Strawberry

Yang Xiao

2023-10-16

```r
library(knitr)
library(kableExtra)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
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 becom
```

```r
library(stringr)
```

Read the file

```r
strawberry <- read_csv("strawberry.csv", col_names = TRUE)

# glimpse(strawberry)
```

Date cleaning

```
drop_one_value_col <- function(df){
  drop <- NULL
  for (i in 1:ncol(df)){
    unique_count <- n_distinct(df[, i])
    if (unique_count == 1){
      drop <- c(drop, i)
    }
  }

  if (length(drop) == 0) {
    print("No columns to drop.")
    return(df)
  } else {
    cat("Columns dropped:", colnames(df)[drop], "\n")
    strawberry <- df[, -drop]
    return(strawberry)
  }
}

## Use the function
str <- drop_one_value_col(strawberry)
```

```
Columns dropped: Week Ending Geo Level Ag District Ag District Code County County ANSI Zip C
```

```
str <- str$col_name
```

```
Warning: Unknown or uninitialised column: `col_name`.
```

```
strawberry <- strawberry |> select(!all_of(str))


vals=strawberry$Value
vals=sub(",","",vals)
vals=sub('""',"",vals)
vals=as.numeric(vals)
```

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

```r
strawberry["Value"]=vals

state_all <- strawberry |> group_by(State) |> count()

strawberry_census <- strawberry |> filter((Program=="CENSUS"))
 strawberry_census <- strawberry_census |>
  separate_wider_delim(  cols = `Data Item`,
                         delim = ",",
                         names = c("Fruit",
                                 "temp1",
                                 "temp2",
                                 "temp3"),
                         too_many = "error",
                         too_few = "align_start"
                       )

strawberry_census <- strawberry_census |>
  separate_wider_delim(  cols = temp1,
                         delim = " - ",
                         names = c("crop_type",
                                 "prop_acct"),
                         too_many = "error",
                         too_few = "align_start"
                       )



strawberry_census$crop_type <- str_trim(strawberry_census$crop_type, side = "both")

strawberry_census$temp2 <- str_trim(strawberry_census$temp2, side = "both")

strawberry_census$temp3 <- str_trim(strawberry_census$temp3, side = "both")



##Fresh Market
## make a copy of the temp2 column named `Fresh Market`.
strawberry_census <- strawberry_census |> mutate(`Fresh Market` = temp2, .after = temp2)

## Remove cells in `Fresh Market` column
##   that begin "MEASURED"
```

```r
strawberry_census$`Fresh Market` <- strawberry_census$`Fresh Market` |> str_replace( "^MEA

## Remove cells in `Fresh Market` column
##   that begin "PROCESSING"
strawberry_census$`Fresh Market` <- strawberry_census$`Fresh Market` |> str_replace( "^P.*

## substitute a space for NA in `Fresh Market` column
strawberry_census$`Fresh Market`[is.na(strawberry_census$`Fresh Market`)] <- ""

## in temp2 column, remove cells that begin "FRESH"
 strawberry_census$temp2 <- strawberry_census$temp2 |> str_replace("^F.*", "")

## Now fix the entries in the `Fresh Market` column
##   Remove "FRESH MARKET - " from the cells
strawberry_census$`Fresh Market` <- strawberry_census$`Fresh Market` |> str_replace("^FRES

## Create a "Process Market" column


# Make a copy of temp2 named `Process Market`
strawberry_census <- strawberry_census |> mutate(`Process Market` = temp2, .after = temp2)

# Remove `Process Market` cells beginning "MEASURED"
strawberry_census$`Process Market` <-  strawberry_census$`Process Market` |> str_replace("

# Substitute space for NA in `Process Market` column
strawberry_census$`Process Market`[is.na(strawberry_census$`Process Market`)] <- ""

# In temp2, remove cells that begin "PROCESSING"
strawberry_census$temp2 <- strawberry_census$temp2 |> str_replace("^P.*", "")

# In `Process Market`, remove "PROCESSING - " from cells
strawberry_census$`Process Market` <-  strawberry_census$`Process Market` |> str_replace("

## substitute a space for NA in prop_acct column
strawberry_census$prop_acct[is.na(strawberry_census$prop_acct)] <- ""

## substitute a space for NA in temp2 column
strawberry_census$temp2[is.na(strawberry_census$temp2)] <- ""

## substitute a space for NA in temp2 column
```

```r
strawberry_census$temp3[is.na(strawberry_census$temp3)] <- ""



# Combine temp2 and temp3 columns into Metric
strawberry_census <- strawberry_census |> unite(temp2, temp3, col = "Metric", sep = "")

# Remove "MEASURED IN " from the cells in the Metric column
strawberry_census$Metric <- strawberry_census$Metric |> str_replace("MEASURED IN ", "")

# Move Metric to the end
strawberry_census <- strawberry_census |> relocate(Metric, .before = Domain)

strawberry_census <- strawberry_census |> relocate(`Process Market`, .before = Metric)

strawberry_census <- strawberry_census |> rename(Totals = prop_acct)
```

CENSUS initial question

Which continent has the highest number of rows (n)? And the ten continents with the highest average value? (Counted as Operations With SALES, CWT, $respectively)

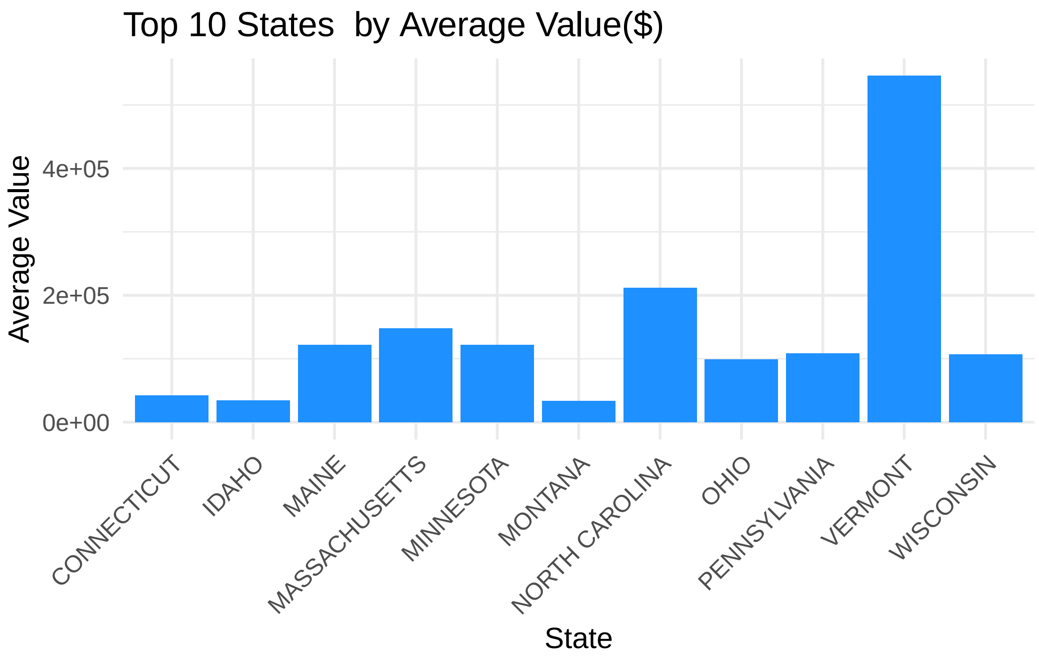
CENSUS EDA and solution

(a)The highest number of rows (n)

```r
##EDA

#CENSUS
## Which state has the most rows($)

strawberry_census_dollar <- strawberry_census |>
  filter(!is.na(Value) & (Metric == "$"))


top_10_states_dollar <- strawberry_census_dollar |>
  group_by(State) |>
  summarise(avg_value = mean(Value), n = n())|>
  arrange(desc(n)) |>
  top_n(10)
```

```
Selecting by n
```

```
library(ggplot2)

ggplot(top_10_states_dollar, aes(x = State, y = avg_value)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States  by Average Value($)",
       x = "State",
       y = "Average Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
print(top_10_states_dollar)
```

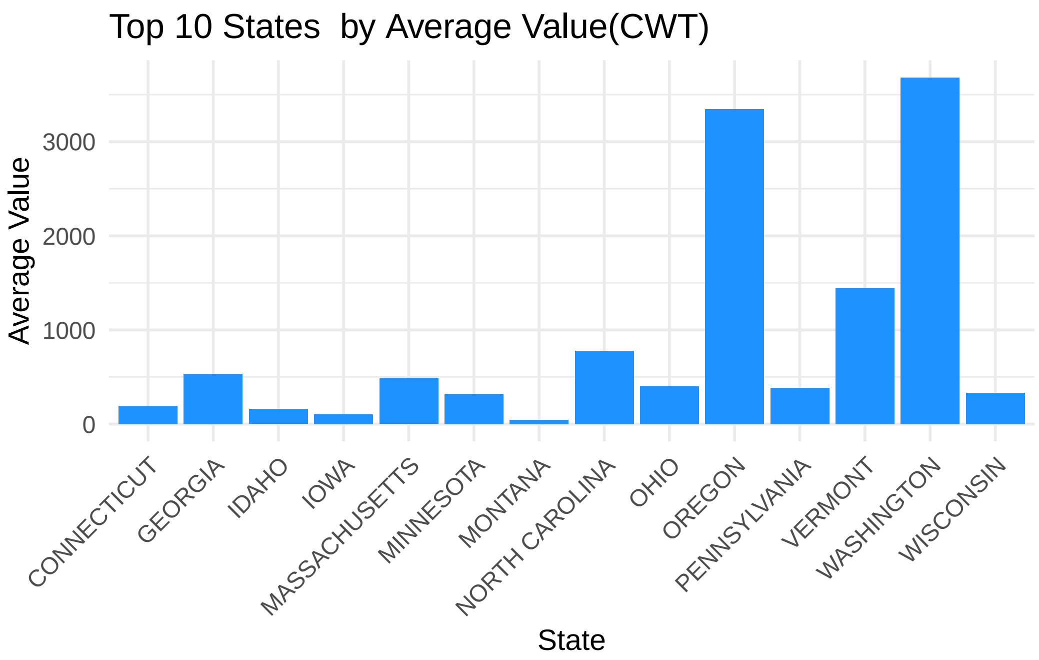
```
# A tibble: 11 x 3
   State          avg_value     n
   <chr>              <dbl> <int>
```

```
 1 CONNECTICUT       42065      6
 2 IDAHO             33943.     6
 3 MASSACHUSETTS    147951.     6
 4 MONTANA           33323.     6
 5 NORTH CAROLINA   211963      6
 6 OHIO              99064      6
 7 PENNSYLVANIA     108495      6
 8 VERMONT          546020      6
 9 WISCONSIN        106694.     6
10 MAINE            121537.     5
11 MINNESOTA        122181.     5
```

```
## Which state has the most rows(CWT)

strawberry_census_CWT <- strawberry_census |>
  filter(!is.na(Value) & (Metric == "CWT"))


top_10_states_CWT <- strawberry_census_CWT |>
  group_by(State) |>
  summarise(avg_value = mean(Value), n = n())|>
  arrange(desc(n)) |>
  top_n(10)
```

```
Selecting by n
```

```
library(ggplot2)

ggplot(top_10_states_CWT, aes(x = State, y = avg_value)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States  by Average Value(CWT)",
       x = "State",
       y = "Average Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



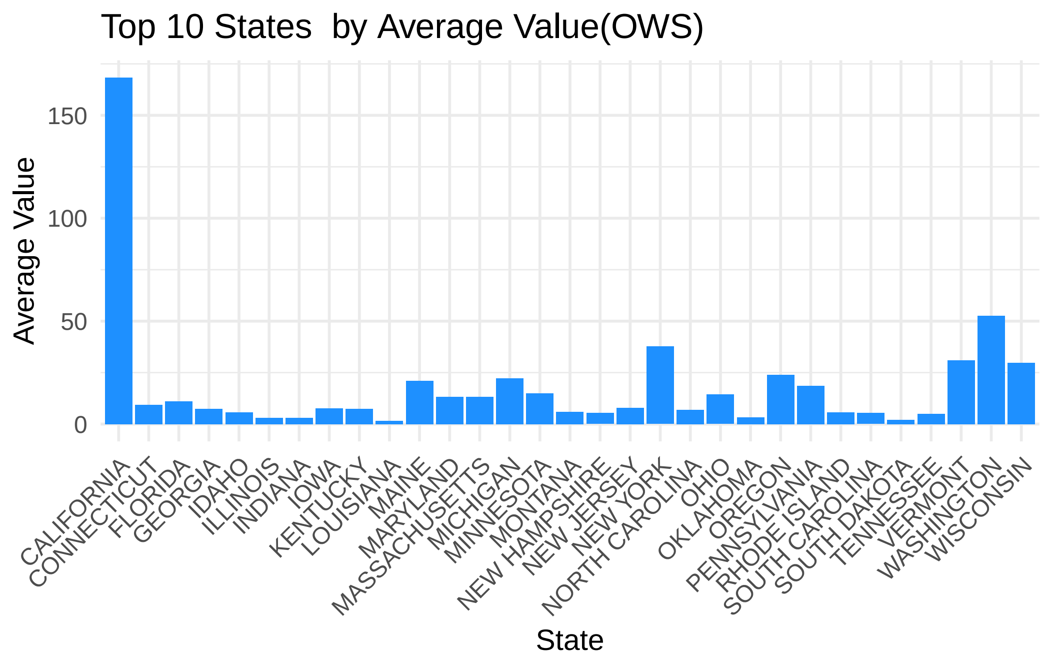
```r
print(top_10_states_CWT)
```

```
# A tibble: 14 x 3
   State          avg_value     n
   <chr>              <dbl> <int>
 1 WASHINGTON         3681.    12
 2 OREGON             3348.    11
 3 MINNESOTA           323.    10
 4 CONNECTICUT         189.     9
 5 GEORGIA             536.     9
 6 IDAHO               159.     9
 7 IOWA                102.     9
 8 MASSACHUSETTS       484.     9
 9 MONTANA              46      9
10 NORTH CAROLINA      781.     9
11 OHIO                403      9
12 PENNSYLVANIA        383.     9
13 VERMONT            1442      9
14 WISCONSIN           334.     9
```

```
## Which state has the most rows(OWS)
strawberry_census_OWS <- strawberry_census |>
  filter(!is.na(Value)) |>
           filter(Totals == "OPERATIONS WITH SALES"|'Fresh Market'=="OPERATIONS WITH SALES


top_10_states_OWS <- strawberry_census_OWS |>
  group_by(State) |>
  summarise(avg_value = mean(Value), n = n())|>
  arrange(desc(n)) |>
  top_n(10)
```

```
Selecting by n
```

```
library(ggplot2)

ggplot(top_10_states_OWS, aes(x = State, y = avg_value)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States  by Average Value(OWS)",
       x = "State",
       y = "Average Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

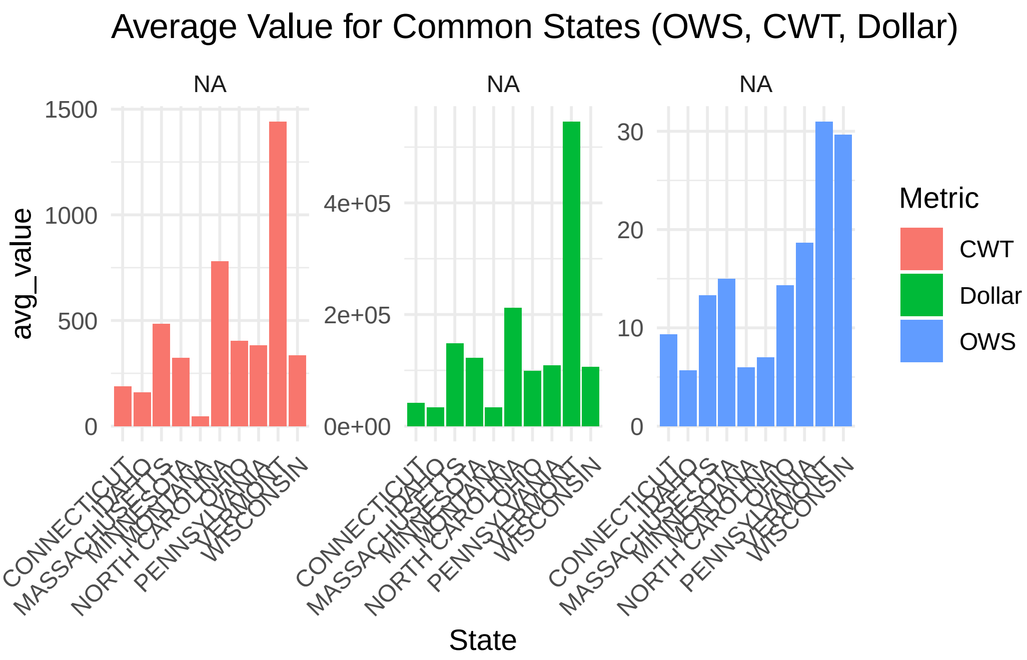


```
print(top_10_states_OWS)
```

```
# A tibble: 31 x 3
   State        avg_value     n
   <chr>            <dbl> <int>
 1 CALIFORNIA      168.       3
 2 CONNECTICUT       9.33     3
 3 FLORIDA          11        3
 4 GEORGIA           7.33     3
 5 IDAHO             5.67     3
 6 ILLINOIS          3        3
 7 INDIANA           3        3
 8 IOWA              7.67     3
 9 KENTUCKY          7.33     3
10 LOUISIANA         1.67     3
# i 21 more rows
```

```
###
# Create data frames for each metric (OWS, CWT, Dollar)
 df_ows <- data.frame(State = top_10_states_OWS$State, Metric = "OWS", avg_value = top_10_
 df_cwt <- data.frame(State = top_10_states_CWT$State, Metric = "CWT", avg_value = top_10_
```

```r
 df_dollar <- data.frame(State = top_10_states_dollar$State, Metric = "Dollar", avg_value

# Combine the data frames
 common_states_data <- rbind(df_ows, df_cwt, df_dollar)

# Find the states that are common among top_10_states_OWS, top_10_states_dollar, and top_1
 common_states <- intersect(top_10_states_OWS$State, intersect(top_10_states_dollar$State,
##Select common state
selected_states <- c("CONNECTICUT", "IDAHO", "MASSACHUSETTS", "MINNESOTA", "MONTANA", "NOR

 common_states_data <- common_states_data %>%
   filter(State %in% selected_states)


 # Create a data frame that includes a numeric label for each state
common_states_data <- common_states_data |>
  mutate(StateLabel = factor(State, levels = common_states))

# Create a vector to store the units for each metric
unit_labels <- c("Unit for OWS", "Unit for CWT", "Unit for Dollar")

# Create a ggplot with facets for each metric
gg <- ggplot(common_states_data, aes(x = State, y = avg_value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Average Value for Common States (OWS, CWT, Dollar)",
    x = "State"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1) )

# Add different y-axis labels for each facet
gg <- gg + facet_wrap(~ Metric, scales = "free_y", labeller = labeller(Metric = unit_label

print(gg)
```



```
cat("The cities with the top 10 ave_sales are($):",top_10_states_dollar$State, "\n")
```

```
The cities with the top 10 ave_sales are($): CONNECTICUT IDAHO MASSACHUSETTS MONTANA NORTH CA
```

```
cat("The cities with the top 10 ave_sales are(CWT):",top_10_states_CWT$State, "\n")
```

```
The cities with the top 10 ave_sales are(CWT): WASHINGTON OREGON MINNESOTA CONNECTICUT GEORGI
```

```
cat("The cities with the top 10 ave_sales are(OWS):",top_10_states_OWS$State, "\n")
```

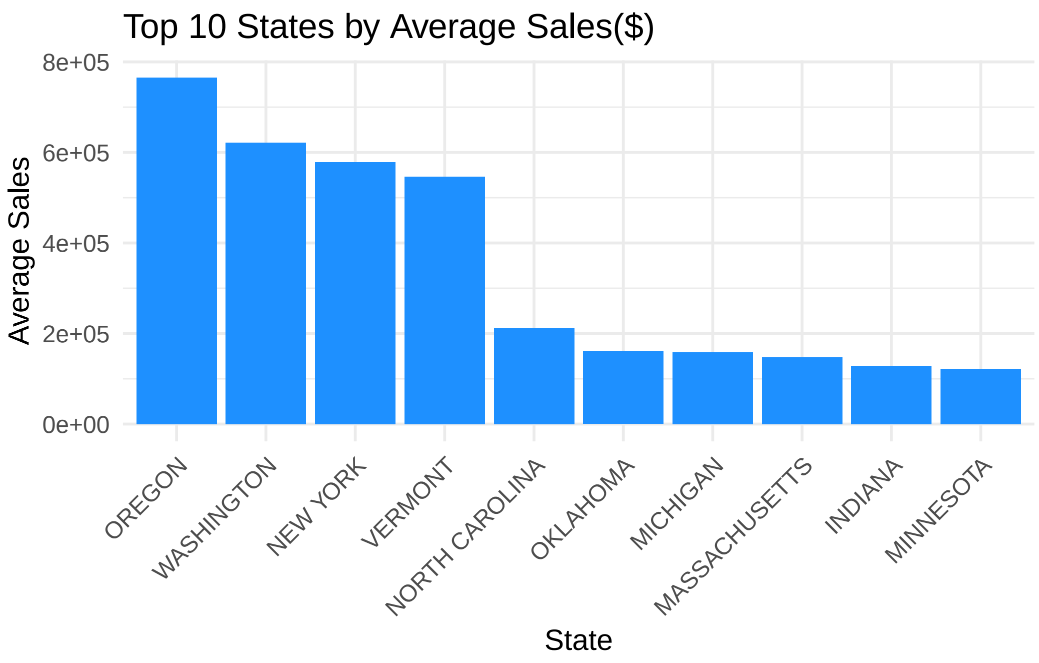
```
The cities with the top 10 ave_sales are(OWS): CALIFORNIA CONNECTICUT FLORIDA GEORGIA IDAHO I
```

```
cat("The cities with the highest overall sales are:",common_states, "\n")
```

```
The cities with the highest overall sales are: CONNECTICUT IDAHO MASSACHUSETTS MINNESOTA MONT
```

(b) the highest average value

```
##Average value rank for CENSUS


##For dollar
strawberry_census_dollar <- strawberry_census |>
  filter(!is.na(Value) & (Metric == "$"))

state_avg_sales_dollar <- strawberry_census_dollar %>%
  group_by(State) %>%
  summarise(avg_sales = mean(Value)) %>%
  top_n(10, wt = avg_sales)




library(ggplot2)

ggplot(state_avg_sales_dollar, aes(x = reorder(State, -avg_sales), y = avg_sales)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States by Average Sales($)",
       x = "State",
       y = "Average Sales") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

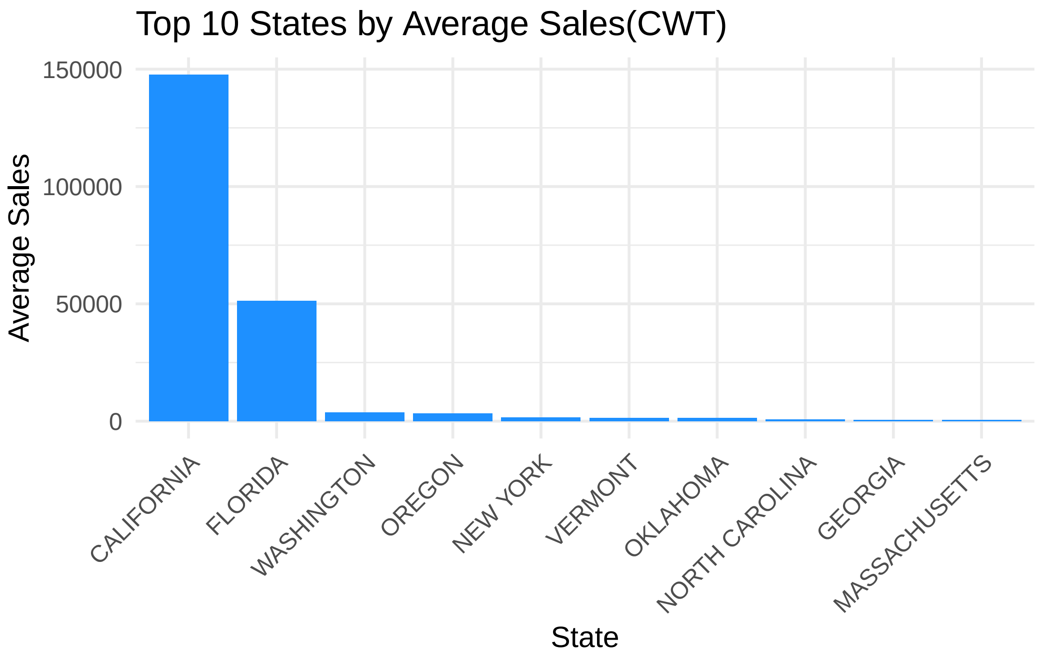


```
print(state_avg_sales_dollar$State)
```

```
[1] "INDIANA"        "MASSACHUSETTS"  "MICHIGAN"       "MINNESOTA"
[5] "NEW YORK"       "NORTH CAROLINA" "OKLAHOMA"       "OREGON"
[9] "VERMONT"        "WASHINGTON"
```

```
###For CWT


strawberry_census_CWT <- strawberry_census |>
  filter(!is.na(Value) & (Metric == "CWT"))


state_avg_sales_CWT <- strawberry_census_CWT %>%
  group_by(State) %>%
  summarise(avg_sales = mean(Value)) %>%
  top_n(10, wt = avg_sales)
```

```r
library(ggplot2)

ggplot(state_avg_sales_CWT, aes(x = reorder(State, -avg_sales), y = avg_sales)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States by Average Sales(CWT)",
       x = "State",
       y = "Average Sales") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

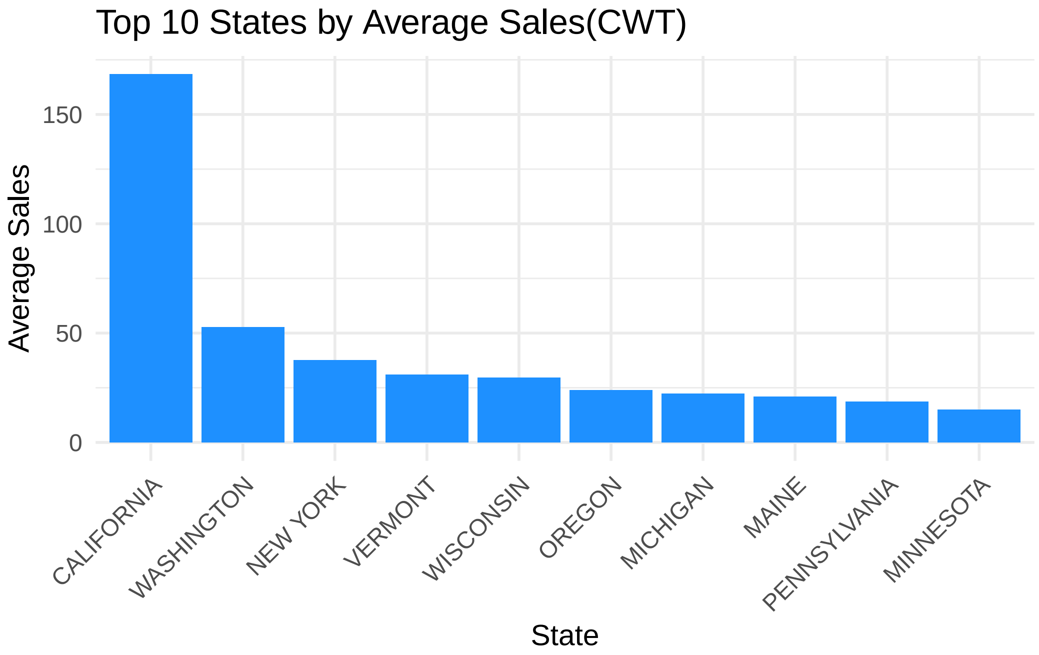


```r
print(state_avg_sales_CWT$State)
```

```
[1] "CALIFORNIA"     "FLORIDA"        "GEORGIA"        "MASSACHUSETTS"
[5] "NEW YORK"       "NORTH CAROLINA" "OKLAHOMA"       "OREGON"
[9] "VERMONT"        "WASHINGTON"
```

```r
##For OWS

strawberry_census_OWS <- strawberry_census |>
  filter(!is.na(Value)) |>
```

```
           filter(Totals == "OPERATIONS WITH SALES"|'Fresh Market'=="OPERATIONS WITH SALES


state_avg_sales_OWS <- strawberry_census_OWS %>%
  group_by(State) %>%
  summarise(avg_sales = mean(Value)) %>%
  top_n(10, wt = avg_sales)




library(ggplot2)

ggplot(state_avg_sales_OWS, aes(x = reorder(State, -avg_sales), y = avg_sales)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  labs(title = "Top 10 States by Average Sales(CWT)",
       x = "State",
       y = "Average Sales") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



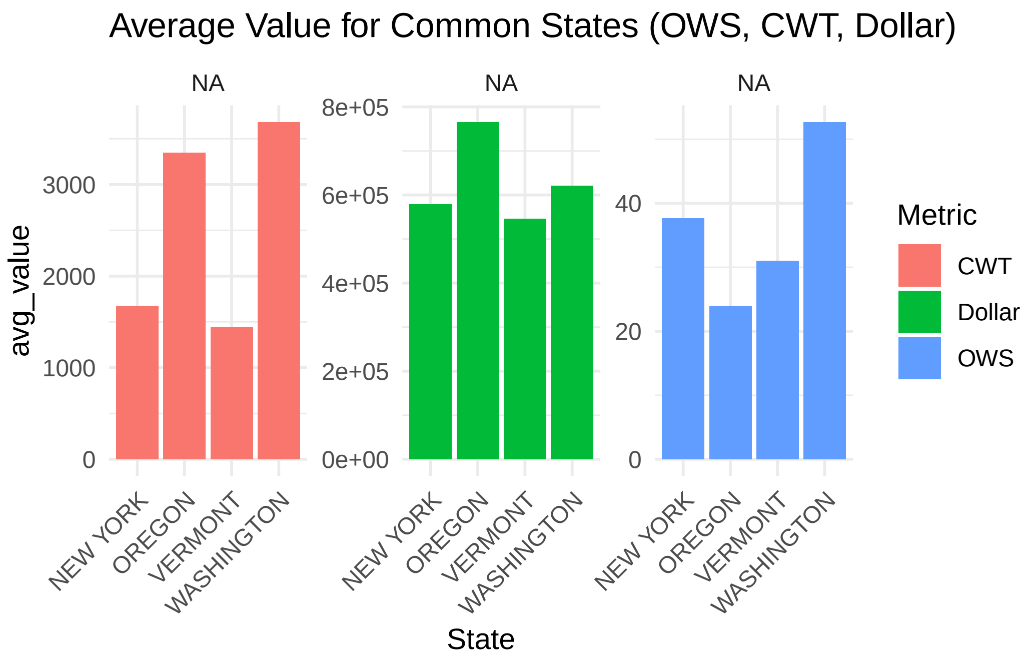
```
print(state_avg_sales_OWS$State)
```

```
 [1] "CALIFORNIA"   "MAINE"        "MICHIGAN"     "MINNESOTA"    "NEW YORK"
 [6] "OREGON"       "PENNSYLVANIA" "VERMONT"      "WASHINGTON"   "WISCONSIN"
```

```
###
# Create data frames for each metric (OWS, CWT, Dollar)
 df_ows <- data.frame(State = state_avg_sales_OWS$State, Metric = "OWS", avg_value = state
 df_cwt <- data.frame(State = state_avg_sales_CWT$State, Metric = "CWT", avg_value = state
 df_dollar <- data.frame(State = state_avg_sales_dollar$State, Metric = "Dollar", avg_valu

# Combine the data frames
 common_states_data <- rbind(df_ows, df_cwt, df_dollar)

# Find the states that are common among top_10_states_OWS, top_10_states_dollar, and top_1
 common_states <- intersect(state_avg_sales_OWS$State, intersect(state_avg_sales_dollar$St
  print(common_states)
```

```
[1] "NEW YORK"   "OREGON"     "VERMONT"    "WASHINGTON"
```

```
##Select common state
selected_states <- c("NEW YORK","OREGON","VERMONT","WASHINGTON")

 common_states_data <- common_states_data %>%
   filter(State %in% selected_states)


 # Create a data frame that includes a numeric label for each state
common_states_data <- common_states_data |>
  mutate(StateLabel = factor(State, levels = common_states))

# Create a vector to store the units for each metric
unit_labels <- c("Unit for OWS", "Unit for CWT", "Unit for Dollar")

# Create a ggplot with facets for each metric
gg <- ggplot(common_states_data, aes(x = State, y = avg_value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Average Value for Common States (OWS, CWT, Dollar)",
    x = "State"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1) )

# Add different y-axis labels for each facet
gg <- gg + facet_wrap(~ Metric, scales = "free_y", labeller = labeller(Metric = unit_label

print(gg)
```



```
cat("The cities with the top 10 ave_sales are($):",state_avg_sales_dollar$State, "\n")
```

```
The cities with the top 10 ave_sales are($): INDIANA MASSACHUSETTS MICHIGAN MINNESOTA NEW YOI
```

```
cat("The cities with the top 10 ave_sales are(CWT):",state_avg_sales_CWT$State, "\n")
```

```
The cities with the top 10 ave_sales are(CWT): CALIFORNIA FLORIDA GEORGIA MASSACHUSETTS NEW Y
```

```
cat("The cities with the top 10 ave_sales are(OWS):",state_avg_sales_OWS$State, "\n")
```

```
The cities with the top 10 ave_sales are(OWS): CALIFORNIA MAINE MICHIGAN MINNESOTA NEW YORK O
```

```r
cat("The cities with the highest overall sales are:",common_states, "\n")
```

```
The cities with the highest overall sales are: NEW YORK OREGON VERMONT WASHINGTON
```

SURVEY initial question

How to convert the chemical code to CAS and further determine the corresponding toxicity? What is the frequency of each toxicity?

EDA and Solution

```r
strwb_survey<- strawberry |> filter((Program=="SURVEY"))
stb_survey <- strwb_survey %>%
  filter(str_detect(`Data Item`, "MEASURED IN")) %>%
  mutate(`Data Item` = str_extract(`Data Item`, "(?<=MEASURED IN ).*"))
stb_survey <- stb_survey %>%
  mutate(
    Chemical = if_else(str_detect(`Domain Category`, "\\(.*=.*\\)"),
                       str_extract(`Domain Category`, "(?<=\\().*?(?=\\=)"),
                       NA_character_),
    Chemical_Code = if_else(str_detect(`Domain Category`, "\\(.*=.*\\)"),
                            str_extract(`Domain Category`, "(?<=\\=).*?(?=\\))"),
                            NA_character_)
  )


stb_survey <- subset(stb_survey, select = -Program)
stb_survey <- subset(stb_survey, select = -`Domain Category`)
```

Dealing with Missing Values, Outliers, and Duplicates

```r
stb_survey <- stb_survey[, !sapply(stb_survey, function(col) all(is.na(col)))]


stb_survey <- stb_survey[!is.na(stb_survey$Value), ]
stb_survey <- stb_survey[stb_survey$State != "OTHER STATES", ]

strawberry_survey_chemical <- stb_survey  |>
  filter(!is.na(Chemical_Code))
```

Transfer the chemical code

```r
# Load the required packages
library(jsonlite)
```

```

Attaching package: 'jsonlite'

The following object is masked from 'package:purrr':

    flatten
```

```r
library(httr)
library(future)
library(furrr)

# function that can translate PC to CAS
get_cas <- function(PC){
    PC <- sprintf("%06d", as.numeric(PC))
    path <- paste0("https://ordspub.epa.gov/ords/pesticides/apprilapi/?q=%7b%22ais%22:%7b%
    r <- GET(url = path)
    r_text <- content(r, as = "text", encoding = "UTF-8")
    df <- fromJSON(r_text, flatten = TRUE)
    df_strwb <- df$items[grepl("Strawberries", df$items$sites, fixed=T),]
    ais <- df_strwb$ais[1]
    pattern <- "\\(([^A-Za-z]+)\\/([0-9-]+)\\)"
    text <- ais
    matches <- regmatches(text, gregexpr(pattern, text))
    cas <- sapply(matches, function(x) gsub(".*\\/([0-9-]+)\\)", "\\1", x))
    if (is.character(cas)) {
        return(cas[1])
}
    else {
        return("can't find")
}
}

# Create a PC t0 CAS form for the survey data
PC_form <- data.frame(
    PC = unique(strawberry_survey_chemical$Chemical_Code)[-1]
)
n = length(PC_form$PC)
CAS <- rep(NA,n)
for (i in 1:n){
    CAS[i] <- get_cas(PC_form$PC[i])
    print(i)
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
[1] 21
[1] 22
[1] 23
[1] 24
[1] 25
[1] 26
[1] 27
[1] 28
```

```
[1] 29
[1] 30
[1] 31
[1] 32
[1] 33
[1] 34
[1] 35
[1] 36
[1] 37
[1] 38
[1] 39
[1] 40
[1] 41
[1] 42
[1] 43
[1] 44
[1] 45
[1] 46
[1] 47
[1] 48
[1] 49
[1] 50
[1] 51
[1] 52
[1] 53
[1] 54
[1] 55
[1] 56
[1] 57
[1] 58
[1] 59
[1] 60
[1] 61
[1] 62
[1] 63
[1] 64
[1] 65
[1] 66
[1] 67
[1] 68
[1] 69
[1] 70
[1] 71
```

```
[1] 72
[1] 73
[1] 74
[1] 75
[1] 76
[1] 77
[1] 78
[1] 79
[1] 80
[1] 81
[1] 82
```

```
PC_form$CAS <- CAS




merged_data_cas <- merge(strawberry_survey_chemical, PC_form, by.x = "Chemical_Code", by.y

toxic <- read_csv("CAS.csv", col_names = TRUE)
```

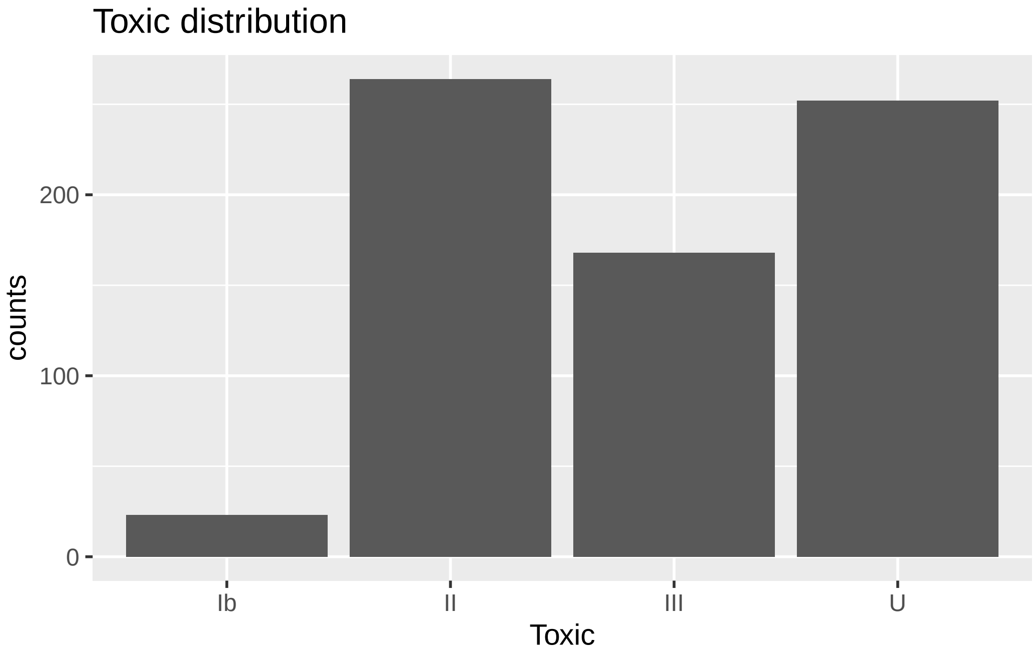
```
Rows: 1044 Columns: 2

-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): CAS, Toxic

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.
```

```
 merged_data_toxic<- merge(merged_data_cas, toxic, by.x = "CAS", by.y = "CAS", all.x = TRU


merged_data_toxic<-merged_data_toxic|>
  filter(!is.na(Toxic))

length(merged_data_toxic$Toxic)
```

```
[1] 707
```

frequency of each toxicity

```
library(ggplot2)

toxic_counts <- merged_data_toxic %>%
  group_by(Toxic) %>%
  summarize(n = n())


ggplot(data = toxic_counts, aes(x = Toxic, y = n)) +
  geom_bar(stat = "identity") +
  labs(title = "Toxic distribution", x = "Toxic", y = "counts")
```



Conclusion

The final table is merged_ Data_ Toxic has corresponding chemical codes, cas, and toxic, and corresponding information. However, some data did not provide you with chemical codes, so only 707 data were obtained.