Untitled

Zihao Zhang

2024-10-07

```
library(knitr)
library(kableExtra)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
strawberry <- read_csv("C:/Users/Owner/Downloads/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.
```

```
glimpse(strawberry)
```

```
## Rows: 12,669
## Columns: 21
## $ Program            <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "~
## $ Year               <dbl> 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 202~
## $ Period             <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YE~
## $ `Week Ending`      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ `Geo Level`        <chr> "COUNTY", "COUNTY", "COUNTY", "COUNTY", "COUNTY", "~
## $ State              <chr> "ALABAMA", "ALABAMA", "ALABAMA", "ALABAMA", "ALABAM~
## $ `State ANSI`       <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01~
## $ `Ag District`      <chr> "BLACK BELT", "BLACK BELT", "BLACK BELT", "BLACK BE~
```

```
## $ `Ag District Code` <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,~
## $ County             <chr> "BULLOCK", "BULLOCK", "BULLOCK", "BULLOCK", "BULLOC~
## $ `County ANSI`      <chr> "011", "011", "011", "011", "011", "011", "101", "1~
## $ `Zip Code`         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Region             <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ watershed_code     <chr> "00000000", "00000000", "00000000", "00000000", "00~
## $ Watershed          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Commodity          <chr> "STRAWBERRIES", "STRAWBERRIES", "STRAWBERRIES", "ST~
## $ `Data Item`        <chr> "STRAWBERRIES - ACRES BEARING", "STRAWBERRIES - ACR~
## $ Domain             <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL~
## $ `Domain Category`  <chr> "NOT SPECIFIED", "NOT SPECIFIED", "NOT SPECIFIED", ~
## $ Value              <chr> "(D)", "3", "(D)", "1", "6", "5", "(D)", "(D)", "2"~
## $ `CV (%)`           <chr> "(D)", "15.7", "(D)", "(L)", "52.7", "47.6", "(D)",~
```

```
## 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."
```

```
## rm(state_all, state_all1)
```

```
drop_one_value_col <- function(df){   ## takes whole dataframe
drop <- NULL

## test each column for a single value
for(i in 1:dim(df)[2]){
if((df |> distinct(df[,i]) |> count()) == 1){
drop = c(drop, i)
} }

## report the result -- names of columns dropped
## consider using the column content for labels
## or headers

if(is.null(drop)){return("none")}else{

   print("Columns dropped:")
   print(colnames(df)[drop])
   strawberry <- df[, -1*drop]
   }
}


## use the function

strawberry <- drop_one_value_col(strawberry)
```

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

```
drop_one_value_col(strawberry)
```

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

```
calif <- strawberry |>  filter(State=="CALIFORNIA")

## look at the unique values in the "Program" column

## in the consol
## unique(calif$Program)

## and look at the data selection widget on
##      https://quickstats.nass.usda.gov

## You can see that CENSUS AND SURVEY are the two sources
## of data. (Why? What's the differences?).  So, let's see
## they differ.

calif_census <- calif |> filter(Program=="CENSUS")

calif_survey  <- calif |>  filter(Program=="SURVEY")

###


##calif_survey <- strawberry |> select(Year, Period, `Data Item`, Value)
```

```
## no assignment -- just exploring

drop_one_value_col(calif_census)
```

```
## [1] "Columns dropped:"
## [1] "Program"    "Period"     "State"      "State ANSI"
```

```
drop_one_value_col(calif_survey)
```

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

```
#|label: split Data Item
# Replace '-' (hyphen with spaces) with a comma.
strawberry <- strawberry |>
  mutate(`Data Item` = str_replace_all(`Data Item`, "- ", ","))

# Split 'Data Item' into 4 columns
```

```
strawberry <- strawberry |>
  separate_wider_delim( cols = 'Data Item',
                        delim = ",",
                        names = c("Fruit", "Category", "Item", "Metric"),
                        too_many = "merge",
                        too_few = "align_start")

# Remove 'measured in' to metric columns
strawberry <- strawberry |>
  mutate(Metric = ifelse(grepl("MEASURED IN", Item), Item, Metric),
         Item = ifelse(grepl("MEASURED IN", Item), NA, Item))

# Remove 'production' to its correct way.
strawberry <- strawberry |>
  mutate(
    Item = ifelse(grepl("PRODUCTION", Metric), "PRODUCTION", Item),
    Metric = ifelse(grepl("PRODUCTION", Metric), sub("PRODUCTION", "", Metric), Metric)
  )

# Remove 'utilized' from category to Item
strawberry <- strawberry |>
  mutate(
    Item = ifelse(grepl("UTILIZED", Category, ignore.case = TRUE),
                  paste("UTILIZED", Item, sep = " "),
                  Item),
    Category = ifelse(grepl("UTILIZED", Category, ignore.case = TRUE), NA, Category)
  )

# Consider a better way to move items in one step.
movingitem <- c("ACRES BEARING", "ACRES NON-BEARING", "ACRES GROWN", "YIELD",
                "ACRES HARVESTED", "ACRES PLANTED", "OPERATIONS WITH AREA BEARING",
                "OPERATIONS WITH AREA GROWN", "OPERATIONS WITH AREA NON-BEARING",
                "PRODUCTION")

# Move terms from 'Metric' or 'Category' to 'Item' without replacing 'Metric' data
strawberry <- strawberry |>
  mutate(Item = ifelse(grepl(paste(movingitem, collapse = "|"), Category, ignore.case = TRUE) & is.na(I
                       ifelse(grepl(paste(movingitem, collapse = "|"), Category, ignore.case = TRUE),
                              paste(Item, Category, sep = ", "), Item)),
         Category = ifelse(grepl(paste(movingitem, collapse = "|"), Category, ignore.case = TRUE), NA, C




## Use too_many and too_few to set up the separation operation.
```

```
#|label: fix the leading space

 # note
strawberry$Category[1]
```

```
## [1] NA
```

```
strawberry$Item[2]
```

```
## [1] "ACRES GROWN"
```

```
strawberry$Metric[6]
```

```
## [1] NA
```

```
strawberry$Domain[1]
```

```
## [1] "TOTAL"
```

```
##
## trim white space

strawberry$Category <- str_trim(strawberry$Category, side = "both")
strawberry$Item <- str_trim(strawberry$Item, side = "both")
strawberry$Metric <- str_trim(strawberry$Metric, side = "both")
```

```
# Split the Domain column into multiple categories
strawberry <- strawberry |>
separate_wider_delim(
cols = Domain,
delim = " , ",
names = c("Area Grown", "Fertilize", "Organic", "Chemical"),
too_many = "merge",
too_few = "align_start"
)
```

```
#Loading variables to each column
strawberry <- strawberry |>
mutate(
Chemical = ifelse(grepl("CHEMICAL", `Area Grown`, ignore.case = TRUE), `Area Grown`, NA),
Organic = ifelse(grepl("ORGANIC", `Area Grown`, ignore.case = TRUE), `Area Grown`, NA),
Fertilize = ifelse(grepl("FERTILIZER", `Area Grown`, ignore.case = TRUE), `Area Grown`, NA),
`Area Grown`= ifelse(grepl("CHEMICAL|ORGANIC|FERTILIZER", `Area Grown`, ignore.case = TRUE), NA, `Area G
)
#Dealing with 'Domain Category' column
strawberry <- strawberry |>
mutate(
Chemical = ifelse(grepl("CHEMICAL", `Domain Category`, ignore.case = TRUE),
`Domain Category`,
Chemical),
Organic = ifelse(grepl("ORGANIC", `Domain Category`, ignore.case = TRUE),
`Domain Category`,
Organic),
Fertilize = ifelse(grepl("FERTILIZER", `Domain Category`, ignore.case = TRUE),
`Domain Category`,
Fertilize),
`Area Grown` = ifelse(grepl("AREA", `Domain Category`, ignore.case = TRUE),
`Domain Category`,
```

```
`Area Grown`),
`Domain Category` = ifelse(grepl("CHEMICAL|ORGANIC|FERTILIZER|AREA", `Domain Category`, ignore.case = TR
)
#Move 'Total' to its best place
strawberry <- strawberry |>
mutate(Item = ifelse(grepl("Total", `Area Grown`, ignore.case = TRUE),
paste("Total", Item, sep = " "),
Item),
`Area Grown` = ifelse(grepl("Total", `Area Grown`, ignore.case = TRUE), NA, `Area Grown`)
)
```

```
strawberry <- strawberry |>
mutate(Chemical = str_replace_all(Chemical, "[,:=()]", ","))

#Split it into three columns
strawberrynew<- strawberry |>
separate_wider_delim(
cols = Chemical,
delim = ",",
names = c("Type", "Ingredient", "Code"), #Separate Chemical into type, ingredient, and code.
too_many = "merge",
too_few = "align_start"
)
#Filling in the columns
strawberrynew<- strawberrynew |>
mutate(
Type = ifelse(Type == "CHEMICAL" | is.na(Type), Ingredient, Type),
Ingredient = ifelse(!is.na(Ingredient), str_extract(Code, "\\b[A-Za-z\\-\\.\\s]+\\b"), Ingredient), #"\
Code = str_replace(Code, "\\b[A-Za-z\\-\\.\\s]+\\b", "")
)
#Clean 'Code' Column
strawberrynew <- strawberrynew |>
mutate(
Code = str_replace_all(Code, "^\\s*,+|,+\\s*$|\\s*,\\s*,+", ""),
Code = str_trim(Code)
)
head(strawberrynew)
```

```
## # A tibble: 6 x 23
##   Program  Year Period `Geo Level` State   `State ANSI` `Ag District`
##   <chr>   <dbl> <chr>  <chr>       <chr>   <chr>        <chr>
## 1 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## 2 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## 3 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## 4 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## 5 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## 6 CENSUS   2022 YEAR   COUNTY      ALABAMA 01           BLACK BELT
## # i 16 more variables: `Ag District Code` <dbl>, County <chr>,
## #   `County ANSI` <chr>, Fruit <chr>, Category <chr>, Item <chr>, Metric <chr>,
## #   `Area Grown` <chr>, Fertilize <chr>, Organic <chr>, Type <chr>,
## #   Ingredient <chr>, Code <chr>, `Domain Category` <chr>, Value <chr>,
## #   `CV (%)` <chr>
```