

615 Assignment Strawberries 1

Haoran Cui

2024-10-02

#Preparing data for analysis —— Strawberries

```
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    3.4.4      v tibble    3.2.1
## v lubridate  1.9.3      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 become errors
```

```
library(dplyr)
library(readr)
library(tidyr)
library(stringr)
library(ggplot2)
```

```
# Load the data from a CSV file and view the first few rows
strawberry <- read_csv("strawberries25_v3.csv", col_names = TRUE)
```

```
## Rows: 12669 Columns: 21
## -- Column specification -----
## Delimiter: ","
## chr (15): 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.
```

```
head(strawberry)
```

```
## # A tibble: 6 x 21
##   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
## # i 14 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>

# Replace any occurrences of "(D)" in Value and CV% columns with NA (missing value)
strawberry <- strawberry %>%
  mutate(
    Value = ifelse(Value == "(D)", NA, Value),
    `CV (%)` = ifelse(`CV (%)` == "(D)", NA, `CV (%)`)
  )
head(strawberry)
```

```
## # A tibble: 6 x 21
##   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
## # i 14 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>
```

```
# Rearrange 'Domain' column into three new columns: chemical category, name, and number
strawberry <- strawberry %>%
  mutate(
    Category = case_when(
      Domain == "Total" ~ NA_character_, # If Domain is "Total", mark as NA
      str_detect(Domain, "CHEMICAL") ~ str_trim(str_remove(Domain, "CHEMICAL, ")), # Remove "CHEMICAL, "
      TRUE ~ Domain
    )
  )
unique(strawberry$Category)
```

```
## [1] "TOTAL" "AREA GROWN" "ORGANIC STATUS" "FUNGICIDE"
## [5] "INSECTICIDE" "OTHER" "HERBICIDE" "FERTILIZER"
head(strawberry)
```

```
## # A tibble: 6 x 22
##   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
## # i 15 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>,
## # Category <chr>
```

```
# Extract "Name" and "Number" from the 'Domain Category' column
```

```
strawberry <- strawberry %>%
```

```
  mutate(
```

```
    Name = case_when(
```

```
      Category == "TOTAL" ~ NA_character_, # If Category is "TOTAL", mark as NA
```

```
      str_detect(`Domain Category`, fixed(Category)) & str_detect(`Domain Category`, "\\(.*\\.\\)") ~
```

```
        str_extract(`Domain Category`, "(?<=\\(\\.\\).*?(?=\\s|=)"), # Extract Name from Domain Category
```

```
      str_detect(`Domain Category`, fixed(Category)) & str_detect(`Domain Category`, "\\(.*\\.\\)") ~
```

```
        str_extract(`Domain Category`, "(?<=\\(\\.\\).*?(?=\\s|=)"), # Another pattern for extraction
```

```
      TRUE ~ NA_character_
```

```
    ),
```

```
    Number = case_when(
```

```
      Category == "TOTAL" ~ NA_real_, # If Category is "TOTAL", mark as NA
```

```
      str_detect(`Domain Category`, fixed(Category)) & str_detect(`Domain Category`, "\\(.*\\.\\)") ~
```

```
        as.numeric(str_extract(`Domain Category`, "(?<=\\(\\.\\).*?(?=\\s|=)")), # Extract Number from Domain Category
```

```
      str_detect(`Domain Category`, fixed(Category)) & str_detect(`Domain Category`, "\\(.*\\.\\)") ~
```

```
        NA_real_, # If no number, mark as NA
```

```
      TRUE ~ NA_real_
```

```
    )
```

```
  )
```

```
strawberry <- strawberry %>%
```

```
  mutate(Category = case_when(
```

```
    `Domain Category` == "NOT SPECIFIED" ~ NA_character_, # If Domain Category is "NOT SPECIFIED", mark as NA
```

```
    TRUE ~ Category # Otherwise, retain the existing Category
```

```
  ))
```

```
head(strawberry)
```

```
## # A tibble: 6 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
```

```
## # 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>,
```

```
## # Category <chr>, Name <chr>, Number <dbl>
```

```
# Clean and extract numerical intervals for planted area, creating Min and Max columns
```

```
strawberry <- strawberry %>%
```

```
  mutate(
```

```
    Min = case_when(
```

```
      str_detect(Name, "100 OR MORE ACRES") ~ 100, # If the text says "100 OR MORE ACRES", Min is 100
```

```
      str_detect(Name, "TO") ~ as.numeric(str_extract(Name, "[0-9.]+")), # Extract Min value from interval
```

```
      TRUE ~ NA_real_
```

```
    ),
```

```

    Max = case_when(
      str_detect(Name, "100 OR MORE ACRES") ~ "MORE", # For "100 OR MORE ACRES", Max is "MORE"
      str_detect(Name, "TO") ~ str_extract(Name, "(?<=TO )^[0-9.]+)", # Extract Max value from interval
      TRUE ~ NA_character_
    )
  )
)

# View the cleaned data
head(strawberry)

## # A tibble: 6 x 26
##   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
## # i 19 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>,
## #   Category <chr>, Name <chr>, Number <dbl>, Min <dbl>, Max <chr>

# Extract 'Unit' from the 'Data Item' column (substring after "MEASURED")
strawberry <- strawberry %>%
  mutate(Unit = str_extract(strawberry$`Data Item`, "(?<=MEASURED ).*"))

# Extract 'Type' by identifying either "BEARING" or "ORGANIC" in the 'Data Item' column
strawberry <- strawberry %>%
  mutate(Type = str_extract(strawberry$`Data Item`, "BEARING|ORGANIC"))

# Extract 'Operation' by removing 'MEASURED', 'BEARING', and 'ORGANIC'
strawberry <- strawberry %>%
  mutate(Operation = str_replace_all(strawberry$`Data Item`, "MEASURED.*|BEARING|ORGANIC", "") %>%
    str_trim())

# Further clean 'Operation' by removing additional terms ('STRAWBERRIES', commas, hyphens)
strawberry <- strawberry %>%
  mutate(Operation = str_replace_all(strawberry$`Data Item`, "MEASURED.*|BEARING|ORGANIC|STRAWBERRIES(,|,|,)",
    str_replace_all("[-,]", "") %>%
    str_trim())

# View the resulting data
head(strawberry)

## # A tibble: 6 x 29
##   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
## # i 22 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>,
## # Category <chr>, Name <chr>, Number <dbl>, Min <dbl>, Max <chr>, Unit <chr>,
## # Type <chr>, Operation <chr>
```

```
# Export the cleaned data to a CSV file
write.csv(strawberry, "cleaned_strawberries.csv", row.names = FALSE)
```

```
# Check the structure of the cleaned dataset
str(strawberry)
```

```
## tibble [12,669 x 29] (S3: tbl_df/tbl/data.frame)
## $ Program      : chr [1:12669] "CENSUS" "CENSUS" "CENSUS" "CENSUS" ...
## $ Year         : num [1:12669] 2022 2022 2022 2022 2022 ...
## $ Period      : chr [1:12669] "YEAR" "YEAR" "YEAR" "YEAR" ...
## $ Week Ending  : logi [1:12669] NA NA NA NA NA NA ...
## $ Geo Level    : chr [1:12669] "COUNTY" "COUNTY" "COUNTY" "COUNTY" ...
## $ State       : chr [1:12669] "ALABAMA" "ALABAMA" "ALABAMA" "ALABAMA" ...
## $ State ANSI   : chr [1:12669] "01" "01" "01" "01" ...
## $ Ag District  : chr [1:12669] "BLACK BELT" "BLACK BELT" "BLACK BELT" "BLACK BELT" ...
## $ Ag District Code: num [1:12669] 40 40 40 40 40 40 40 40 40 40 ...
## $ County       : chr [1:12669] "BULLOCK" "BULLOCK" "BULLOCK" "BULLOCK" ...
## $ County ANSI  : chr [1:12669] "011" "011" "011" "011" ...
## $ Zip Code     : logi [1:12669] NA NA NA NA NA NA ...
## $ Region       : logi [1:12669] NA NA NA NA NA NA ...
## $ watershed_code : chr [1:12669] "00000000" "00000000" "00000000" "00000000" ...
## $ Watershed    : logi [1:12669] NA NA NA NA NA NA ...
## $ Commodity    : chr [1:12669] "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" "STRAWBERRIES" ...
## $ Data Item    : chr [1:12669] "STRAWBERRIES - ACRES BEARING" "STRAWBERRIES - ACRES GROWN" "STRAWBERRIES - ACRES GROWN" ...
## $ Domain       : chr [1:12669] "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
## $ Domain Category: chr [1:12669] "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" "NOT SPECIFIED" ...
## $ Value        : chr [1:12669] NA "3" NA "1" ...
## $ CV (%)       : chr [1:12669] NA "15.7" NA "(L)" ...
## $ Category     : chr [1:12669] NA NA NA NA ...
## $ Name         : chr [1:12669] NA NA NA NA ...
## $ Number       : num [1:12669] NA NA NA NA NA NA NA NA NA NA ...
## $ Min          : num [1:12669] NA NA NA NA NA NA NA NA NA NA ...
## $ Max          : chr [1:12669] NA NA NA NA ...
## $ Unit         : chr [1:12669] NA NA NA NA ...
## $ Type         : chr [1:12669] "BEARING" NA "BEARING" "BEARING" ...
## $ Operation    : chr [1:12669] "ACRES" "ACRES GROWN" "ACRES NON" "OPERATIONS WITH AREA" ...
```

```
# Convert 'Value' to numeric, removing non-numeric characters
strawberry$Value <- as.numeric(gsub("[^0-9.]", "", strawberry$Value))
```

```
# Convert 'CV (%)' to numeric, removing non-numeric characters (including %, parentheses)
strawberry$`CV (%)` <- as.numeric(gsub("[^0-9.]", "", strawberry$`CV (%)`))
```

```
# Check if conversion was successful
str(strawberry$Value)
```

```
## num [1:12669] NA 3 NA 1 6 5 NA NA 2 2 ...
```

```

str(strawberry$`CV (%)`)

## num [1:12669] NA 15.7 NA NA 52.7 47.6 NA NA 55.7 52.7 ...
# Check for any NAs introduced after conversion
sum(is.na(strawberry$Value))

## [1] 4744

sum(is.na(strawberry$`CV (%)`))

## [1] 7934

# Summary statistics for 'Value' and 'CV (%)'
summary(strawberry$Value)

##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.    NA's
## 0.000e+00 2.000e+00 4.000e+00 1.123e+07 2.100e+01 3.584e+09 4744

summary(strawberry$`CV (%)`)

##      Min. 1st Qu.  Median     Mean 3rd Qu.     Max.    NA's
##      0.60  29.50   41.60   43.43  56.10   99.90   7934

# Histogram for 'Value'
ggplot(strawberry, aes(x = Value)) +
  geom_histogram(binwidth = 10, col = "lightblue", fill = "lightblue") +
  labs(title = "Distribution of Value", x = "Value", y = "Frequency")

## Warning: Removed 4744 rows containing non-finite values (`stat_bin()`).
## Warning: Computation failed in `stat_bin()`
## Caused by error in `bin_breaks_width()`:
## ! The number of histogram bins must be less than 1,000,000.
## i Did you make `binwidth` too small?

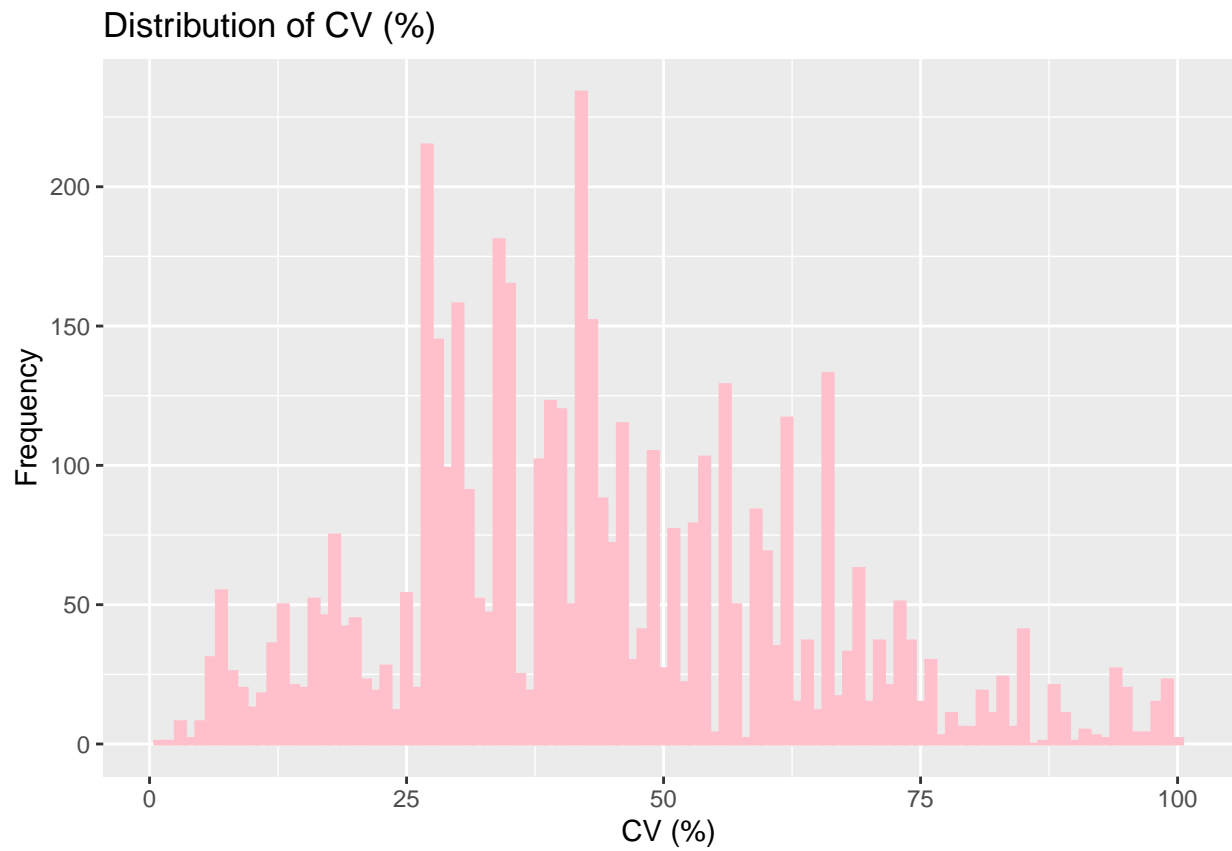
```

Distribution of Value



```
# Histogram for 'CV (%)'  
ggplot(strawberry, aes(x = `CV (%)`)) +  
  geom_histogram(binwidth = 1, col = "pink", fill = "pink") +  
  labs(title = "Distribution of CV (%)", x = "CV (%)", y = "Frequency")
```

```
## Warning: Removed 7934 rows containing non-finite values (`stat_bin()`).
```



```
# Bar plot for 'Type' column  
ggplot(strawberry, aes(x=Type)) +  
  geom_bar(fill="orange") +  
  theme(axis.text.x = element_text(angle=45, hjust=1)) +  
  labs(title="Distribution of Type")
```