Strawberry

Yuhan Pu

2023-10-16

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

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

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(stringr)
library(dplyr)
library(readr)
```

```
strawberry <- read.csv("strawberry.csv")
```

```
strawberry$Value[strawberry$Value==' (D)']<-NA
strawberry$Value[strawberry$Value==' (NA)']<-NA
```

```
strawberry$CV....[strawberry$CV....=='(D)']<-NA
strawberry$CV....[strawberry$CV....=='(H)']<-NA
```

```
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"      "Geo.Level"        "Ag.District"      "Ag.District.Code"
##  [5] "County"           "County.ANSI"      "Zip.Code"         "Region"
##  [9] "watershed_code"   "Watershed"        "Commodity"
```

```
drop_one_value_col(strawberry)
```

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

```
glimpse(strawberry)
```

```
## Rows: 4,314
## Columns: 10
## $ Program         <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CEN~
## $ Year            <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, ~
## $ Period          <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR"~
## $ State           <chr> "ALASKA", "ALASKA", "ALASKA", "ALASKA", "ALASKA", "ALA~
## $ State.ANSI      <int> 2, 2, 2, 2, 2, 2, 2, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 9, ~
## $ Data.Item       <chr> "STRAWBERRIES, ORGANIC - OPERATIONS WITH SALES", "STRA~
## $ Domain          <chr> "ORGANIC STATUS", "ORGANIC STATUS", "ORGANIC STATUS", ~
## $ Domain.Category <chr> "ORGANIC STATUS: (NOP USDA CERTIFIED)", "ORGANIC STATU~
## $ Value           <chr> "2", NA, NA, NA, "2", NA, NA, "142", "1,413,251", "311~
## $ CV....          <chr> NA, NA, NA, NA, NA, NA, NA, "19.2", "51.6", "46.0", "5~
```

```
state_all <- strawberry |> group_by(State) |> count()

if(sum(state_all$n) == dim(strawberry)[1]){print("Every row has value in the State column.")}
```

```
## [1] "Every row has value in the State column."
```

```
strawberry <- strawberry[!is.na(strawberry$Value), ]
strawberry$Value <- gsub(",", "", strawberry$Value)
strawberry$Value <- gsub('"', "" , strawberry$Value)
strawberry$Value<-as.integer(strawberry$Value)
```

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

```
## Warning: NAs introduced by coercion to integer range
```

```
strawberry<-na.omit(strawberry)
```

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

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

## check that all of the rows are accounted for

nrow(strawberry) == (nrow(strwb_census) + nrow(strwb_survey))
```

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

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

## clean up the environment
```

```
strwb_census <- strwb_census |>
  separate_wider_delim(  cols = `Data.Item`,
                         delim = ",",
                         names = c("Fruit",
                                   "temp1",
                                   "temp2",
                                   "temp3"),
                         too_many = "error",
                         too_few = "align_start"
                       )
```

```
strwb_census <- strwb_census |>
  separate_wider_delim(  cols = temp1,
                         delim = " - ",
                         names = c("crop_type",
                                   "prop_acct"),
                         too_many = "error",
                         too_few = "align_start"
                       )
```

```
uni<-unique(strwb_survey$Data.Item)
uni
```

```
##  [1] "STRAWBERRIES - PRICE RECEIVED, MEASURED IN $ / CWT"
##  [2] "STRAWBERRIES - PRODUCTION, MEASURED IN CWT"
##  [3] "STRAWBERRIES, NOT SOLD - PRODUCTION, MEASURED IN CWT"
##  [4] "STRAWBERRIES, UTILIZED - PRODUCTION, MEASURED IN CWT"
##  [5] "STRAWBERRIES - PRODUCTION, MEASURED IN $"
##  [6] "STRAWBERRIES - PRODUCTION, MEASURED IN TONS"
##  [7] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB"
##  [8] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB / ACRE / APPLICATION, AVG"
##  [9] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN LB / ACRE / YEAR, AVG"
```

```
## [10] "STRAWBERRIES, BEARING - APPLICATIONS, MEASURED IN NUMBER, AVG"
## [11] "STRAWBERRIES, UTILIZED - PRODUCTION, MEASURED IN TONS"
## [12] "STRAWBERRIES, FRESH MARKET - PRICE RECEIVED, MEASURED IN $ / CWT"
## [13] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, MEASURED IN $ / CWT"
## [14] "STRAWBERRIES, PROCESSING - PRICE RECEIVED, MEASURED IN $ / TON"
## [15] "STRAWBERRIES, FRESH MARKET - PRODUCTION, MEASURED IN $"
## [16] "STRAWBERRIES, FRESH MARKET, UTILIZED - PRODUCTION, MEASURED IN CWT"
## [17] "STRAWBERRIES, PROCESSING - PRODUCTION, MEASURED IN $"
## [18] "STRAWBERRIES, PROCESSING, UTILIZED - PRODUCTION, MEASURED IN CWT"
## [19] "STRAWBERRIES, PROCESSING, UTILIZED - PRODUCTION, MEASURED IN TONS"
## [20] "STRAWBERRIES, PROCESSING - PRODUCTION, MEASURED IN CWT"
```

```
glimpse(strwb_census)
```

```
## Rows: 556
## Columns: 14
## $ Program         <chr> "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CENSUS", "CEN~
## $ Year            <int> 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, 2021, ~
## $ Period          <chr> "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR", "YEAR"~
## $ State           <chr> "CALIFORNIA", "CALIFORNIA", "CALIFORNIA", "CALIFORNIA"~
## $ State.ANSI      <int> 6, 6, 6, 6, 6, 6, 6, 9, 9, 12, 12, 13, 13, 13, 13, 13,~
## $ Fruit           <chr> "STRAWBERRIES", "STRAWBERRIES", "STRAWBERRIES", "STRAW~
## $ crop_type       <chr> " ORGANIC", " ORGANIC", " ORGANIC", " ORGANIC", " ORGA~
## $ prop_acct       <chr> "OPERATIONS WITH SALES", "PRODUCTION", "SALES", "SALES~
## $ temp2           <chr> NA, " MEASURED IN CWT", " MEASURED IN $", " MEASURED I~
## $ temp3           <chr> NA, NA, NA, NA, NA, " MEASURED IN CWT", NA, NA, NA, NA~
## $ Domain          <chr> "ORGANIC STATUS", "ORGANIC STATUS", "ORGANIC STATUS", ~
## $ Domain.Category <chr> "ORGANIC STATUS: (NOP USDA CERTIFIED)", "ORGANIC STATU~
## $ Value           <int> 142, 1413251, 311784980, 1412627, 141, 1401384, 7, 8, ~
## $ CV....          <chr> "19.2", "51.6", "46.0", "51.7", "20.4", "50.6", "60.0"~
```

```
strwb_census$crop_type <- str_trim(strwb_census$crop_type, side = "both")

strwb_census$temp2 <- str_trim(strwb_census$temp2, side = "both")

strwb_census$temp3 <- str_trim(strwb_census$temp3, side = "both")
```

```
a <- strwb_census |> distinct(temp2)
```

```
strwb_census <- strwb_census |> mutate(`Fresh Market` = temp2, .after = temp2)
```

```
strwb_census$`Fresh Market` <- strwb_census$`Fresh Market` |> str_replace( "^MEA.*", "")
strwb_census$`Fresh Market` <- strwb_census$`Fresh Market` |> str_replace( "^P.*", "")
```

```
strwb_census$`Fresh Market`[is.na(strwb_census$`Fresh Market`)] <- ""
strwb_census$temp2 <- strwb_census$temp2 |> str_replace("^F.*", "")
strwb_census$`Fresh Market` <- strwb_census$`Fresh Market` |> str_replace("^FRESH MARKET - ", "")
```

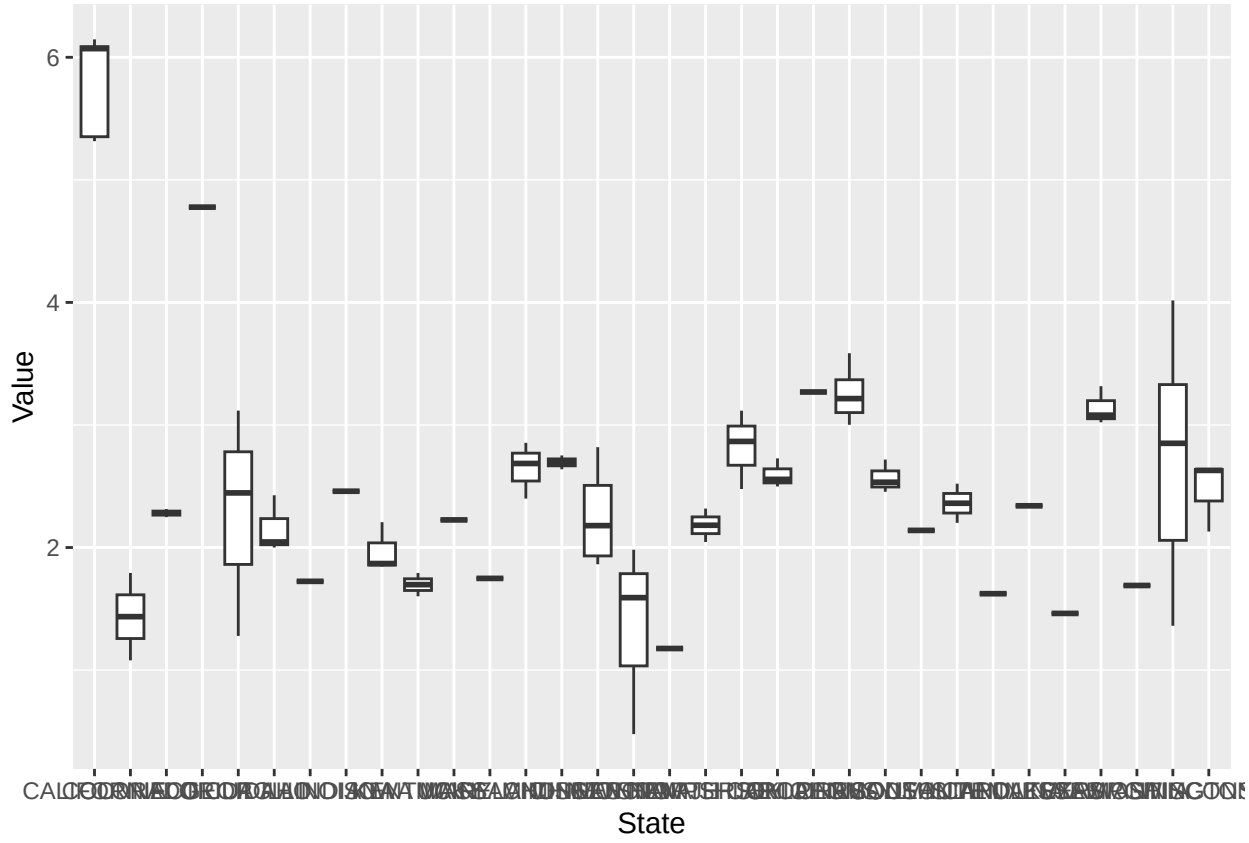
```
strwb_census <- strwb_census |> mutate(`Process Market` = temp2, .after = temp2)
strwb_census$`Process Market` <-  strwb_census$`Process Market` |> str_replace("^MEA.*", "")
strwb_census$`Process Market`[is.na(strwb_census$`Process Market`)] <- ""
strwb_census$temp2 <- strwb_census$temp2 |> str_replace("^P.*", "")
strwb_census$`Process Market` <-  strwb_census$`Process Market` |> str_replace("PROCESSING - ", "")
```

```
strwb_census <- strwb_census |> unite(temp2, temp3, col="Metric", sep="")
strwb_census$Metric <- strwb_census$Metric |> str_replace("MEASURED IN ", "")
strwb_census <- strwb_census |> relocate(Metric, .before = Domain)
strwb_census <- strwb_census |> relocate(`Process Market`, .before = Metric)
strwb_census <- strwb_census |> rename(Totals = prop_acct)
```

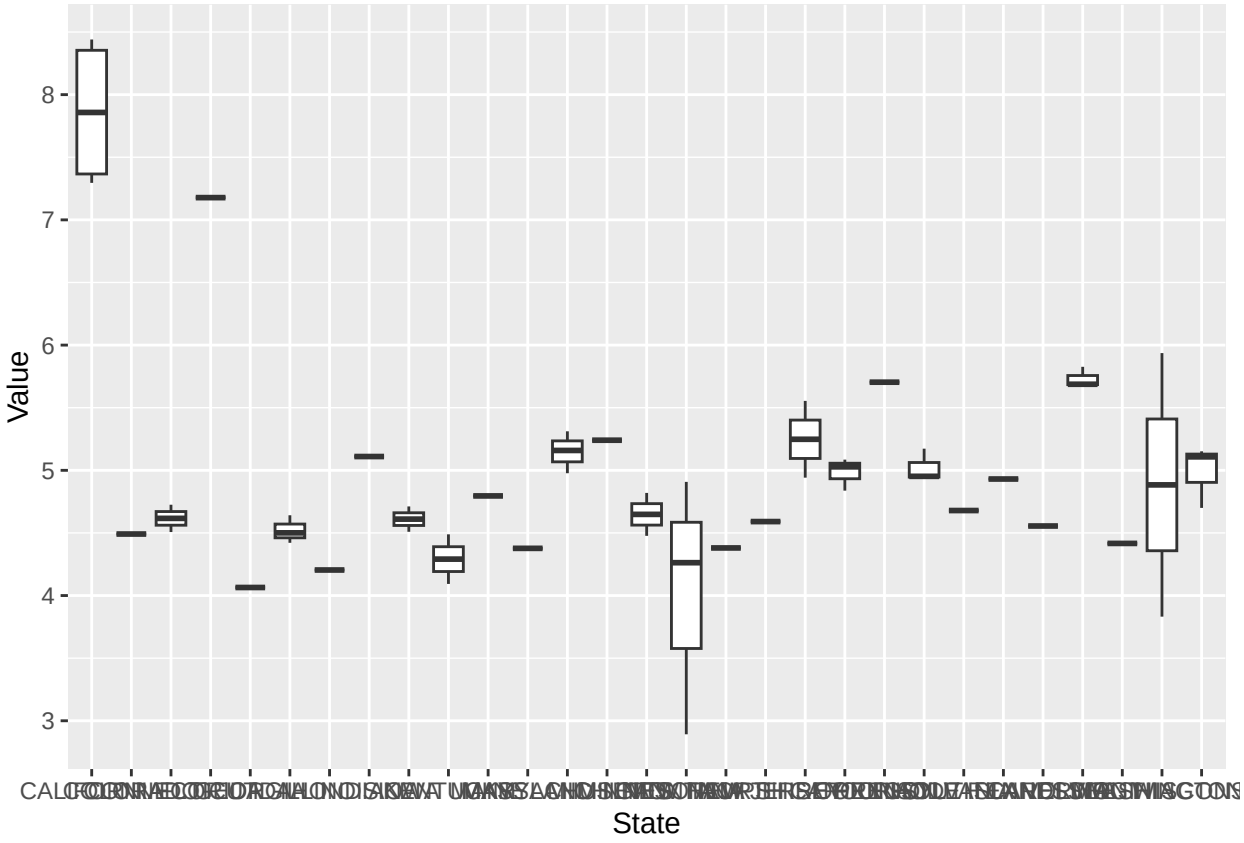
```
vals <- strwb_census$Value
g1 <- sub(",", "", vals)
g2 <- gsub(",", "", vals)
dcomma <- function(c){
  suppressWarnings({
  xnew = as.numeric(gsub(",", "", c))
  fns = unique(c[is.na(xnew)])
  vtran = list("new_vec" = xnew, "footnotes" = fns)
  return(vtran)
  })
}
```

First I want to visualize the 'CWT' and '$' of each state

```
subset1<-strwb_census%>%filter(Metric=='CWT')
ggplot(subset1, aes(x = State, y = log10(Value))) +
  geom_boxplot() +
  xlab("State") +
  ylab("Value")
```



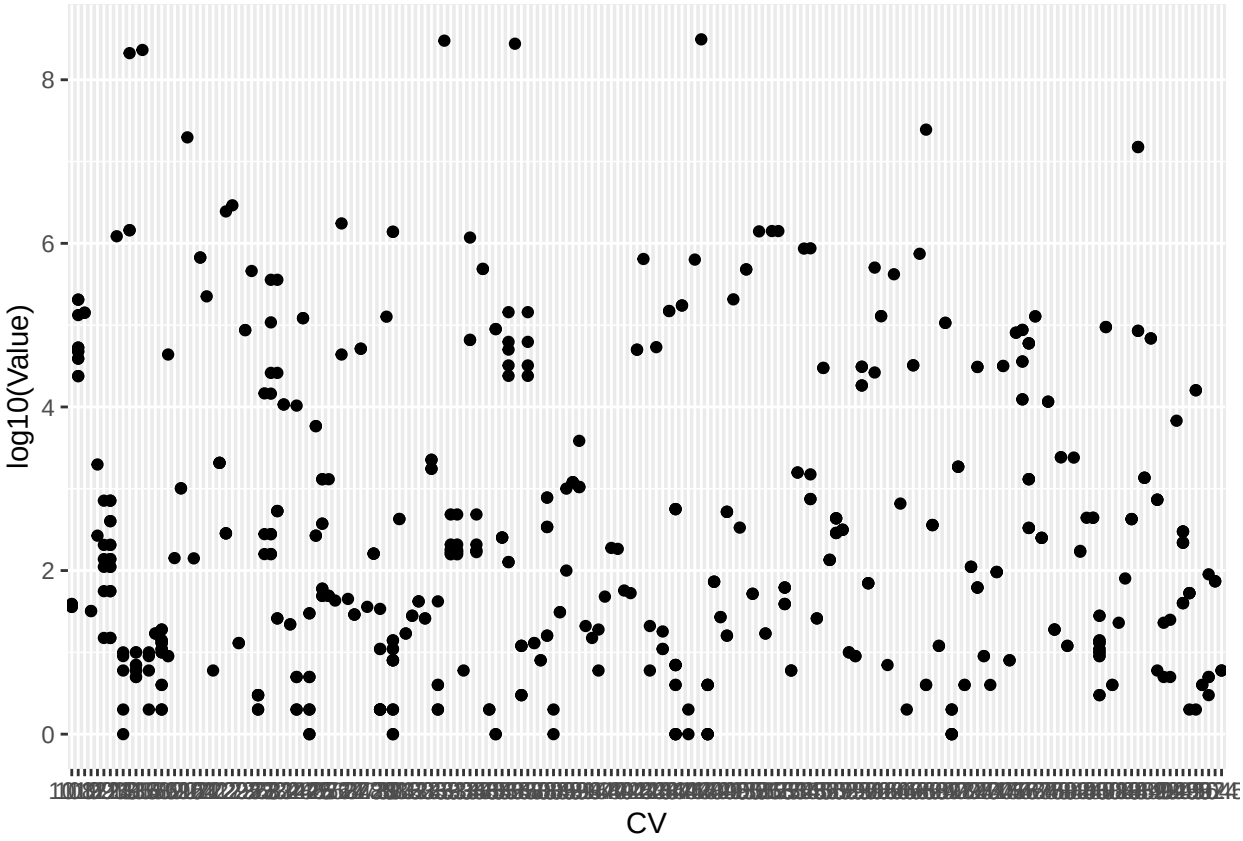
```
subset2<-strwb_census%>%filter(Metric=='$')
ggplot(subset2, aes(x = State, y = log10(Value))) +
  geom_boxplot() +
  xlab("State") +
  ylab("Value")
```



I found for both sales ending in CWT and $ as metric are the highest in California and they are far greater than other states.

I want to make sure the relationship between CV and Value

```
ggplot(strwb_census, aes(x = CV...., y = log10(Value))) +
  geom_point()+
  xlab("CV") +
  ylab("log10(Value)")
```

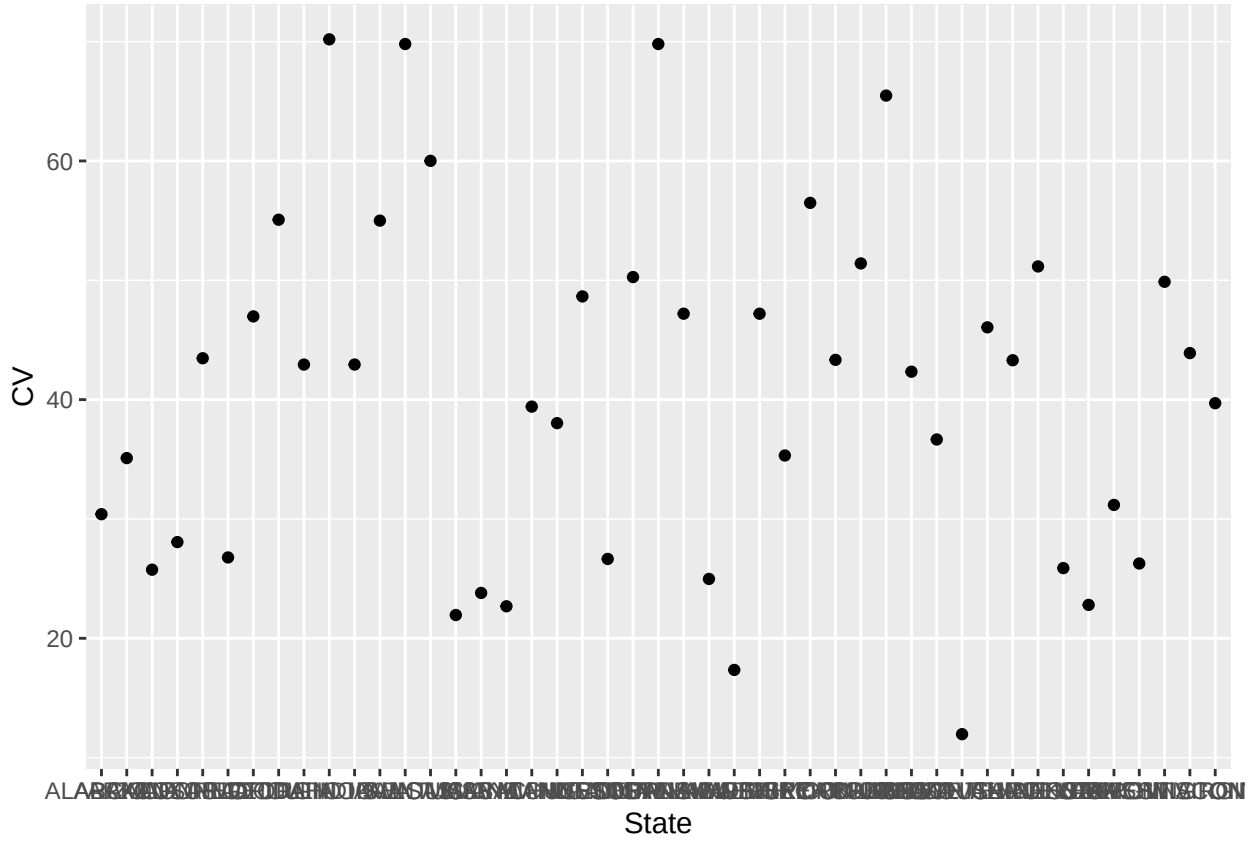


It seems like there is no obvious relationship between CV and Value

Then I want to show the CV and its related states

```
summary_data1 <- strwb_census %>%
  group_by(State) %>%
  summarize(mean = mean(as.numeric(CV....)),
            .groups='drop')

ggplot(summary_data1, aes(x = State, y = mean)) +
  geom_point()+
  xlab("State") +
  ylab("CV")
```



I want to select the top 10 states

```
top_10_states <- summary_data1 %>%
  arrange(desc(mean)) %>%
  head(10)
top_10_states
```

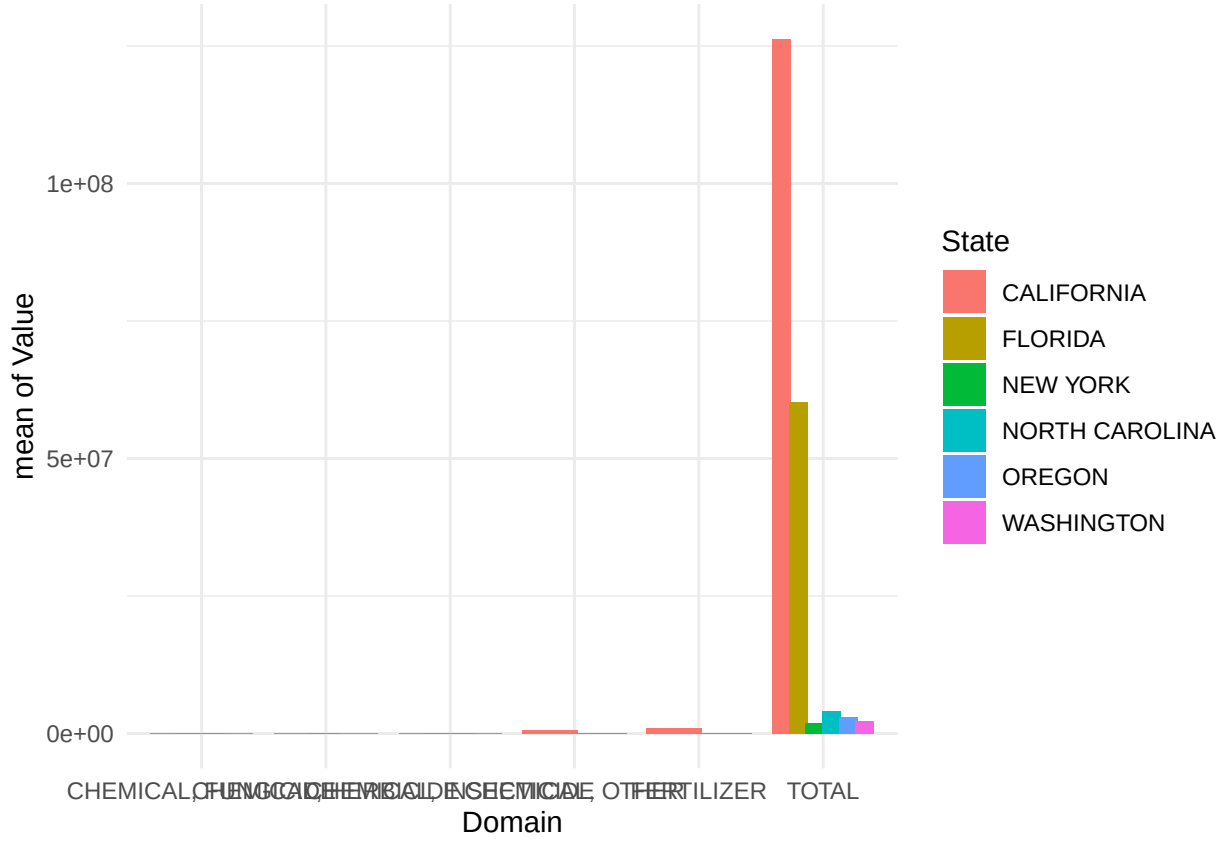
```
## # A tibble: 10 x 2
##    State           mean
##    <chr>          <dbl>
##  1 ILLINOIS        70.2
##  2 KANSAS          69.8
##  3 NEBRASKA        69.8
##  4 OKLAHOMA        65.5
##  5 KENTUCKY        60.0
##  6 NORTH CAROLINA  56.5
##  7 GEORGIA         55.1
##  8 IOWA            55
##  9 OHIO            51.4
## 10 TENNESSEE       51.2
```

Then for strawberry survey, I want to see the mean value based on different chemicals used in Domain.

```
summary_data1 <- strwb_survey %>%
  group_by(Domain,State) %>%
  summarize(mean = mean(Value),
```

```
            .groups='drop')

ggplot(summary_data1, aes(x = Domain, y = mean, fill = as.factor(State))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8)) +
  labs(x = "Domain", y = "mean of Value") +
  scale_fill_discrete(name = "State") +
  theme_minimal()
```



We found that the value for different domains, the most frequent one is Total, I also want to know the top 10 states

```
top_10_states <- summary_data1 %>%
  arrange(desc(mean)) %>%
  head(10)
top_10_states
```

```
## # A tibble: 10 x 3
##    Domain          State                mean
##    <chr>           <chr>               <dbl>
##  1 TOTAL           CALIFORNIA     126315561.
##  2 TOTAL           FLORIDA         60264035.
##  3 TOTAL           NORTH CAROLINA   4160839.
##  4 TOTAL           OREGON           3016359.
##  5 TOTAL           WASHINGTON       2254981.
##  6 TOTAL           NEW YORK         1826790.
```

```
##  7 FERTILIZER      CALIFORNIA        977822.
##  8 CHEMICAL, OTHER CALIFORNIA        554140.
##  9 CHEMICAL, OTHER FLORIDA            67250
## 10 FERTILIZER      FLORIDA            49838.
```

The conclusion can be that: California always have the largest value and CV.