Strawberry

Zhenwei Weng

2024-10-11

```r
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(stringr)
straw <- read.csv("strawberries25_v3.csv", header = TRUE)
head(straw)
```

```
##   Program Year Period Week.Ending Geo.Level   State State.ANSI Ag.District
## 1  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
## 2  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
## 3  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
## 4  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
## 5  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
## 6  CENSUS 2022   YEAR          NA    COUNTY ALABAMA          1  BLACK BELT
##   Ag.District.Code  County County.ANSI Zip.Code Region watershed_code Watershed
## 1               40 BULLOCK          11       NA     NA              0        NA
## 2               40 BULLOCK          11       NA     NA              0        NA
## 3               40 BULLOCK          11       NA     NA              0        NA
## 4               40 BULLOCK          11       NA     NA              0        NA
## 5               40 BULLOCK          11       NA     NA              0        NA
## 6               40 BULLOCK          11       NA     NA              0        NA
##      Commodity                                       Data.Item Domain
## 1 STRAWBERRIES                    STRAWBERRIES - ACRES BEARING  TOTAL
## 2 STRAWBERRIES                      STRAWBERRIES - ACRES GROWN  TOTAL
## 3 STRAWBERRIES                STRAWBERRIES - ACRES NON-BEARING  TOTAL
## 4 STRAWBERRIES     STRAWBERRIES - OPERATIONS WITH AREA BEARING  TOTAL
## 5 STRAWBERRIES       STRAWBERRIES - OPERATIONS WITH AREA GROWN  TOTAL
## 6 STRAWBERRIES STRAWBERRIES - OPERATIONS WITH AREA NON-BEARING  TOTAL
##   Domain.Category Value CV....
## 1   NOT SPECIFIED   (D)     (D)
## 2   NOT SPECIFIED     3    15.7
## 3   NOT SPECIFIED   (D)     (D)
## 4   NOT SPECIFIED     1     (L)
## 5   NOT SPECIFIED     6    52.7
## 6   NOT SPECIFIED     5    47.6
```

```
#Loads R packages for data manipulation, reads the "strawberries25_v3.csv" file into 'straw'.
```

```
drop_col <- function(df) {
  df %>% select_if(~ length(unique(.)) > 1)
}
straw_clean <- drop_col(straw)
state <- straw_clean %>%
  group_by(State) %>%
  count()
count(state)
```

```
## # A tibble: 52 x 2
## # Groups:   State [52]
##    State           n
##    <chr>       <int>
##  1 ALABAMA         1
##  2 ALASKA          1
##  3 ARIZONA         1
##  4 ARKANSAS        1
##  5 CALIFORNIA      1
##  6 COLORADO        1
##  7 CONNECTICUT     1
##  8 DELAWARE        1
##  9 FLORIDA         1
## 10 GEORGIA         1
## # i 42 more rows
```

```
sum(state$n) == dim(straw_clean)[1]
```

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

```
#I defines a function 'drop_col' to remove columns in a dataframe that have only one unique value.
#Then applies this function to 'straw' to create a cleaned dataframe 'straw_clean'.
#After that, I group 'straw_clean' by the 'State' column and counts each
#group, storing the results in 'state'.
#Finally, I counts the groups in 'state' and verifies if the total count in 'state'
#equals the number of rows in 'straw_clean'.
```

```
summary <- straw_clean %>%
  group_by(State) %>%
  summarize(count = n())
print(summary)
```

```
## # A tibble: 52 x 2
##    State       count
##    <chr>       <int>
##  1 ALABAMA       154
##  2 ALASKA         41
##  3 ARIZONA        47
##  4 ARKANSAS      120
##  5 CALIFORNIA   2575
##  6 COLORADO      105
##  7 CONNECTICUT    70
##  8 DELAWARE       22
##  9 FLORIDA      1569
## 10 GEORGIA       284
## # i 42 more rows
```

```
California_census <- straw_clean %>%
  filter(State == "CALIFORNIA", Program == "CENSUS") %>%
  select(Year, `Data.Item`, Value)
head(California_census)
```

```
##   Year                                   Data.Item Value
## 1 2022                STRAWBERRIES - ACRES BEARING   (D)
## 2 2022                  STRAWBERRIES - ACRES GROWN   (D)
## 3 2022 STRAWBERRIES - OPERATIONS WITH AREA BEARING     3
## 4 2022   STRAWBERRIES - OPERATIONS WITH AREA GROWN     3
## 5 2022                STRAWBERRIES - ACRES BEARING   (D)
## 6 2022                  STRAWBERRIES - ACRES GROWN   (D)
```

```
California_survey <- straw_clean %>%
  filter(State == "CALIFORNIA", Program == "SURVEY") %>%
  select(Year, Period, `Data.Item`, Value)


#First, I group the 'straw_clean' dataframe by 'State' and
#calculate the count of records per group, storing the results in 'summary'.
#Then, I filter records where 'State' is "California"
#and 'Program' is "CENSUS", selecting the 'Year', 'Data.Item', and 'Value' columns,
#with the results stored in 'California_census'.
#Lastly, I similarly filter records for "California" under the 'Program' "SURVEY",
#selecting the 'Year', 'Period', 'Data.Item',
#and 'Value' columns, and store the results in 'California_survey'.
```

```
process_line <- function(line) {
  line <- as.character(line)
  line <- gsub("[---]", "-", line)
  parts <- unlist(strsplit(line, " - "))
  fruit <- "Strawberries"
  if (length(parts) == 2) {
    item_metric <- unlist(strsplit(parts[2], ","))
    category <- trimws(gsub("^STRAWBERRIES,? ?", "", parts[1]))
    if (category == "") {
      category <- NA
    }
    item <- trimws(ifelse(length(item_metric) > 0, item_metric[1], "N/A"))
```

```r
    metric <- trimws(ifelse(length(item_metric) > 1, item_metric[2], "N/A"))
  } else if (length(parts) == 3) {
    category <- trimws(gsub("^STRAWBERRIES,? ?", "", parts[2]))
    if (category == "") {
      category <- NA
    }
    item_metric <- unlist(strsplit(parts[3], ","))
    item <- trimws(ifelse(length(item_metric) > 0, item_metric[1], "N/A"))
    metric <- trimws(ifelse(length(item_metric) > 1, item_metric[2], "N/A"))
  } else {
    category <- trimws(gsub("^STRAWBERRIES,? ?", "", parts[1]))
    if (category == "") {
      category <- NA
    }
    item <- "N/A"
    metric <- "N/A"
  }
  return(list(Fruit = fruit, Category = category, Item = item, Metric = metric))
}

straw_clean <- cbind(straw_clean, do.call(rbind, lapply(straw_clean$Data.Item, function(x) {
  as.data.frame(process_line(x), stringsAsFactors = FALSE)
})))


#I defined a function called 'process_line' to process strings, primarily to parse
#data items and metrics related to strawberries. The function starts by converting
#the input line into a character string, then replaces various types of dashes with
#a standard dash using regular expressions. It then splits the string by " - ", extracting
#categories, items, and metrics. Different processes are applied based on the number
#of parts split to ensure correct information extraction. Finally, this information
#is organized into a list and returned. Afterwards, I use 'lapply' to apply the
#'process_line' function to each item in 'straw_clean$Data.Item', combine the results
#into data frames, and merge them back with the original dataframe 'straw_clean.'
```

```r
dom_cate <- straw_clean %>%
  group_by(Domain.Category) %>%
  count()
count(dom_cate)
```

```
## # A tibble: 191 x 2
## # Groups:   Domain.Category [191]
##    Domain.Category                                                         n
##    <chr>                                                               <int>
##  1 AREA GROWN: (0.1 TO 0.9 ACRES)                                          1
##  2 AREA GROWN: (1.0 TO 4.9 ACRES)                                          1
##  3 AREA GROWN: (100 OR MORE ACRES)                                         1
##  4 AREA GROWN: (15.0 TO 24.9 ACRES)                                        1
##  5 AREA GROWN: (25.0 TO 49.9 ACRES)                                        1
##  6 AREA GROWN: (5.0 TO 14.9 ACRES)                                         1
##  7 AREA GROWN: (50.0 TO 99.9 ACRES)                                        1
##  8 CHEMICAL, FUNGICIDE: (AZOXYSTROBIN = 128810)                            1
##  9 CHEMICAL, FUNGICIDE: (BACILLUS AMYLOLIQUEFAC F727 = 16489)             1
## 10 CHEMICAL, FUNGICIDE: (BACILLUS AMYLOLIQUEFACIENS MBI 600 = 129082)     1
## # i 181 more rows
```

```
straw_clean <- straw_clean %>%
  separate_wider_delim(cols = `Domain.Category`, delim = ": ",
                       names = c("use", "details"),
                       too_many = "error", too_few = "align_start") %>%
   mutate(
    name = str_extract(details, "(?<=\\().*?(?=\\=)"),
    code = str_extract(details, "(?<=\\= ).*?(?=\\))")
  )
straw_clean$use <- gsub("^CHEMICAL, ", "", straw_clean$use)
straw_clean$Value <- as.numeric(as.character(straw_clean$Value))
```

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

```
straw_clean$CV.... <- as.numeric(as.character(straw_clean$CV....))
```

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

```
straw_clean <- straw_clean %>%
  select(-Data.Item)
head(straw_clean)
```

```
## # A tibble: 6 x 21
##   Program  Year Period Geo.Level State   State.ANSI Ag.District Ag.District.Code
##   <chr>   <int> <chr>  <chr>     <chr>        <int> <chr>                  <int>
## 1 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## 2 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## 3 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## 4 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## 5 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## 6 CENSUS   2022 YEAR   COUNTY    ALABAMA          1 BLACK BELT                40
## # i 13 more variables: County <chr>, County.ANSI <int>, Domain <chr>,
## #   use <chr>, details <chr>, Value <dbl>, CV.... <dbl>, Fruit <chr>,
## #   Category <chr>, Item <chr>, Metric <chr>, name <chr>, code <chr>
```

```
#I grouped and counted 'straw_clean' by 'Domain.Category', then split this column
#into 'use' and 'details', extracting to create new 'name' and 'code' columns. I
#also cleaned the prefix from the 'use' column and converted 'Value' and 'CV....'
#to numeric types. Finally, I removed the 'Data.Item' column.
#To here, I've already finish cleaning the data.
```