Strawberries_Assignment

Guangyu Ding

2024-10-07

`{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)` Preparing data for analysis

Acquire, explore, clean & structure, EDA

Data cleaning and organization

“An introduction to data cleaning with R” by Edwin de Jonge and Mark van der Loo

“Problems, Methods, and Challenges in Comprehensive Data Cleansing” by Heiko Müller and Johann-Christoph Freytag

Strawberries

Questions

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?

Do Strawberry farmers make money?

How do the strawberries I buy get to my market?

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.

read and explore the data

Set-up

```{r} #| label: load libraries and set options #| warning: false #| message: false #|
```

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

~~~
Read the data and take a first look

```{r}
#| label: read data - glimpse

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

glimpse(strawberry)
~~~

I have 12699 rows and 21 columns.

All I can see from the glimpse is I have date, location, values and coefficients of variation.

Examine the data. How is it organized?

```{r} #| label: explore organization 1

## 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.")}

## rm(state_all, state_all1)

````
remove columns with a single value in all rows

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

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)

drop_one_value_col(strawberry)
```

To get better look at the data, look at California.

```{r} #| label: explore WEST VIRGINIA only

WVIR <- strawberry |> filter(State==“WEST VIRGINIA”)

## 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.

WVIR_census <- WVIR |> filter(Program==“CENSUS”)

WVIR_survey <- WVIR |> filter(Program==“SURVEY”)

```
###

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

~~~
Explore California to understand the census and survey

```{r}
#| label: explore WVIR census and survey

## no assignment -- just exploring

drop_one_value_col(WVIR_census)

drop_one_value_col(WVIR_survey)
~~~

Conclusions from California data exploration.

Now return to the entire data set.

take the lessons learned by examinging the California data

Two strategies – columns first, rows first

Split the census data from the survey data. drop single value columns

separate composite columns

Data Item into (fruit, category, item)

```{r} #|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” )

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

````
There is a problem you have to fix -- a leading space.

```{r}
#|label: fix the leading space

 # note
strawberry$Category[1]
# strawberry$Item[2]
# strawberry$Metric[6]
# strawberry$Domain[1]
##
## 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")
```

now exam the Fruit column – find hidden sub-columns

~~~
```{r}

unique(strawberry$Fruit)
~~~

generate a list of rows with the production and price information

```
spr <- which((strawberry$Fruit=="STRAWBERRIES - PRODUCTION") | (strawberry$Fruit==“STRAWBERRIES - PRICE RECEIVED”))

strw_prod_price <- strawberry |> slice(spr)
```

this has the census data, too

```
strw_chem <- strawberry |> slice(-1*spr) ## too soon
```

~~~

now examine the rest of the columns

Which ones need to be split?

split sales and chemicals into two dataframes

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

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

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

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

nrow(strawberry) == (nrow(strw_b_chem) + nrow(strw_b_sales))

## Move marketing-related rows in strw_b_chem
## to strw_b_sales
~~~

plots

~~~
```{r} #|label: plot 1
~~~

plot1_data <- strawberry |> select(c(Year, State, Category, Value)) |> filter((Year == 2021) & (Category == “ORGANIC - OPERATIONS WITH SALES”))

plot1_data$Value <− as.numeric(plot1_data$Value)

plot1_data <- plot1_data |> arrange(desc(Value))

ggplot(plot1_data, aes(x=reorder(State, -Value), y=Value)) + geom_bar(stat = “identity”) + theme(axis.text.x=element_text(angle=45,hjust=1)) + labs(x = “States”, y = “Count”, title =“Number of Organic Strawberry operations with Sales in 2021”)

~~~
```{r}
## plot 2

plot2_data <- strawberry |>
  select(c(Year, State, Category, Item, Value)) |>
  filter((Year == 2021) &
           (Category == "ORGANIC - SALES") &
           (Item == "MEASURED IN $") &
           (Value != "(D)"))


plot2_data$Value <- as.numeric(gsub(",", "", plot2_data$Value))

plot2_data <- plot1_data |> arrange(desc(Value))

ggplot(plot2_data, aes(x=reorder(State, -Value), y=Value)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x=element_text(angle=45,hjust=1)) +
  labs(x = "States", y = "Sales",
title ="Organic Strawberry Sales ($) in 2021")
~~~

The codes I use here form a summary of the top 10 states with the highest strawberries production. This answer the first question, which asks where are the strawberries growing.

```{r} # Q1 # Create a summary of strawberry production by State and County state_county_summary <- strawberry %>% group_by(State, County) %>% summarise(Count = n()) %>% arrange(desc(Count))

# Display the summary table

print(state_county_summary)

state_summary <- state_county_summary %>% group_by(State) %>% summarise(Count = sum(Count)) %>% arrange(desc(Count))
```

Plot the top 10 states by strawberry production count

top_states <- head(state_summary, 10)

ggplot(top_states, aes(x = reorder(State, Count), y = Count)) + geom_bar(stat = “identity”, fill = “orange”) + coord_flip() + labs(title = “Top 10 States by Strawberry Production”, x = “State”, y = “Production Count”) + theme_minimal()

~~~
The code here provides a list of the chemical data mainly used on straw
berries. We can change the number in top_chemicals variable to see the
name of the chemical and then check whether or not there are carcinogen
ic poisons.

```{r}
# Q2

# Filter rows that contain information about chemicals used
chemical_data <- strawberry %>%
  filter(grepl("CHEMICAL", `Domain Category`)) %>%
  select(State, County, `Domain Category`)

# Display the chemical data
print(head(chemical_data))

# Summarize the use of different chemicals
chemical_summary <- chemical_data %>%
  group_by(`Domain Category`) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))

# Plot the most common chemicals used on strawberries
top_chemicals <- head(chemical_summary, 10)

ggplot(top_chemicals, aes(x = reorder(`Domain Category`, Count), y = Co
unt)) +
  geom_bar(stat = "identity", fill = "red") +
  coord_flip() +
  labs(title = "Top 10 Chemicals Used on Strawberries",
       x = "Chemical Type", y = "Count of Usage") +
  theme_minimal()
~~~

Question 3 asks whether the strawberries are good for our health or not. I think the Domain.Category column in strawberries25 can determine whether or not it is health, because organic fruits do not use pesticides during the cultivation process. Base on the plot, most of strawberries are organic, which are health.

``{r} # Q3 # Filter organic and non-organic data organic_data <- strawberry %>%   filter(grepl("ORGANIC STATUS: \\(NOP USDA CERTIFIED\\)",Domain Category`))

Count the number of organic vs. non-organic data entries

organic_count <- nrow(organic_data) non_organic_count <- nrow(strawberry) - organic_count

Create a data frame for visualization

organic_summary <- data.frame(Type = c(“Organic”, “Non-Organic”), Count = c(organic_count, non_organic_count))

Plot organic vs. non-organic production

ggplot(organic_summary, aes(x = Type, y = Count, fill = Type)) + geom_bar(stat = “identity”) + labs(title = “Comparison of Organic vs. Non-Organic Strawberry Production”, x = “Production Type”, y = “Count”) + theme_minimal()

~~~
There may be some columns relate to both the organic and deadly disease
s to answer question 4, but I can't figure it out. As for question 5, I
 think we should buy organic strawberries since all the status of the o
rganic strawberries are NOP USDA CERTIFIED

```{r}
# Q4
library(dplyr)

# Filter the data to only include rows related to organic strawberries
organic_strawberries <- strawberry %>%
  filter(grepl("ORGANIC STATUS: \\(NOP USDA CERTIFIED\\)", `Domain Cate
gory`))

# Check for diseases or harmful chemicals
diseases_info <- organic_strawberries %>%
  filter(grepl("DISEASE|PATHOGEN", tolower(`Domain Category`)))

# View the results
print(diseases_info)

# Q5
# Filter the data to only include rows related to organic strawberries
organic_strawberries <- strawberry %>%
~~~

```
  filter(grepl("ORGANIC STATUS: \\(NOP USDA CERTIFIED\\)", `Domain Cate
gory`))

# Print the resulting data
print(organic_strawberries)

# Alternatively, you can view the data in a more interactive way using:
View(organic_strawberries)
```

Question 6 asks do strawberry farmers make money. We choose the Data Item column to observe. Based on the number of operations listed in the plot, we can say that strawberry farmers make money in some states of US.

```{r} # Q6 # Filter data related to operations with strawberries operation_data <- strawberries %>% filter(grepl("OPERATIONS", Data.Item))

## Summarize the number of operations

operation_summary <- operation_data %>% group_by(State) %>% summarise(Operations = n()) %>% arrange(desc(Operations))

## Plot the number of operations by state

top_operations <- head(operation_summary, 10)

ggplot(top_operations, aes(x = reorder(State, Operations), y = Operations)) + geom_bar(stat = "identity", fill = "blue") + coord_flip() + labs(title = "Number of Strawberry Farming Operations by State", x = "State", y = "Number of Operations") + theme_minimal()

```

Now what I did is splitting the chemical data into use, name and code.
The new data has three new columns with the chemical data so that these
 data are displayed clearly.

```{r}
# Install and load dplyr
if (!require("dplyr")) install.packages("dplyr")
library(dplyr)

# Add new columns, and the new columns are all the chemical data in the
 Domain Category column.
strawberries <- strawberry %>%
  mutate(
    use = ifelse(grepl("^CHEMICAL, ", `Domain Category`),
                 sub("^CHEMICAL, (.*?):.*", "\\1", `Domain Category`),
                 NA),
    name = ifelse(grepl(": \\((.*) = ", `Domain Category`),
                  sub("^CHEMICAL, .*: \\((.*?) = .*\\)$", "\\1", `Domai
n Category`),
                  NA),
    code = ifelse(grepl("= (\\d+)\\)$", `Domain Category`),
                  sub(".*= (\\d+)\\)$", "\\1", `Domain Category`),
                  NA)
  )

# Display the updated data frame
head(strawberries)
```