Strawberries

2024-10-11

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

We determine whether or not every line is associated with a state and if it is, we return to the reader stating that it does.

```r
## Is every line associated with a state?
state_all <- strawberry |> distinct(State)
state_all1 <- strawberry |> group_by(State) |> count()

## every row is associated with a state
if(sum(state_all1$n) == dim(strawberry)[1]){print("Yes every row in the data is associated with a state
```

```
## [1] "Yes every row in the data is associated with a state."
```

We define a function called drop_one_value_col and check to see if there is one unique value. This is because we want to get rid of the character values which is "NA" so that we can continue with just values in our data frame.

```r
#|label: function def - drop 1-item columns

# Define a function to drop columns that contain only one unique value
drop_one_value_col <- function(df){
  drop <- NULL  # Initialize an empty vector to keep track of columns to drop

  # Loop through each column in the dataframe
  for(i in 1:dim(df)[2]){
    # Check if the current column has only one unique value
    if((df |> distinct(df[, i]) |> count()) == 1){
      drop = c(drop, i)  # Add the index of the column to the 'drop' vector
    }
  }

  # If no columns were found to drop, return "none"
  if(is.null(drop)){
    return("none")
  }
  else{
    print("Columns dropped:")
    print(colnames(df)[drop])
    strawberry <- df[, -1 * drop]  # Remove the identified columns from the dataframe
  }
}
```

```
# Apply the function to the 'strawberry' dataframe to drop one-value columns
strawberry <- drop_one_value_col(strawberry)
```

```
## [1] "Columns dropped:"
## [1] "Week Ending"    "Zip Code"       "Region"         "watershed_code"
## [5] "Watershed"      "Commodity"
```

```
# Call the function again on the updated 'strawberry' dataframe
drop_one_value_col(strawberry)
```

```
## [1] "none"
```

We clean the dataset by filtering out any rows that are not related to the national or state level. This is to make the data more precise for future analysis.

```
# Get unique values in the 'Geo Level' column of the 'strawberry' dataframe
unique(strawberry$`Geo Level`)
```

```
## [1] "COUNTY"   "NATIONAL" "STATE"
```

```
# Filter the 'strawberry' dataframe to keep only rows where 'Geo Level' is either "NATIONAL" or "STATE"
strawberry <- strawberry |> filter(`Geo Level` == "NATIONAL" | `Geo Level` == "STATE")
```

We split the original Strawberries data into census and survey data frames to work with.

```
straw_cen <- strawberry |> filter(Program == "CENSUS")
straw_cen <- straw_cen |> drop_one_value_col()
```

```
## [1] "Columns dropped:"
## [1] "Program"          "Period"           "Ag District"      "Ag District Code"
## [5] "County"           "County ANSI"
```

```
straw_sur <- strawberry |> filter(Program == "SURVEY")
straw_sur <- straw_sur %>% drop_one_value_col()
```

```
## [1] "Columns dropped:"
## [1] "Program"          "Ag District"      "Ag District Code" "County"
## [5] "County ANSI"      "CV (%)"
```

```
nrow(strawberry) == (nrow(straw_sur) + nrow(straw_cen))
```

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

In our straw_sur we have the Domain Category that needs cleaning. The reason for this is that there is a lot of info in one column that will be hard to pinpoint when we need specific data. Therefore, we are splitting into separate categories: Chemical and Number so that we can easily call what we want.

```r
# Modify the 'straw_sur' dataframe using a series of transformations
straw_sur <- straw_sur %>%
  # Replace the content of 'Domain Category' with a new format using gsub
  mutate(`Domain Category` = gsub(".*: \\(([^=]+) = ([0-9]+)\\)", "\\1,\\2", `Domain Category`)) %>%

  # Separate the 'Domain Category' into two new columns: 'Chemical' and 'Number'
  separate(`Domain Category`, into = c("Chemical", "Number"), sep = ",") %>%

  mutate(Chemical = trimws(Chemical),
         Number = as.numeric(trimws(Number)))
```

We separate the Domain into Domain and use.

```r
# Modify the 'straw_sur' dataframe to separate the 'Domain' column into two new columns: 'Domain' and '
straw_sur <- straw_sur %>%
  separate(Domain, into = c("Domain", "use"), sep = ",", extra = "merge")
```

We clean our straw_sur data frame into more specific detail.

```r
# Modify the 'straw_sur' dataframe using a series of transformations
straw_sur <- straw_sur %>%
  mutate(measurement = str_extract(`Data Item`, "(?<=MEASURED\\s).*")) %>%
  mutate(`Data Item` = str_remove(`Data Item`, "MEASURED.*")) %>%
  separate(`Data Item`, into = c("Data Item", "category"), sep = "[,-]", extra = "merge", fill = "right

# Select specific columns from the 'straw_sur' dataframe
straw_sur <- straw_sur %>%
  select(1:7, 13, 8:12)

straw_sur <- straw_sur %>%
  # Remove the word 'IN' from the 'measurement' column
  mutate(measurement = gsub("\\bIN\\b", "", measurement)) %>%
  mutate(measurement = trimws(measurement))

# Select specific columns from the 'straw_sur' dataframe again
straw_sur <- straw_sur %>%
  select(1:8, 13, 9:12)

# Remove the 'Data Item' column from the 'straw_sur' dataframe
straw_sur <- straw_sur %>%
  select(-`Data Item`)

# Remove any commas from the 'category' column in the 'straw_sur' dataframe
straw_sur$category <- gsub(",", "", straw_sur$category)
```

We split the straw_sur data frame into two more specific data frames called sur_total and sur_chem.

```r
sur_total <- straw_sur %>% filter(Domain == "TOTAL")
sur_chem <- straw_sur %>% filter(Domain == "CHEMICAL")
sur_total = drop_one_value_col(sur_total)
```

```
## [1] "Columns dropped:"
## [1] "Domain"   "use"      "Chemical" "Number"
```

```r
sur_chem = drop_one_value_col(sur_chem)
```

```
## [1] "Columns dropped:"
## [1] "Period"    "Geo Level" "Domain"
```

We clean the straw_cen data frame by creating two new columns: “strawberries” and “category”

```r
# Modify the 'straw_cen' dataframe by separating the 'Data Item' column into two new columns: 'strawber
straw_cen <- straw_cen |>
  separate_wider_delim(cols = `Data Item`, delim = " - ",
                       names = c("strawberries", "Category"),
                       too_many = "error", too_few = "align_start")
```

We modify the straw_cen dataframe by separating the “strawberries” column into three new columns: “strawberries”, “ORGANIC”, and “organic_detail”.

```r
# Modify the 'straw_cen' dataframe by separating the 'strawberries' column into three new columns: 'str
straw_cen <- straw_cen |>
  separate_wider_delim(cols = strawberries, delim = ", ",
                       names = c("strawberries", "ORGANIC", "organic_detail"),
                       too_many = "error", too_few = "align_start")

straw_cen <- straw_cen |> drop_one_value_col()
```

```
## [1] "Columns dropped:"
## [1] "strawberries"
```

```r
organic_cen <- straw_cen |> filter(ORGANIC == "ORGANIC")

sum(is.na(straw_cen$ORGANIC))
```

```
## [1] 662
```

```r
straw_cen <- straw_cen[(is.na(straw_cen$ORGANIC)), ]
straw_cen <- straw_cen |> drop_one_value_col()
```

```
## [1] "Columns dropped:"
## [1] "Year"           "ORGANIC"        "organic_detail"
```

We modify the straw_cen dataframe by separating the “Category” column into two new columns: “COL1” and “COL2”

```r
# Modify the 'straw_cen' dataframe by separating the 'Category' column into two new columns: 'COL1' and
straw_cen <- straw_cen |>
  separate_wider_delim(cols = `Category`, delim = " ",
                       names = c("COL1", "COL2"),
                       too_many = "merge", too_few = "align_start")

# Remove the word "WITH" from the 'COL2' column
straw_cen$COL2 <- str_replace(straw_cen$COL2, "WITH ", "")

# Rename the columns 'COL1' and 'COL2' to 'Measure' and 'Bearing_type' respectively
straw_cen <- straw_cen |> rename(Measure = COL1, Bearing_type = COL2)
```

We clean the straw_cen data frame and standardize the data.

```
straw_cen <- straw_cen |> rename(size_bracket = `Domain Category`)
straw_cen$size_bracket <- str_replace(straw_cen$size_bracket, "NOT SPECIFIED", "TOTAL")
straw_cen$size_bracket <- str_replace(straw_cen$size_bracket, "AREA GROWN: ", "")
```

We separate the Category column into two new columns: “Category” and “measurement”

```
organic_cen <- organic_cen |> drop_one_value_col()
```

```
## [1] "Columns dropped:"
## [1] "ORGANIC"          "Domain"          "Domain Category"
```

```
organic_cen <- organic_cen %>%
  separate(Category, into = c("Category", "measurement"), sep = " MEASURED ", extra = "merge", fill = "
organic_cen$Category <- gsub(",", "", organic_cen$Category)
organic_cen$measurement <- gsub("IN", "", organic_cen$measurement, ignore.case = TRUE)
```

We create a new dataframe “no_NA” by modifying “straw_cen”

```
no_NA <- straw_cen |>
  mutate(Value = na_if(Value, "(D)"),
         Value = na_if(Value, "(Z)"),
         Value = as.numeric(gsub(",", "", Value))) |>
  na.omit()

# Fit a linear model 'fit1' predicting 'Value' based on 'Measure', 'Bearing_type', 'State', and 'size_b
fit1 <- lm(Value ~ Measure + Bearing_type + State + size_bracket, data = no_NA)
```

From here on out, we will be only dealing with the outputs. All the cleaning that we want is not complete and now it is time to show what we can do with the new cleaned data frames what we have from our original “Strawberries” data frame.

```
#|label: listing out the data frames

print(head(strawberry))
```

```
## # A tibble: 6 x 15
##   Program  Year Period ‘Geo Level‘ State    ‘State ANSI‘ ‘Ag District‘
##   <chr>   <dbl> <chr>  <chr>       <chr>    <chr>        <chr>
## 1 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## 2 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## 3 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## 4 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## 5 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## 6 CENSUS   2022 YEAR   NATIONAL    US TOTAL <NA>         <NA>
## # i 8 more variables: ‘Ag District Code‘ <dbl>, County <chr>,
## #   ‘County ANSI‘ <chr>, ‘Data Item‘ <chr>, Domain <chr>,
## #   ‘Domain Category‘ <chr>, Value <chr>, ‘CV (%)‘ <chr>
```

```r
print(head(straw_sur))
```

```
## # A tibble: 6 x 12
##    Year Period  ‘Geo Level‘ State ‘State ANSI‘ category measurement Value Domain
##   <dbl> <chr>   <chr>       <chr> <chr>        <chr>    <chr>       <chr> <chr>
## 1  2024 YEAR    NATIONAL    US T~ <NA>         " FRESH~ $ / CWT     10.9  TOTAL
## 2  2024 YEAR    NATIONAL    US T~ <NA>         " PROCE~ $ / TON     4.04  TOTAL
## 3  2023 MARKET~ NATIONAL    US T~ <NA>         " PRICE~ $ / CWT     123   TOTAL
## 4  2023 MARKET~ NATIONAL    US T~ <NA>         " FRESH~ $ / CWT     142   TOTAL
## 5  2023 MARKET~ NATIONAL    US T~ <NA>         " PROCE~ $ / CWT     43.8  TOTAL
## 6  2023 MARKET~ STATE       CALI~ 06           " PRICE~ $ / CWT     121   TOTAL
## # i 3 more variables: use <chr>, Chemical <chr>, Number <dbl>
```

```r
print(head(straw_cen))
```

```
## # A tibble: 6 x 9
##   ‘Geo Level‘ State  ‘State ANSI‘ Measure Bearing_type Domain size_bracket Value
##   <chr>       <chr>  <chr>        <chr>   <chr>        <chr>  <chr>        <chr>
## 1 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (0.1 TO 0.9~ 963
## 2 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (1.0 TO 4.9~ 3,195
## 3 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (100 OR MOR~ 46,2~
## 4 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (15.0 TO 24~ 2,514
## 5 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (25.0 TO 49~ 4,231
## 6 NATIONAL    US TO~ <NA>         ACRES   BEARING      AREA ~ (5.0 TO 14.~ 3,396
## # i 1 more variable: ‘CV (%)‘ <chr>
```

```r
print(head(sur_chem))
```

```
## # A tibble: 6 x 9
##    Year State      ‘State ANSI‘ category measurement Value use   Chemical Number
##   <dbl> <chr>      <chr>        <chr>    <chr>       <chr> <chr> <chr>     <dbl>
## 1  2023 CALIFORNIA 06           " APPLI~ LB          (D)   " FU~ OXATHIA~ 128111
## 2  2023 CALIFORNIA 06           " APPLI~ LB          (D)   " IN~ CYCLANI~  26202
## 3  2023 CALIFORNIA 06           " APPLI~ LB          (D)   " IN~ PERMETH~ 109701
## 4  2023 CALIFORNIA 06           " APPLI~ LB          (NA)  " OT~ ISARIA ~ 115003
## 5  2023 CALIFORNIA 06           " APPLI~ LB / ACRE ~ (D)   " FU~ OXATHIA~ 128111
## 6  2023 CALIFORNIA 06           " APPLI~ LB / ACRE ~ (D)   " IN~ CYCLANI~  26202
```

```r
print(head(sur_total))
```

```
## # A tibble: 6 x 8
##    Year Period          ‘Geo Level‘ State ‘State ANSI‘ category measurement Value
##   <dbl> <chr>           <chr>       <chr> <chr>        <chr>    <chr>       <chr>
## 1  2024 YEAR            NATIONAL    US T~ <NA>         " FRESH~ $ / CWT     10.9
## 2  2024 YEAR            NATIONAL    US T~ <NA>         " PROCE~ $ / TON     4.04
## 3  2023 MARKETING YEAR NATIONAL    US T~ <NA>         " PRICE~ $ / CWT     123
## 4  2023 MARKETING YEAR NATIONAL    US T~ <NA>         " FRESH~ $ / CWT     142
## 5  2023 MARKETING YEAR NATIONAL    US T~ <NA>         " PROCE~ $ / CWT     43.8
## 6  2023 MARKETING YEAR STATE       CALI~ 06           " PRICE~ $ / CWT     121
```

```
print(head(organic_cen))
```

```
## # A tibble: 6 x 9
##    Year ‘Geo Level‘ State ‘State ANSI‘ organic_detail Category measurement Value
##   <dbl> <chr>       <chr> <chr>        <chr>          <chr>    <chr>       <chr>
## 1  2021 NATIONAL    US T~ <NA>         <NA>           ACRES H~  <NA>       5,301
## 2  2021 NATIONAL    US T~ <NA>         <NA>           OPERATI~  <NA>       546
## 3  2021 NATIONAL    US T~ <NA>         <NA>           OPERATI~  <NA>       546
## 4  2021 NATIONAL    US T~ <NA>         <NA>           PRODUCT~ " CWT"      1,49~
## 5  2021 NATIONAL    US T~ <NA>         <NA>           SALES    " $"        335,~
## 6  2021 NATIONAL    US T~ <NA>         <NA>           SALES    " CWT"      1,49~
## # i 1 more variable: ‘CV (%)‘ <chr>
```

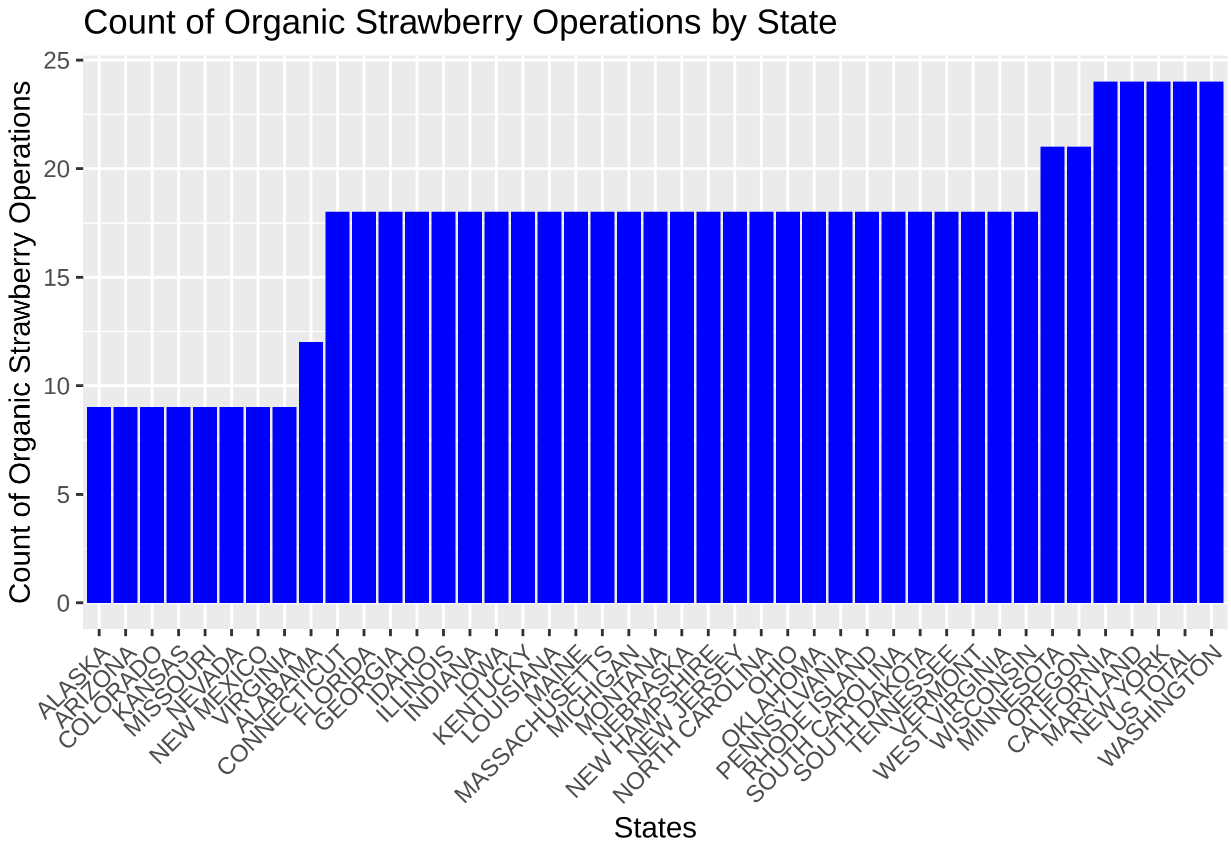
```
print(head(no_NA))
```

```
## # A tibble: 6 x 9
##   ‘Geo Level‘ State  ‘State ANSI‘ Measure Bearing_type Domain size_bracket Value
##   <chr>       <chr>  <chr>        <chr>   <chr>        <chr>  <chr>        <dbl>
## 1 STATE       ALABA~ 01           ACRES   BEARING      TOTAL  TOTAL          162
## 2 STATE       ALABA~ 01           ACRES   GROWN        TOTAL  TOTAL          171
## 3 STATE       ALABA~ 01           ACRES   NON-BEARING  TOTAL  TOTAL            9
## 4 STATE       ALABA~ 01           OPERAT~ AREA BEARING TOTAL  TOTAL          107
## 5 STATE       ALABA~ 01           OPERAT~ AREA GROWN   TOTAL  TOTAL          119
## 6 STATE       ALABA~ 01           OPERAT~ AREA NON-BE~ TOTAL  TOTAL           18
## # i 1 more variable: ‘CV (%)‘ <chr>
```

```
#|label: Count of Organic Strawberry Operations by State Plot

# Count of organic operations by state
state_counts <- organic_cen %>%
  group_by(State) %>%
  summarize(Count = n())

ggplot(state_counts, aes(x = reorder(State, Count), y = Count)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "States", y = "Count of Organic Strawberry Operations",
       title = "Count of Organic Strawberry Operations by State")
```



```r
#|label: showing every number in numerical order

# Adjust the column names based on your findings
chemical_summary <- sur_chem %>%
  group_by(Chemical) %>%
  summarize(Total_Number = sum(Number, na.rm = TRUE), .groups = "drop") %>%
  arrange(Total_Number)

# Print the summary
print(chemical_summary)
```

```
## # A tibble: 172 x 2
##    Chemical                Total_Number
##    <chr>                          <dbl>
##  1 2                                  0
##  2 CHEMICAL                           0
##  3 HYDROGEN PEROXIDE              11900
##  4 BIFENAZATE                     23440
##  5 MUSTARD OIL                    24505
##  6 BT KURSTAKI EG7841             32265
##  7 BT KURSTAKI SA-12              65180
##  8 MANCOZEB                       72520
##  9 STREPTOMYCES LYDICUS           94905
## 10 BT SUB AIZAWAI GC-91           96390
## # i 162 more rows
```