Data Analysis on Strawberry Farming

Madeleine Livaudais

Where do strawberries come from in the USA?

When you go the market and look in the produce section, you see all sorts of fruits and vegetables, but do you ever stop to think about where they are coming from or the farming process that got them there? Here we will specifically look at the production of strawberries in the United States.

The United States Department of Agriculture (USDA) collects census and survey data regularly on all of the major crops grown in the United States. To analyze the strawberry crop, we will look at data collected by the USDA through census' and surveys from 1997 to 2025. We will focus on the six states that produce the majority of the strawberries consumed: California, Florida, Washington, Oregon, North Carolina and New York.

Main data source: USDA Quickstats

What fertilizers are being used in strawberry farming?

My group was asked to research fertilizer use on strawberry crops. We sourced some data to analyze this from the same USDA database above. The four fertilizers that the USDA collected data about are: Nitrogen, Phosphate, Potash and Sulfur. Nitrogen is helpful to support green, leafy growth. Phosphate helps the root, flower and fruit development. Potash (or potassium) helps a plant's stress tolerance and disease resistance. Sulfur helps a overall plant health and enzyme and nutrient metabolism. These are all important nutrients for healthy plant growth, along with others.

Are these fertilizers used on conventional and/or organic strawberries?

Since all four of these nutrients are key for crop growth, they are used by farmers growing conventional and organic strawberries. However, the source of the fertilizer differs because there are organic and inorganic fertilizers.

(Need to add more on this… just haven't finished the research yet)

Data Cleaning Steps

First, we read in the data from our USDA source and remove any columns that only have one data value and thus will not be useful in our data analysis.

```
# Read in the csv data file
strawb <- read_csv("strawberry_10Oct25.csv", col_names = TRUE)
```

```
Rows: 2710 Columns: 21
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (12): Program, Period, Geo Level, State, State ANSI, watershed_code, Co...
dbl   (1): Year
lgl   (7): Ag District, Ag District Code, County, County ANSI, Zip Code, Reg...
date  (1): Week Ending

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.
```

```
# Run the function on the strawberry data
strawb <- drop_one_value_col(strawb)
```

```
[1] "Columns dropped: Geo Level, Ag District, Ag District Code, County, County ANSI, Zip Code
```

Upon first look at the data, we have numeric data. In order to be able to work with the data and perform exploratory analysis, we must convert the data into a numeric structure.

```
# Converting the Value and CV(%) columns from character into numeric structure.
strawb <- strawb %>%
  mutate(Value = as.numeric(gsub(",", "", strawb$Value))) %>%
  mutate(`CV (%)` = as.numeric(strawb$`CV (%)`))
```

The column `Data Item` gives us a lot of information. In order to make it more useful, we will remove the phrases common to all elements of the data set. Thus, it becomes a more informative column about the numeric data values.

```
# Remove strawberries from the start of every line in Data Item
strawb$`Data Item` <- str_replace(strawb$`Data Item`, "^ST.*IES((\ -\ )|(,\ ))", "")
```

The data comes from surveys and census'. There are some differences in the data collection that will make it difficult to clean the dataset while it is all together. Here we split the data into separate survey and census data frames for further cleaning.

```
# Filter for only the SURVEY data
strawb_survey <- strawb %>%
  filter(Program == "SURVEY")

# Filter for only the CENSUS data
strawb_census <- strawb %>%
  filter(Program == "CENSUS")

# Run drop one value columns again
strawb_survey <- drop_one_value_col(strawb_survey)
```

```
[1] "Columns dropped: Program, CV (%)"
```

```
strawb_census <- drop_one_value_col(strawb_census)
```

```
[1] "Columns dropped: Program, Period, Week Ending"
```

Census Data for strawberries

Once we've split the data, the census data is ready to pivot so that each row represents the year for a state and each observation for that year is in a column.

```
strawb_census <- strawb_census %>%
  pivot_wider(
    id_cols = c(Year, State),          # Columns that uniquely identify each new wide row
    names_from = `Data Item`,          # Column whose values will become the new column names
    values_from = c(Value, `CV (%)`)   # Column whose values will fill the new columns
  )
```

Survey Data for strawberries

Now that the census data is complete, let's take a look at the survey data.

The survey data is a bit more complicated because each state has a bit of variability in how they report the information. We will need to split up the survey data by state. Before making

that split, some of the values are a forecast for the year, but the year value is also provided. Because of this, if the `Period` column contains the word "forecast" we will remove that line from the data set.

```
# Remove rows predicting "FORCAST"
strawb_survey <- strawb_survey %>%
  filter(!str_detect(Period, "FORECAST"))
```

Now, let's split the survey data up into the data from each of the six states and look at each state separately. For most of the states, we will remove the `Week Ending` column and the `State ANSI` column as they would be removed by the drop one value column function. I also wrote a function to drop duplicate rows since sometimes everything is the same except the period, which differs as "year" to "marketing year". This function will be run on each state's data set. Below we can clean New York, Washington, and Oregon with little effort.

```
# Filter for NY and remove unneeded columns
new_york <- strawb_survey %>%
  filter(State == "NEW YORK") %>%
  select(-`Week Ending`, -`State ANSI`)

# Run function to remove duplicate rows
new_york <- drop_duplicate_rows(new_york)
```

```
# Filter for WA and remove unneeded columns
washington <- strawb_survey %>%
  filter(State == "WASHINGTON") %>%
  select(-`Week Ending`, -`State ANSI`)

# Run function to remove duplicate rows
washington <- drop_duplicate_rows(washington)
```

```
# Filter for OR and remove unneeded columns
oregon <- strawb_survey %>%
  filter(State == "OREGON") %>%
  select(-`Week Ending`, -`State ANSI`)

# Run function to remove duplicate rows
oregon <- drop_duplicate_rows(oregon)
```

North Carolina reported their data in weekly increments for the `Period` column, so we needed to address that and average out the year based on the weeks.

```r
# Filter for NC and remove unneeded columns
ncarolina <- strawb_survey %>%
  filter(State == "NORTH CAROLINA") %>%
  select(-`State ANSI`)

# Run function to remove duplicate rows
ncarolina <- drop_duplicate_rows(ncarolina)

# Summarize the weekly values to a year average
ncarolina <- ncarolina %>%
  group_by(Year, State, `Data Item`) %>%
  summarize(Value = sum(Value), Period = "YEAR", .groups = "drop") %>%
  ungroup() %>%
  # remove the rows with period as previous year because we have them as the actual year
  filter(!str_detect(`Data Item`, "PREVIOUS YEAR,")) %>%
  relocate(Period, .after = Year)
```

California and Florida below ran the drop duplicate rows function and then needed an extra step to remove the monthly data as we are focusing on yearly data. The monthly data was just a further breakdown of the year data that already existed.

```r
# Filter for CA and remove unneeded columns
calif <- strawb_survey %>%
  filter(State == "CALIFORNIA") %>%
  select(-`Week Ending`, -`State ANSI`) %>%
  filter(!str_detect(Period, "FORECAST"))

# Run function to remove duplicate rows
calif <- drop_duplicate_rows(calif)

# Filter for year periods
calif <- calif %>%
  filter(str_detect(Period, "YEAR"))
```

```r
# Filter for FL and remove unneeded columns
florida <- strawb_survey %>%
  filter(State == "FLORIDA") %>%
  select(-`Week Ending`, -`State ANSI`) %>%
  filter(!str_detect(Period, "FORECAST"))

# Run function to remove duplicate rows
florida <- drop_duplicate_rows(florida)
```

```
# Filter for year periods
florida <- florida %>%
  filter(str_detect(Period, "YEAR"))
```

Once each state is cleaned up, we can recombine the data into a single data frame and use pivot wider to better analyze the survey data.

```
# Create a list containing all of them
states_list <- list(new_york, oregon, ncarolina, calif, florida, washington)

# Bind all the data frames together into a single data table
strawb_survey <- rbindlist(states_list)

# Clean up the environment
rm(calif, florida, new_york, washington, ncarolina, oregon, states_list)
```

```
strawb_survey <- strawb_survey %>%
  pivot_wider(
    id_cols = c(Year, State),     # Columns that uniquely identify each new wide row
    names_from = `Data Item`,     # Column whose values will become the new column names
    values_from = Value           # Column whose values will fill the new columns
  )
```

Fertilizer Data for strawberries

We sourced the data from the USDA and first read in the csv file and run the drop one value columns function to remove unnecessary columns from the data.

```
# Read in csv file from USDA
strawb_fertilizers <- read.csv("strawberry_fertilizers.csv")

# Run the function on the strawberry data
strawb_fertilizers <- drop_one_value_col(strawb_fertilizers)
```

```
[1] "Columns dropped: Program, Period, Week.Ending, Geo.Level, Ag.District, Ag.District.Code
```

To make sure all the numeric data is useable in analysis, we change the class of the data to numeric rather than characters.

```r
# Convert to numeric
strawb_fertilizers <- strawb_fertilizers %>%
  mutate(Value = as.numeric(gsub(",", "", Value)))
```

Now let's clean up the columns that will become our column headers. We need to edit the column so we know which fertilizer is being discusses and then edit the `Data Item` column for what procedure is being performed with the designated fertilizer.

```r
# Rename Fertilizer column and clean it up
strawb_fertilizers <- strawb_fertilizers %>%
  rename(Fertilizer = Domain.Category) %>%
  mutate(Fertilizer = sub(".*\\((.*?)\\).*", "\\1", Fertilizer))
```

```r
# Remove strawberries from the start of every line in Data Item
strawb_fertilizers <- strawb_fertilizers %>%
  mutate(Data.Item = str_remove(Data.Item, ".*-"))
```

Once the character columns are cleaner, we can then pivot wider, so each row represents a single year for the state.

```r
strawb_fertilizers <- strawb_fertilizers %>%
  pivot_wider(
    id_cols = c(Year, State),  # Columns that uniquely identify each new wide row
    names_from = c(Fertilizer, Data.Item), # Columns whose values will become the new column
    values_from = Value   # Column whose values will fill the new columns
  )
```

What can graphs/visuals tell us about this data?

```r
# Haven't gotten to this yet, but will use the data to make some graphs.
```