Strawberries 1

Paul Moon

10/7/2024

```
#<READ>
#A lot of the code in this homework was used from USDA-NASS data cleaning-ver2.qmd that was given to us

#The reason for this is because I thought that the qmd contained information of the solutions of this h

#THEN. After I finished reviewing the qmd and learning the contents within it. I worked on splitting th

#The strawberries25_v3.csv file in my Github is the new and improved dataset.
```

```
#This is to read the data that we were supposed to use.
strawberry <- read_csv("strawberries25_v3.csv", col_names = TRUE)
```

```
## Rows: 12669 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (18): 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.
```

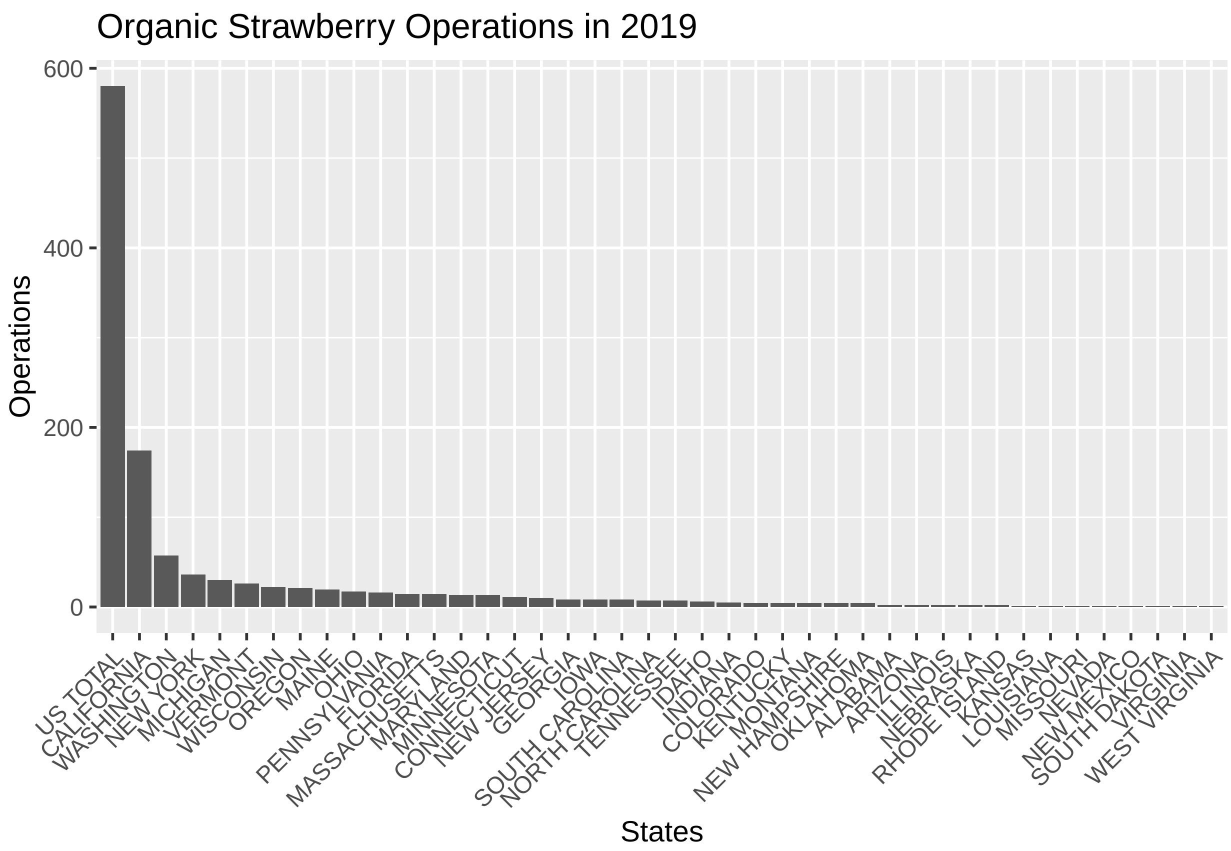
```
#We first look at the entire data set and then test to see whether or not the column has a single varia
oneCol <- function(df){
  drop <- NULL
  for(i in 1:dim(df)[2]){
    if((df |> distinct(df[, i]) |> count()) == 1){
      drop = c(drop, i)
     }
  }

#We write the outputs of the dropped columns
if(is.null(drop)){
  return("None")
  }
else{
  #I decided to take out the outputs since it was messy and unneeded.
  strawberry <- df[, -1 * drop]
  }
}
```

```r
#Here we get the new columns that we worked with and input that into our data set.
strawberry <- oneCol(strawberry)
```

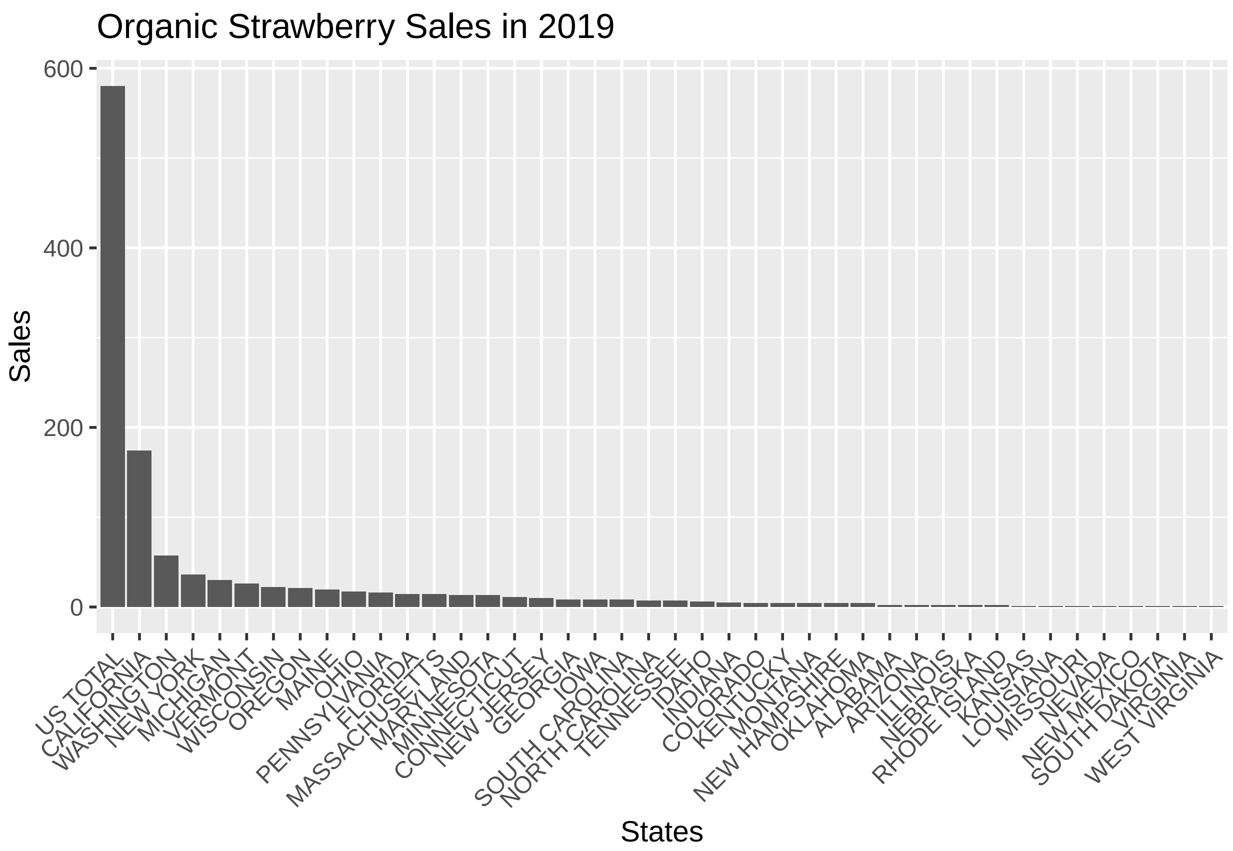
```r
#Recollecting the data to use in our separate_wider_delim
strawberry <- strawberry |>

#Here, we separate the columns by category and make the data set "wider" by the listed names
separate_wider_delim(cols = `Data Item`, delim = ",",
                     names = c("Fruit", "Category", "Item", "Metric"),
                     too_many = "error", too_few = "align_start")
#Did not understand this format of separate_wider_delim with the too_many and too_few until now.

#This is to change the old categories into the new trimmed category that deleted any extra 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")

sales <- strawberry |> filter(Program == "CENSUS")
chem <- strawberry |> filter(Program == "SURVEY")
nrow(strawberry) == (nrow(chem) + nrow(sales))
```

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

```r
#Plotting some data.
straw1 <- strawberry |>
#Setting up pulling out our data from the data set
  select(c(Year, State, Category, Value)) |>
#For my example, I wanted to use the year 2019.
  filter((Year == 2019) & (Category == "ORGANIC - OPERATIONS WITH SALES"))

straw1$Value <- as.numeric(straw1$Value)
straw1 <- straw1 |> arrange(desc(Value))
#This reorder allows us to make the graph cleaner because it orders them in size order.
ggplot(straw1, aes(reorder(State, -Value), Value)) +
  geom_bar(stat = "identity") +
#This is used to make the axis more readable
  theme(axis.text.x = element_text(angle = 45,hjust = 1)) +
  labs(x = "States", y = "Operations",
title ="Organic Strawberry Operations in 2019")
```



```
straw2 <- strawberry |>
#Again, setting up pulling out our data from the data set
  select(c(Year, State, Category, Item, Value)) |>
  filter((Year == 2019) & (Category == "ORGANIC - SALES") &
           (Item == "MEASURED IN $") & (Value != "(D)"))

straw2$Value <- as.numeric(gsub(",", "", straw2$Value))
straw2 <- straw1 |> arrange(desc(Value))
#This reorder allows us to make the graph cleaner because it orders them in size order.
ggplot(straw2, aes(reorder(State, -Value), 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 2019")
```



```r
strawberry <- read_csv("strawberries25_v3.csv", col_names = TRUE, show_col_types = FALSE)

#I modified the DOMAIN CATEGORY into three different categories.
strawberry <- strawberry %>%
  mutate(`Domain Category` = ifelse(is.na(`Domain Category`), "", `Domain Category`)) %>%
  mutate(
    Chemical = gsub(", .*", "", `Domain Category`),
#Get the first part before the comma
    Pesticide = gsub(".*, (.*):.*", "\\1", `Domain Category`),
#Get the second part before the colon
    Number = gsub(".*= (\\d+).*", "\\1", `Domain Category`)
#Get the number after '='
  )

print(strawberry)
```

```
## # A tibble: 12,669 x 24
##    Program  Year Period ‘Week Ending‘ ‘Geo Level‘ State   ‘State ANSI‘
##    <chr>   <dbl> <chr>  <lgl>         <chr>       <chr>   <chr>
##  1 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  2 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  3 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  4 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  5 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  6 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  7 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  8 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
##  9 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
## 10 CENSUS   2022 YEAR   NA            COUNTY      ALABAMA 01
## # i 12,659 more rows
## # i 17 more variables: ‘Ag District‘ <chr>, ‘Ag District Code‘ <dbl>,
## #   County <chr>, ‘County ANSI‘ <chr>, ‘Zip Code‘ <lgl>, Region <lgl>,
## #   watershed_code <chr>, Watershed <lgl>, Commodity <chr>, ‘Data Item‘ <chr>,
## #   Domain <chr>, ‘Domain Category‘ <chr>, Value <chr>, ‘CV (%)‘ <chr>,
## #   Chemical <chr>, Pesticide <chr>, Number <chr>
```

```r
write_csv(strawberry, "strawberries25_v3.csv")
```