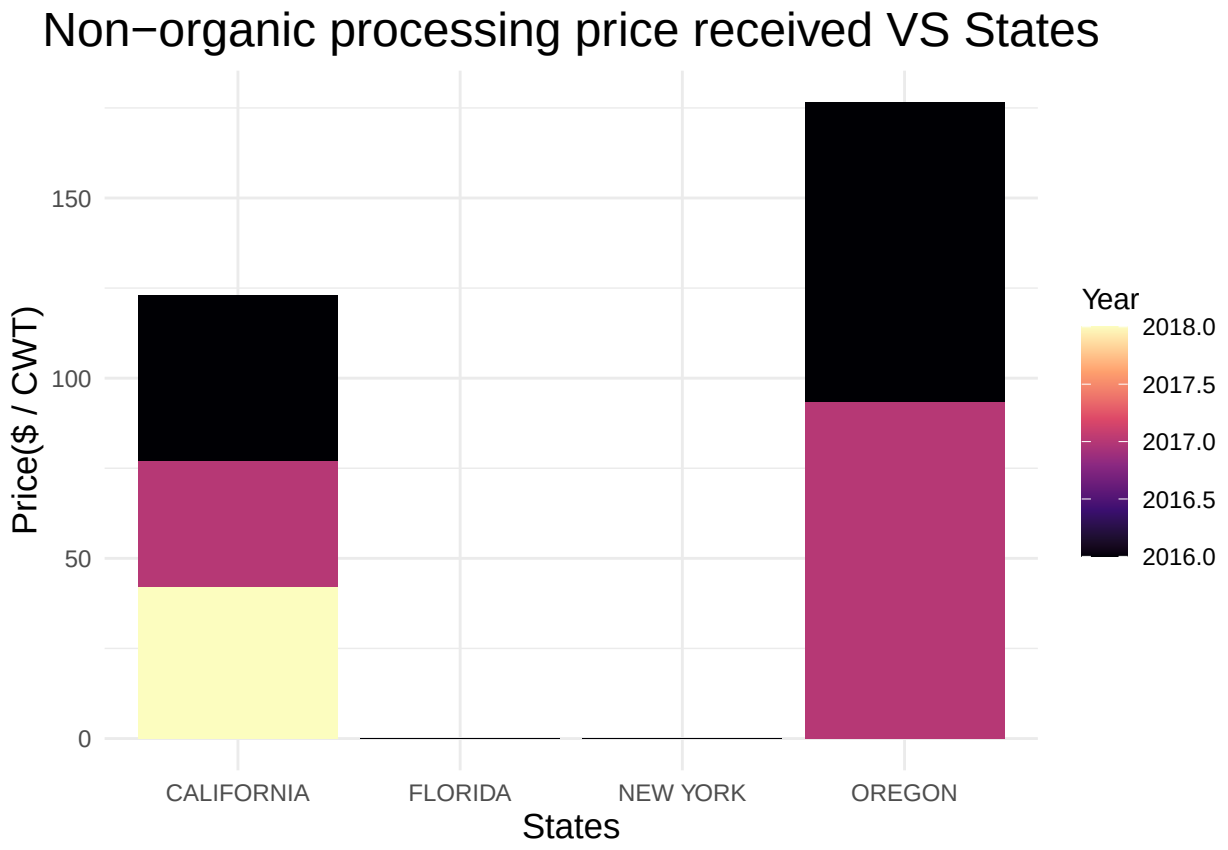
```
## Warning: Removed 10 rows containing missing values (position_stack).
```




```
ggplot(data2_price) +
  aes(
    x = State,
    y = `PROCESSING - PRICE RECEIVED`,
    fill = Year
  ) +
  geom_col() +
  scale_fill_viridis_c(option = "magma", direction = 1) +
  labs(
    x = "States",
    y = "Price($ / CWT)",
    title = "Non-organic processing price received VS States"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(
      size = 18L,
      hjust = 0.5
    ),
```

```
    axis.title.y = element_text(size = 13L),
    axis.title.x = element_text(size = 13L)
  )
```

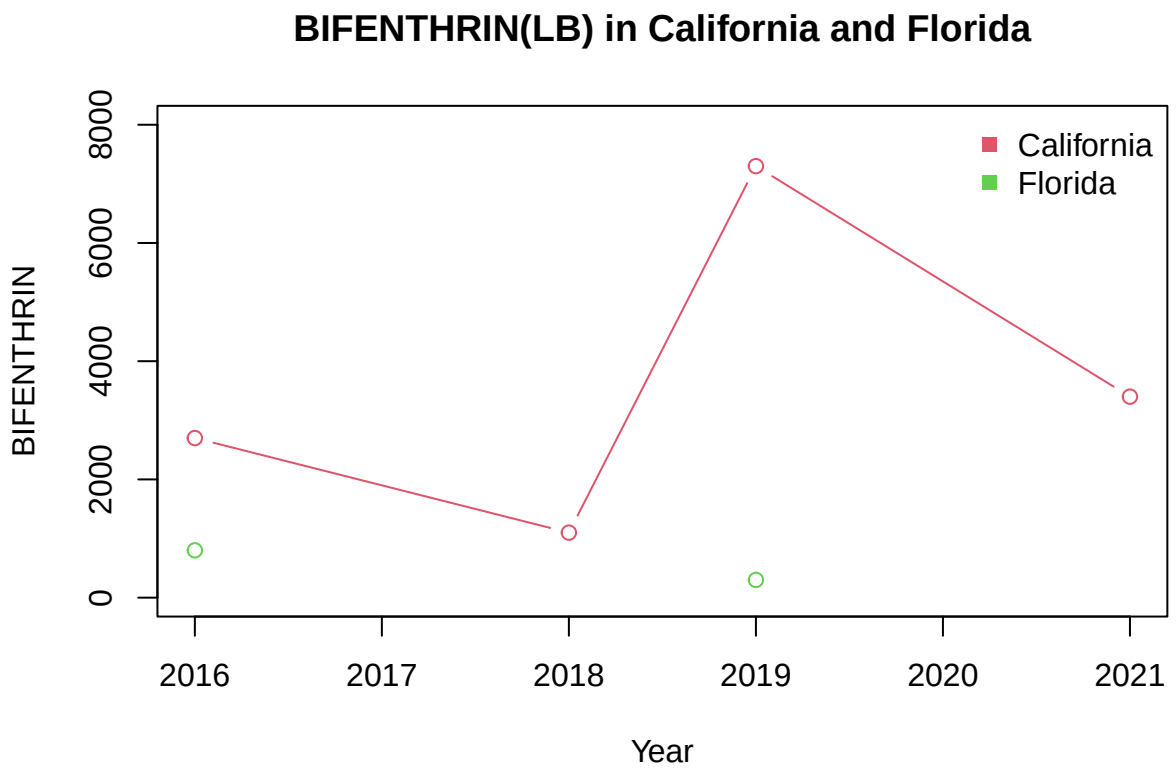
```
## Warning: Removed 10 rows containing missing values (position_stack).
```




```
# EDA about chemical data
# poison chemicals: BIFENTHRIN, CHLOROPICRIN, DICHLOROPROPENE
# plot Bifenthrin in different States and Year
data_BIFENTHRIN <- chem_data %>%
  filter(Name == "BIFENTHRIN")

data_BIFENTHRIN
```

```
## # A tibble: 6 x 11
##   State State~1 Type  Units Domain Name  Code   `2016`   `2018`  `2019`   `2021`
##   <chr>   <dbl> <chr> <chr> <chr>  <chr> <chr>   <dbl>    <dbl>   <dbl>    <dbl>
## 1 CALI~       6 LB    LB    INSEC~ BIFE~ 1288~ 2.7 e+3 1100     7.3 e+3 3400
## 2 CALI~       6 LB /~ AVG   INSEC~ BIFE~ 1288~ 1.07e-1    0.098 1.09e-1    0.123
## 3 CALI~       6 LB /~ AVG   INSEC~ BIFE~ 1288~ 1.63e-1    0.163 3.46e-1    0.191
## 4 FLOR~      12 LB    LB    INSEC~ BIFE~ 1288~ 8   e+2   NA     3   e+2   NA
## 5 FLOR~      12 LB /~ AVG   INSEC~ BIFE~ 1288~ 1.17e-1   NA     5   e-2   NA
## 6 FLOR~      12 LB /~ AVG   INSEC~ BIFE~ 1288~ 1.62e-1   NA     7.7 e-2   NA
## # ... with abbreviated variable name 1: `State ANSI`
```

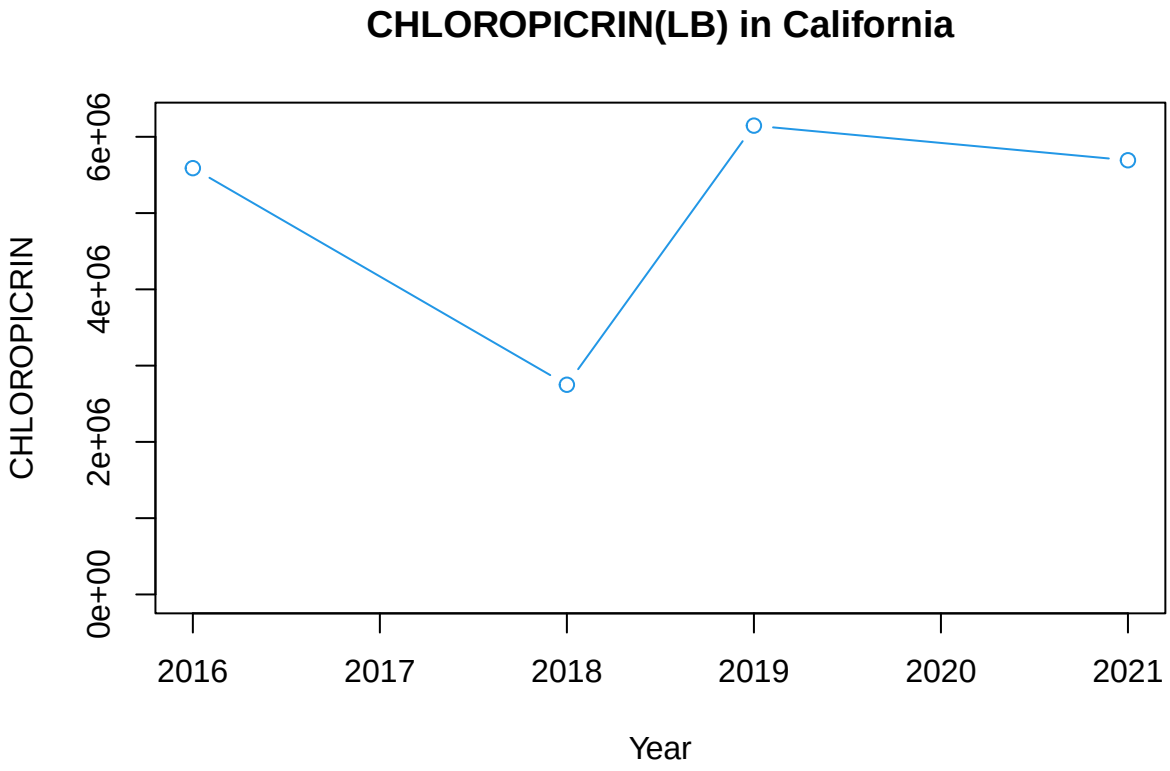
```r
# plot the BIFENTHRIN in LB units, and compared the values in California and Florida
data_BIFENTHRIN %<>% filter(Units == "LB") %>% select(!colnames(data_BIFENTHRIN[2:7]))

x <- colnames(data_BIFENTHRIN)[2:5]
y <- data_BIFENTHRIN[, 2:5]
plot(x, y[1, ], col = 2, type = "b", ylim = c(0, 8000), xlab = "Year", ylab = "BIFENTHRIN", main = "BIFE
lines(x, y[2, ], col = 3, type = "b")
legend("topright", pch = c(15, 15), legend = c("California", "Florida"), col = c(2, 3), bty = "n")
```




```r
# plot CHLOROPICRIN in different States and Year
data_CHLOROPICRIN <- chem_data %>%
  filter(Name == "CHLOROPICRIN")

data_CHLOROPICRIN
```

```
## # A tibble: 3 x 11
##   State      State ~1 Type  Units Domain Name  Code  `2016` `2018` `2019` `2021`
##   <chr>         <dbl> <chr> <chr> <chr>  <chr> <chr>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 CALIFORNIA        6 LB    LB    OTHER  CHLO~ 81501 5.59e6 2.75e6 6.15e6 5.69e6
## 2 CALIFORNIA        6 LB /~ AVG   OTHER  CHLO~ 81501 2.52e2 2.37e2 2.44e2 1.97e2
## 3 CALIFORNIA        6 LB /~ AVG   OTHER  CHLO~ 81501 2.81e2 2.57e2 3.46e2 2.21e2
## # ... with abbreviated variable name 1: `State ANSI`
```

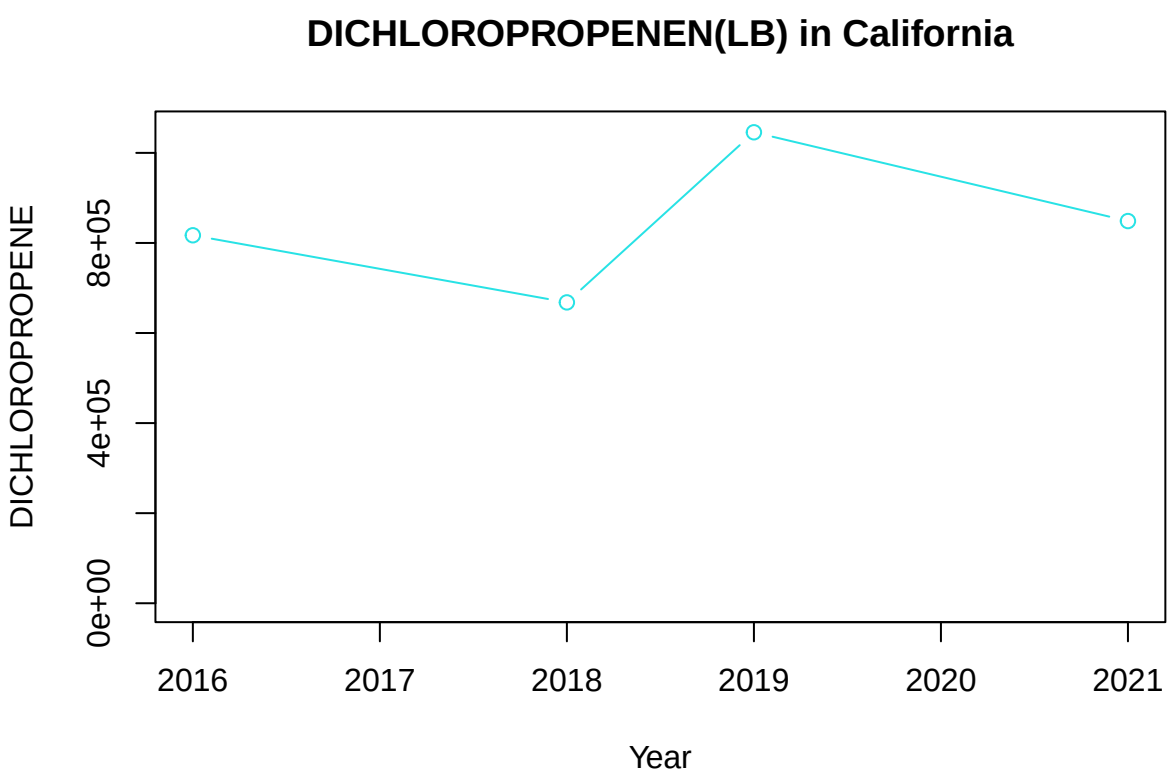
```
# plot the CHLOROPICRIN in LB units
data_CHLOROPICRIN %<>% filter(Units == "LB") %>% select(!colnames(data_CHLOROPICRIN[2:7]))
x <- colnames(data_CHLOROPICRIN)[2:5]
y <- data_CHLOROPICRIN[, 2:5]
plot(x, y[1, ], col = 4, type = "b", ylim = c(0, 6200000), xlab = "Year", ylab = "CHLOROPICRIN", main =
```




```
# plot DICHLOROPROPENE in different States and Year
data_DICHLOROPROPENE <- chem_data %>%
  filter(Name == "DICHLOROPROPENE")

data_DICHLOROPROPENE
```

```
## # A tibble: 3 x 11
##   State      State ~1 Type  Units Domain Name  Code  `2016` `2018` `2019` `2021`
##   <chr>         <dbl> <chr> <chr> <chr>  <chr> <chr>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 CALIFORNIA        6 LB    LB    OTHER  DICH~ 29001 8.17e5 6.68e5 1.05e6 8.49e5
## 2 CALIFORNIA        6 LB /~ AVG   OTHER  DICH~ 29001 1.04e2 8.32e1 8.36e1 7.07e1
## 3 CALIFORNIA        6 LB /~ AVG   OTHER  DICH~ 29001 1.12e2 8.52e1 1.26e2 7.75e1
## # ... with abbreviated variable name 1: `State ANSI`
```

```
# plot the DICHLOROPROPENE in LB units
data_DICHLOROPROPENE %<>% filter(Units == "LB") %>% select(!colnames(data_DICHLOROPROPENE[2:7]))
x <- colnames(data_DICHLOROPROPENE)[2:5]
y <- data_DICHLOROPROPENE[, 2:5]
plot(x, y[1, ], col = 5, type = "b", ylim = c(0, 1050000), xlab = "Year", ylab = "DICHLOROPROPENE", mai
```




```
# try to find some safe chemicals
grep("nitrogen", chem_data$Name, ignore.case = TRUE)
```

```
## [1] 262 268 274 279 283 287
```

```
grep("Phosphorous", chem_data$Name, ignore.case = TRUE)
```

```
## integer(0)
```

```
grep("Phosphate", chem_data$Name, ignore.case = TRUE)
```

```
## [1]  69 139 208 263 269 275 280 284 288
```

```
grep("Potassium", chem_data$Name, ignore.case = TRUE)
```

```
## [1]  18  70  91 140 160 209
```

```
# do analysis by using Nitrogen
row_nitrogen <- grep("nitrogen", chem_data$Name, ignore.case = TRUE)
data_NITROGEN <- chem_data[row_nitrogen, ]
data_NITROGEN %<>% filter(Units == "LB") %>% select(!colnames(data_NITROGEN[2:7]))
```

```r
# plot Nitrogen in different years and states
x <- colnames(data_NITROGEN)[2:5]
y <- data_NITROGEN[, 2:5]
plot(x, y[1, ], col = 4, type = "b", ylim = c(0, 10700000), xlab = "Year", ylab = "NITROGEN", main = "NI
lines(x, y[2, ], col = 2, type = "b")
legend("topright", pch = c(15, 15), legend = c("California", "Florida"), col = c(4, 2), bty = "n")
```