Stawberries

MA615

2024-09-25

Preparing data for analysis

Introduction: foundations

Before we begin to work with the strawberry data, let's talk about how we will approach the work.

Data cleaning and organization

Cleaning and organizing data for analysis is an essential skill for data scientists. Serious data analyses must be presented with the data on which the results depend. The credibility of data analysis and modelling depends on the care taken in data preparation and organization.

References

In their handbook "An introduction to data cleaning with R" by Edwin de Jonge and Mark van der Loo, de Jonge and van der Loo go into detail about specific data cleaning isssues and how to handle them in R.

"Problems, Methods, and Challenges in Comprehensive Data Cleansing" by Heiko Müller and Johann-Christoph Freytag is a good companion to the de Jonge and van der Loo handbook, offering additional issues in their discussion.

Attitudes

Mechanistic descriptions of data cleaning methods are insufficient.

Data is the product (or by-product) of purposeful human activity

Much of the data used in analysis accessed on local databases or online which may create the impression that the data have been carefully curated. Beware. Data are produced by people for a purpose, with a point-of-view, and at a time and location that may affect the data. The provenance and lineage of the data are meta data you should include when reporting analysis. Data collection is purposeful human activity with all of the risks and weaknesses that are part of any purposeful human activity.

Data is language

Data has meaning. Data can be included in sentences related to the meaning of the data. Cleaning and organizing data should be informed by the meaning the data convey and how that meaning relates to the research you are doing do achieve this important result.

- Immerse yourself in the data. Put data into context.
- Visualize the data to find problems, confirm your understandings, and plan your data organization. People do a bad job of seeing meaningful patterns in data but a good job of seeing patterns of all kinds when data are rendered as plots. As you product and show visualizations, ask your self and those who view your presentations, "what do you see?" and "what do you wonder?"

Example: Strawberries

Public information

WHO says strawberries may not be so safe for you–2017March16

Pesticides + poison gases = cheap, year-round strawberries 2019March20

Multistate Outbreak of Hepatitis A Virus Infections Linked to Fresh Organic Strawberries-2022March5

Strawberry makes list of cancer-fighting foods-2023May31

What is the question?

- Where they are grown? By whom?
- Are they really loaded with carcinogenic poisons?
- Are they really good for your health? Bad for your health?
- Are organic strawberries carriers of deadly diseases?

- When I go to the market should I buy conventional or organic strawberries?

The data

The data set for this assignment has been selected from: [USDA_NASS_strawb_2024SEP25
The data have been stored on NASS here: USDA_NASS_strawb_2024SEP25

and has been stored on the blackboard as strawberries25_v3.csv.

USDA NASS

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

Read the file

```
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.
```

```
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)",~
```

Examine the data. How is it organized?

```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

sum(state_all1$n) == dim(strawberry)[1]
```

```
[1] TRUE
```

```r
## to get an idea of the data -- looking at california only

calif_census <- strawberry |> filter((State=="CALIFORNIA") & (Program=="CENSUS"))

calif_census <- calif_census |> select(Year, `Data Item`, Value)

###
```

```r
calif_survey <- strawberry |> filter((State=="CALIFORNIA") & (Program=="SURVEY"))

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

Remove columns with a single value in all columns and county in `Geo Level`

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

drop_one_value_col <- function(df){
drop <- NULL
for(i in 1:dim(df)[2]){
if((df |> distinct(df[,i]) |> count()) == 1){
drop = c(drop, i)
} }

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"
```

```r
drop_one_value_col(strawberry)
```

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

```r
strawberry <- strawberry |>
  filter(`Geo Level` == "NATIONAL" | `Geo Level` == "STATE")
```

separate composite columns

`Data Item` into (fruit, category, item)

```
#|label: split Data Item

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

Split `Fruit` by `-` in order to remove `strawberry` from each cell.

```
strawberry <- strawberry |>
  separate(col = `Fruit`,
           into = c(NA, "Fruit"),
           sep = "-",
           extra = "merge",
           fill = "right")
unique(strawberry$Fruit)
```

```
 [1] " ACRES BEARING"                       " ACRES GROWN"
 [3] " ACRES NON-BEARING"                   " OPERATIONS WITH AREA BEARING"
 [5] " OPERATIONS WITH AREA GROWN"          " OPERATIONS WITH AREA NON-BEARING"
 [7] NA                                     " PRICE RECEIVED"
 [9] " ACRES HARVESTED"                     " ACRES PLANTED"
[11] " PRODUCTION"                          " YIELD"
[13] " APPLICATIONS"                        " TREATED"
```

Split `Domain Category` into three columns accordingly to `Chemical Name`, `Use` and `Code`. Removed additional signs such as `()`

```
strawberry <- strawberry |>
  separate_wider_delim( cols = 'Domain Category', delim = ":", names = c("Use", "Chem&Number"

strawberry <- strawberry |>
  separate_wider_delim( cols = 'Chem&Number', delim = "=", names = c("Chemical Name", "Chem&N

strawberry <- strawberry |>
  mutate(
    `Chemical Name` = gsub("[()]", "", `Chemical Name`),
    Code = gsub("\\)", "", `Chem&Number`)
  )

strawberry <- strawberry|> select(-`Chem&Number`)

strawberry <- strawberry |>
  relocate(Code, .after = `Chemical Name`) |>
  relocate(Use, .after = Code)
```

There is a problem you have to fix – a leading space.

```
#|label: fix the leading space

strawberry$Category[1]
```

```
[1] NA
```

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

sum(is.na(strawberry$Metric))
```

```
[1] 2393
```

Fixing Metric as some of its values were found in `Category` and `Item` Columns.

Created a New Column `Bearing Type` and filled it with adequate values from other columns.

```r
strawberry <- strawberry |>
  mutate(Metric = coalesce( if_else(str_detect(Category, "MEASURED"), Category,Metric))) |>
  mutate(Category = if_else(str_detect(Category, "MEASURED"), NA_character_, Category))
strawberry <- strawberry |>
  mutate(Metric = coalesce( if_else(str_detect(Item, "MEASURED"), Item,Metric))) |>
  mutate(Item = if_else(str_detect(Item, "MEASURED"), NA_character_, Item))

strawberry$`Bearing Type` <- NA
strawberry <- strawberry |>
  relocate(`Bearing Type`, .after = Fruit)

strawberry <- strawberry |>
  mutate(`Bearing Type` = coalesce(if_else(str_detect(Category, "BEARING"), "BEARING", `Bear
  mutate( Category = str_remove_all(Category, "BEARING")) |>
  mutate(`Bearing Type` = coalesce(if_else(str_detect(Fruit, "NON-BEARING"), "NON-BEARING",
  mutate( Fruit = str_remove_all(Fruit, "NON-BEARING")) |>
  mutate(`Bearing Type` = coalesce(if_else(str_detect(Fruit, "BEARING"), "BEARING", `Bearing
  mutate( Fruit = str_remove_all(Fruit, "BEARING"))
```

Created Measure from Fruit and started taking misplaced values from other columns into it. Removing extra words from some columns.

```r
strawberry <- strawberry |> rename(Measure = Fruit)
strawberry <- strawberry |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "UTILIZED - PRODUCTION"), "UTILI
  mutate( Category = str_remove_all(Category, "UTILIZED - PRODUCTION")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Item, "UTILIZED - PRODUCTION"), "UTILIZED -
  mutate( Item = str_remove_all(Item, "UTILIZED - PRODUCTION")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "ACRES HARVESTED"), "ACRES HARVES
  mutate( Category = str_remove_all(Category, "ACRES HARVESTED")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "OPERATIONS WITH SALES"), "OPERAT
  mutate( Category = str_remove_all(Category, "OPERATIONS WITH SALES")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "SALES"), "SALES", NA_character_)
  mutate( Category = str_remove_all(Category, "SALES")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "PRODUCTION"), "PRODUCTION", NA_c
  mutate( Category = str_remove_all(Category, "PRODUCTION")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "OPERATIONS WITH AREA HARVESTED"
  mutate( Category = str_remove_all(Category, "OPERATIONS WITH AREA HARVESTED")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Category, "PRICE RECEIVED"), "PRICE RECEIV
  mutate( Category = str_remove_all(Category, "PRICE RECEIVED")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Item, "OPERATIONS WITH SALES"), "OPERATION
  mutate( Item = str_remove_all(Item, "OPERATIONS WITH SALES")) |>
  mutate(Measure = coalesce(Measure,
                             if_else(str_detect(Item, "SALES"), "SALES", NA_character_)))
  mutate( Item = str_remove_all(Item, "SALES"))

 strawberry <- strawberry |>
   mutate(
     Category = gsub("[-]", "", Category),
    Item = gsub("[-]", "", Item)
   )
strawberry$Category <- str_trim(strawberry$Category, side = "both")
strawberry$Item <- str_trim(strawberry$Item, side = "both")
strawberry$Metric <- str_trim(strawberry$Metric, side = "both")

strawberry <- strawberry |>
  mutate(Category = str_remove_all(Category,"ORGANIC"),
         Item = str_remove_all(Item, "ORGANIC"),
         Use = str_remove_all(Use, "ORGANIC STATUS"),
         Use = str_remove_all(Use, "CHEMICAL,"),
         Domain = str_remove_all(Domain, ", FUNGICIDE"),
         Domain = str_remove_all(Domain, ", INSECTICIDE"),
         Domain = str_remove_all(Domain, ", HERBICIDE"),
         Domain = str_remove_all(Domain, "FERTILIZER"),
         Domain = str_remove_all(Domain, ", OTHER")
         )
```

Taking extra values into “Measure Column” which can help in reducing the NAs in the “Measure” Column.

Combined Category and Item after finding almost same values.

Removing the Blank Space in some of the Columns.

```r
values_to_shift <- c("AVG", "ADJUSTED BASE", "10 YEAR AVG", "10 YEAR AVG FOR PARITY PURPOSES"

strawberry <- strawberry |>
  mutate(
    Measure = ifelse(Item %in% values_to_shift,
                     paste(Measure, Item , sep = "-"),
                     Measure),
    Item = ifelse(Item %in% values_to_shift, NA_character_, Item)
  ) |>
  mutate(Applications = str_c(coalesce(Category, ""), coalesce(Item, ""))) |>
  mutate(Domain = na_if(Domain, ""),
         Applications = na_if(Applications, ""),
         Use = na_if(Use, "")) |>
  relocate(Applications, .after = Measure) |>
  select(-Item, -Category)

strawberry$Measure <- str_trim(strawberry$Measure, side = "both")
strawberry$`Chemical Name`<- str_trim(strawberry$`Chemical Name`, side = "both")
strawberry$Code <- str_trim(strawberry$Code, side = "both")
strawberry$Use <- str_trim(strawberry$Use, side = "both")
```

Now Examine the Rest of the Columns

Which ones need to be split?

Split Sales , Chemicals, Organic and Non-Organic into Different Dataframes

(do this last after separating rows into separate data frames) (THEN rename the columns to correspond the analysis being done with the data frames)

```r
#|label: split srawberry into census and survey pieces

census <- strawberry |> filter(Program == "CENSUS")

survey <- strawberry |> filter(Program == "SURVEY")

nrow(strawberry) == (nrow(census) + nrow(survey))
```

```
[1] TRUE
```

```
census <- census |>
  mutate(Certified = if_else(`Chemical Name` == " NOP USDA CERTIFIED", "NOP USDA CERTIFIED",
  rename(`Size-Bracket` = `Chemical Name`) |>
  relocate(Certified, .after = Domain)

## Move marketing-related rows in strw_b_chem
## to strw_b_sales
census_organic <- census |> filter(Domain == "ORGANIC STATUS")
census_non_organic <- census |> filter(Domain != "ORGANIC STATUS")
survey_chemical <- survey |> filter(Domain == "CHEMICAL")
survey_non_chemical <- survey |> filter(Domain != "CHEMICAL")
```

Writing Code into different CSV Files

```
write.csv(strawberry, file = "strawberry_cleaneddata.csv")
write.csv(census, file = "census_data.csv")
write.csv(survey, file = "survey_data.csv")
write.csv(census_organic, file = "census_organic.csv")
write.csv(census_non_organic, file = "census_non_organic.csv")
write.csv(survey_chemical, file = "survey_chemical.csv")
write.csv(survey_non_chemical, file = "survey_non_chemical.csv")
```

epa numbers

Active Pesticide Product Registration Informational Listing

CAS for Methyl Bromide

pesticide chemical search

toxic chemical dashboard

pubChem

The EPA PC (Pesticide Chemical) Code is a unique chemical code number assigned by the EPA to a particular pesticide active ingredient, inert ingredient or mixture of active ingredients.

Investigating toxic pesticides

start here with chem PC code

step 2 to get label (with warnings) for products using the chemical

Pesticide Product and Label System

Search by Chemical

CompTox Chemicals Dashboard

Active Pesticide Product Registration Informational Listing

OSHA chemical database

Pesticide Ingredients

NPIC Product Research Online (NPRO)

Databases for Chemical Information

Pesticide Active Ingredients

TSCA Chemical Substance Inventory

glyphosate