Strawberry_EDA

Wenshuo Cai

2023-10-24

Load the data

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

glimpse(strawberry)
```

```
## Rows: 4,314
## Columns: 11
## $ ...1              <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Program           <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "C~
## $ Year              <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021~
## $ Period            <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEA~
## $ State             <chr> "ALASKA", "ALASKA", "ALASKA", "ALASKA", "ALASKA", "A~
## $ `State ANSI`      <chr> "02", "02", "02", "02", "02", "02", "02", "06", "06"~
## $ `Data Item`       <chr> "STRAWBERRIES, ORGANIC - OPERATIONS WITH SALES", "ST~
## $ Domain            <chr> "ORGANIC STATUS", "ORGANIC STATUS", "ORGANIC STATUS"~
## $ `Domain Category` <chr> "ORGANIC STATUS: (NOP USDA CERTIFIED)", "ORGANIC STA~
## $ Value             <chr> "2", "(D)", "(D)", "(D)", "2", "(D)", "(D)", "142", ~
## $ `CV (%)`          <chr> "(H)", "(D)", "(D)", "(D)", "(H)", "(D)", "(D)", "19~
```

Data Cleaning

```
## [1] "No single-value columns found"
```

```
## Rows: 4,314
## Columns: 11
## $ ...1              <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Program           <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "C~
## $ Year              <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021~
## $ Period            <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEA~
## $ State             <chr> "ALASKA", "ALASKA", "ALASKA", "ALASKA", "ALASKA", "A~
## $ `State ANSI`      <chr> "02", "02", "02", "02", "02", "02", "02", "06", "06"~
## $ `Data Item`       <chr> "STRAWBERRIES, ORGANIC - OPERATIONS WITH SALES", "ST~
## $ Domain            <chr> "ORGANIC STATUS", "ORGANIC STATUS", "ORGANIC STATUS"~
## $ `Domain Category` <chr> "ORGANIC STATUS: (NOP USDA CERTIFIED)", "ORGANIC STA~
## $ Value             <dbl> 2, NA, NA, NA, 2, NA, NA, 142, NA, NA, NA, 141, NA, ~
## $ `CV (%)`          <chr> "(H)", "(D)", "(D)", "(D)", "(H)", "(D)", "(D)", "19~
```

###Examine the data.

```
## is every line associated with a state?

## state_all1 contains the number of rows containing data
## for each of the 47 strawberry-growing states.
```

```
state_all1 <- strawberry |> group_by(State) |> count()

## test if every row is associated with a state by summing the
## counts and testing for equality with the total rows in the
## data frame

if(sum(state_all1$n) == dim(strawberry)[1]){print("Every row has value in the State column.")}
```

```
## [1] "Every row has value in the State column."
```

```
state_max <- state_all1$State[which(state_all1$n ==  max(state_all1$n)  )]
print(state_max)
```

```
## [1] "CALIFORNIA"
```

```
## filter rows of California data from the CENSUS data
calif_census <- strawberry |> filter((State=="CALIFORNIA") & (Program=="CENSUS"))


## ## filter rows of California data from the SURVEY data
calif_survey <- strawberry |> filter((State=="CALIFORNIA") & (Program=="SURVEY"))


census_col <- colnames(calif_census)

survey_col <- colnames(calif_survey)
```

```
#|label: split srawberry into census and survey pieces
#|echo:false

strwb_census <- strawberry |> filter(Program == "CENSUS")

strwb_survey <- strawberry |> filter(Program == "SURVEY")

## check that all of the rows are accounted for

nrow(strawberry) == (nrow(strwb_census) + nrow(strwb_survey))
```

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

```
## Move marketing-related rows in strw_b_chem
## to strw_b_sales
strwb_census <- strwb_census |>
  separate_wider_delim(  cols = `Data Item`,
                         delim = ",",
                         names = c("Fruit",
                                 "temp1",
                                 "temp2",
                                 "temp3"),
                         too_many = "error",
                         too_few = "align_start"
                       )



## split temp1 into crop_type, Prop_acct
```

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


# Define a function to clean and trim a column
clean_and_trim <- function(data_frame, column) {
  data_frame %>%
    mutate(!!column := str_trim(!!sym(column), side = "both")) %>%
    mutate(!!column := na_if(!!sym(column), "")) # Convert NA to empty string
}

# Define a function to remove specific patterns from a column
remove_patterns <- function(data_frame, column, patterns) {
  for (pattern in patterns) {
    data_frame <- data_frame %>%
      mutate(!!column := str_replace_all(!!sym(column), pattern, ""))
  }
  data_frame
}

# Clean and trim columns
strwb_census <- clean_and_trim(strwb_census, "crop_type")
strwb_census <- clean_and_trim(strwb_census, "temp2")
strwb_census <- clean_and_trim(strwb_census, "temp3")

# Handle `Fresh Market` column
strwb_census <- strwb_census %>%
  mutate(`Fresh Market` = temp2) %>%
  remove_patterns("Fresh Market", c("^MEA.*", "^P.*", "^FRESH MARKET - "))

# Handle `Process Market` column
strwb_census <- strwb_census %>%
  mutate(`Process Market` = temp2) %>%
  remove_patterns("Process Market", c("^MEA.*", "PROCESSING - "))

# Handle `temp2` column
strwb_census <- remove_patterns(strwb_census, "temp2", c("^F.*", "^P.*"))

# Combine `temp2` and `temp3`, clean `Metric` column, and relocate columns
strwb_census <- strwb_census %>%
  unite("Metric", c("temp2", "temp3"), sep = "") %>%
  mutate(Metric = str_replace(Metric, "MEASURED IN ", "")) %>%
  relocate(Metric, .before = Domain) %>%
  rename(Totals = Prop_acct)
```

```
##  [1]   2  NA  NA  NA   2  NA  NA 142  NA  NA  NA 141  NA  NA   7  NA  NA   8 183
## [20]  NA
```

```
## $Indices
## integer(0)
##
## $Footnotes
## [1] NA
```

```
##   [1]   2  NA 142 141   7   8 183  10   4  19   6 111   5  53  74  12  62  16
##  [19] 504  17 127 251  26 564  21 749  18 659   3  90  11  96   1   9 132  45
##  [37]  43 734 360  27 522  31  49  80  28 426 174 170  15 181 177  14 100 288
##  [55]  70  40 168  13 158 485  30 435  73  39 208  36  34 300 315 880 341 332
##  [73] 219  57  23  22 135 189 184  48 206 279 267 161  60 402  56 715 427 254
##  [91] 253 780  32 534  25 285 138 159  42 374  29  52 335 442
```

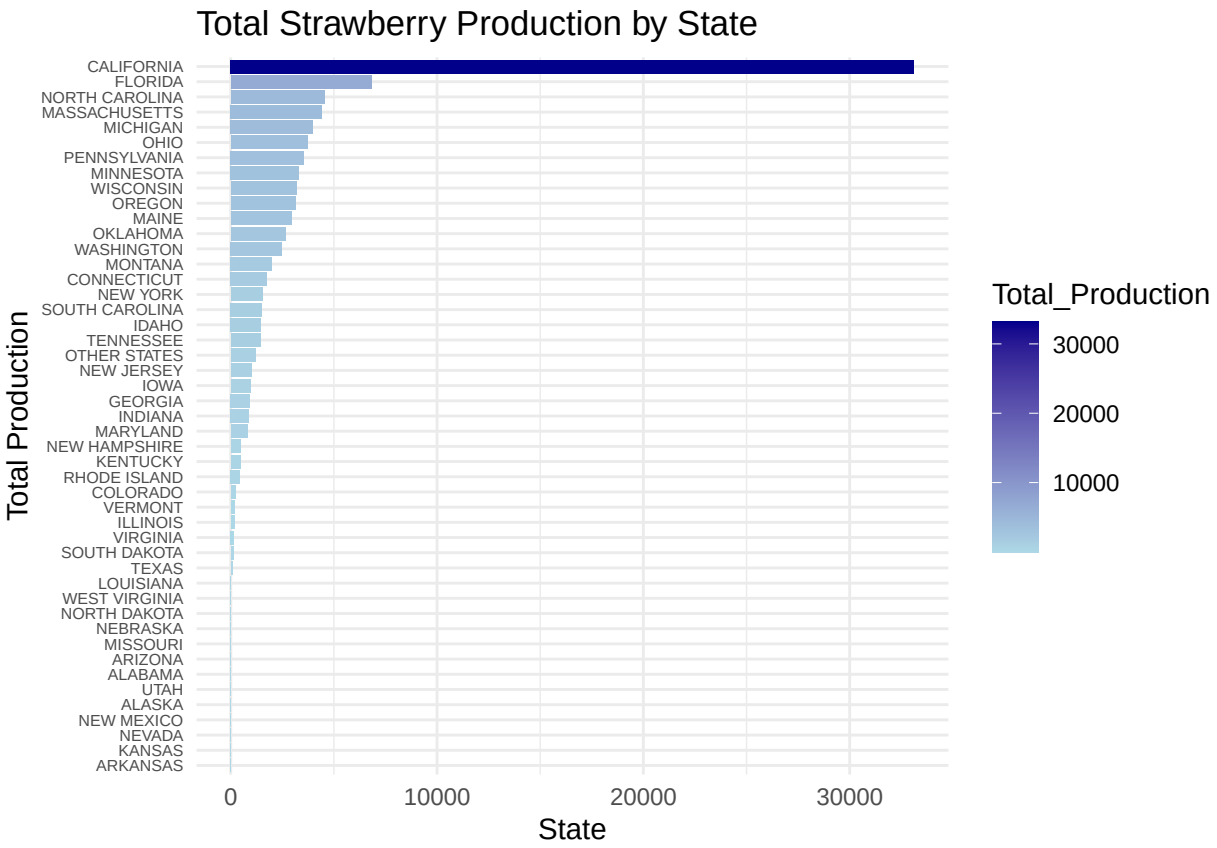
```
##  [1]  NA  NA  NA   2  NA  NA  NA  53  96  NA 183  NA  NA   9  NA  NA   1   2  NA
## [20]  NA
```

EDA

```
library(dplyr)
library(ggplot2)

# Load the dataset
strawberry_data <- read.csv("strawberry.csv")

# Convert the 'Value' column to numeric, handling non-numeric entries gracefully
strawberry_data$Value <- as.numeric(as.character(strawberry_data$Value))
```

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

```
# Aggregate the strawberry production by state
strawberry_production_by_state <- strawberry_data %>%
  group_by(State) %>%
  summarise(Total_Production = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(Total_Production))

# Filter out low-production states (e.g., states with Total_Production >= a threshold)
production_threshold <- 500  # Define your threshold here
strawberry_production_filtered <- strawberry_production_by_state %>%
  filter(Total_Production >= production_threshold)

# Create a bar plot to visualize the strawberry production by state with gradient fill
ggplot(strawberry_production_by_state, aes(x = Total_Production, fill = Total_Production, y = reorder(S
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Gradient fill
  labs(
    title = "Total Strawberry Production by State",
    x = "State",
    y = "Total Production"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 5.75))  # Adjust the size of y-axis labels
```



California Dominates Strawberry Production: The state of California stands out as the leading producer of strawberries, with the highest total production. It is indicated by the longest horizontal bar on the chart. While California leads in production, there is a considerable variation in production levels among other states. Some states have relatively low production, while others have moderate to high production.

References

NASS help

Quick Stats Glossary

Quick Stats Column Definitions