Berries Project

Yuxi Wang

2020/10/11

1 Data Cleaning

1.1 Data Import

After downloading data from the National Agriculture Statistics Service (NASS) of United States Department of Agriculture(USDA), we have the dataset containinig information about three types of berries: Blueberries, Strawberries and Raspberries.

Since there are only 8 out 21 columns that are useful for further analysis, we will drop those first for simplicity of the dataset.

```
dt <- read.csv(file="/Users/mac/Desktop/berries.csv",header=T)
berry_raw <- dt %>%
  select(Year,Period,State,Commodity,Data.Item,Domain,Domain.Category,Value)
head(berry_raw)
```

```
##   Year         Period      State   Commodity
## 1 2019 MARKETING YEAR CALIFORNIA BLUEBERRIES
## 2 2019 MARKETING YEAR CALIFORNIA BLUEBERRIES
## 3 2019 MARKETING YEAR CALIFORNIA BLUEBERRIES
## 4 2019 MARKETING YEAR CALIFORNIA RASPBERRIES
## 5 2019 MARKETING YEAR CALIFORNIA RASPBERRIES
## 6 2019 MARKETING YEAR CALIFORNIA RASPBERRIES
##                                                                Data.Item Domain
## 1                BLUEBERRIES, TAME - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
## 2 BLUEBERRIES, TAME, FRESH MARKET - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
## 3   BLUEBERRIES, TAME, PROCESSING - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
## 4                     RASPBERRIES - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
## 5       RASPBERRIES, FRESH MARKET - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
## 6         RASPBERRIES, PROCESSING - PRICE RECEIVED, MEASURED IN $ / LB  TOTAL
##   Domain.Category Value
## 1   NOT SPECIFIED  2.85
## 2   NOT SPECIFIED  3.56
## 3   NOT SPECIFIED  0.29
## 4   NOT SPECIFIED  2.69
## 5   NOT SPECIFIED   (D)
## 6   NOT SPECIFIED   (D)
```

1.2 Initial Screening of the Data

From the output above, we can notice that there are a lot of categorical varibles. However, `Value` is supposed to be a numeric varible according to the defination on the website.

By looking at the column of `Value` , many (D),(NA),(X) and (Z) appears to be the reason why this column is defined as categorical. So, we will replace those with NA.

```
berry_raw$Value <- as.numeric(berry_raw$Value)
```

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

```
# Replace (D),(NA),(X) and (Z) with NA
berry_raw[berry_raw =="(D)"] <- NA
berry_raw[berry_raw =="(NA)"] <- NA
berry_raw[berry_raw =="(X)"] <- NA
berry_raw[berry_raw =="(Z)"] <- NA
```

Since those irregular “NA”s have been replaced, a summary of the dataset should be made for further exploration of the data.

```
# Summary of berry_raw
summary(berry_raw)
```

```
##       Year         Period             State            Commodity
##  Min.   :2015   Length:13238       Length:13238       Length:13238
##  1st Qu.:2016   Class :character   Class :character   Class :character
##  Median :2017   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2017
##  3rd Qu.:2019
##  Max.   :2019
##
##   Data.Item            Domain          Domain.Category        Value
##  Length:13238       Length:13238       Length:13238       Min.   :  0.000
##  Class :character   Class :character   Class :character   1st Qu.:  0.550
##  Mode  :character   Mode  :character   Mode  :character   Median :  1.831
##                                                           Mean   : 49.564
##                                                           3rd Qu.: 26.000
##                                                           Max.   :960.000
##                                                           NA's   :8854
```

```
view(berry_raw)
```

1.3 Further Data Cleaning on Blueberries

After finishing the initial screening of the dataset, we use the `filter` function to extract data of strawberry to conduct further study. The summary of the strawberry dataset shows that there are 4958 NAs in the column `Value`. Since those observations does not contain much information, we choose to delete them.

1.3.1 Cleaning: `Data Item`

```
strawberry_raw <- berry_raw %>% filter(Commodity=="STRAWBERRIES")
# Summary of the dataset
summary(strawberry_raw)
```

```
##       Year         Period             State            Commodity
##  Min.   :2015   Length:3476        Length:3476        Length:3476
##  1st Qu.:2016   Class :character   Class :character   Class :character
##  Median :2018   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2017
##  3rd Qu.:2019
##  Max.   :2019
##
##   Data.Item            Domain          Domain.Category        Value
##  Length:3476        Length:3476        Length:3476        Min.   :  0.000
##  Class :character   Class :character   Class :character   1st Qu.:  0.307
##  Mode  :character   Mode  :character   Mode  :character   Median :  2.000
##                                                           Mean   : 63.618
##                                                           3rd Qu.: 37.000
##                                                           Max.   :960.000
##                                                           NA's   :2247
```

```
strawberry_raw2 <- strawberry_raw %>% drop_na()
item_pre <- strawberry_raw2$Data.Item
# Replace "-" with "," for the convenience of spliting
item <- gsub(" - ",",",item_pre)
view(item)
```

```
# Type of the strawberry
type_stberry <- str_extract_all(item,"(BEARING){1}")
```

1.3.2 Cleaning: `Domain Category`

Then, we will separate the chemical type and the detail of certain kind of chemical from the column `Domain Category` by using `separate` function in tidyverse package.

```
chemical_obj <- data.frame(strawberry_raw2$Domain.Category)
```

2 EDA of the data

2.1 exploring the hole data set

2.1.1 summarize data by grouping “measurement”

Since we can find the measurement of each data are different, we need to group them and summarize them.

```
# Measurement of the strawberry
strawberry_raw2$unit <- str_extract_all(item,"MEASURED IN.*[^, /AVG]|ACRES.*")
# Also, we have to delate the comma and space
strawberry_raw2$unit <- str_replace(strawberry_raw2$unit,",","")
strawberry_raw2$unit <- trimws(strawberry_raw2$unit)
strawberry_raw2$unit <- as.character(strawberry_raw2$unit)
strawberry_sum <- strawberry_raw2 %>%
  group_by(unit)%>%
  summarize(
    count=n(),
    value=sum(Value)
    )
```

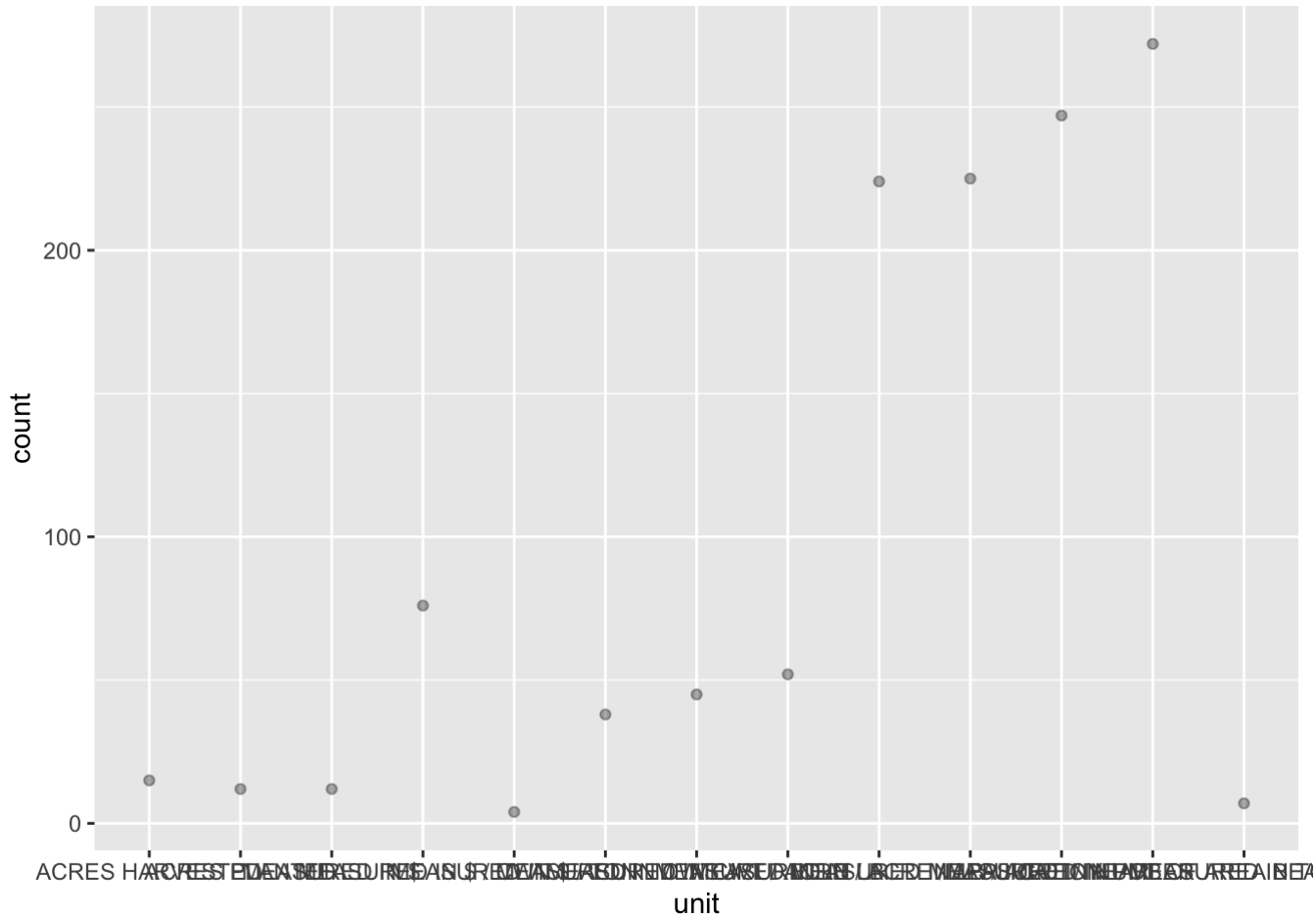
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
view(strawberry_raw2)
print(strawberry_sum)
```

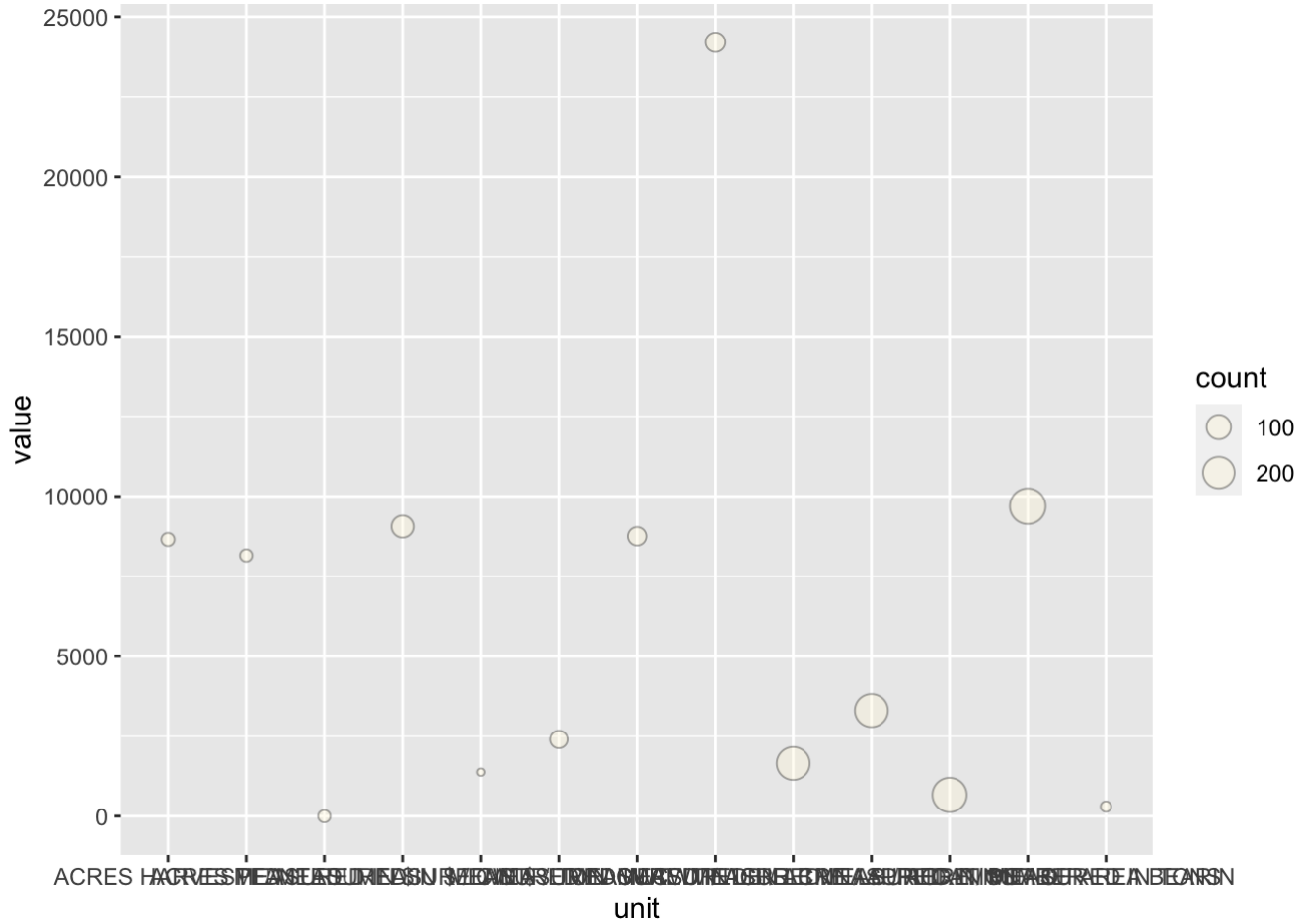
```
## # A tibble: 13 x 3
##    unit                               count  value
##    <chr>                              <int>  <dbl>
##  1 ACRES HARVESTED                       15  8650
##  2 ACRES PLANTED                         12  8150
##  3 MEASURED IN $                         12     0
##  4 MEASURED IN $ / CWT                   76  9054.
##  5 MEASURED IN $ / TON                    4  1375
##  6 MEASURED IN CWT                       38  2400
##  7 MEASURED IN CWT / ACRE                45  8750
##  8 MEASURED IN LB                        52 24200
##  9 MEASURED IN LB / ACRE / APPLICATION  224  1649.
## 10 MEASURED IN LB / ACRE / YEAR         225  3304.
## 11 MEASURED IN NUMBER                   247   664.
## 12 MEASURED IN PCT OF AREA BEARIN       272  9691
## 13 MEASURED IN TONS                       7   299
```

2.1.2 Since we know that their are 13 measurements of the Value, we first plot the total plot of all these item.

```
ggplot(data = strawberry_sum, mapping = aes(x = unit, y = count))+
   geom_point(shape=21,fill="black", alpha = 1/3)
```




```
ggplot(data = strawberry_sum, mapping = aes(x = unit, y = value, size = count)) +
  geom_point(shape=21,fill="cornsilk", alpha = 1/3)
```



The first plot shows that the number of each measurement. While the second plot show that the value of each measurement. this has a little meaning, but I think we have to have a roughly knowledge of the value of each measurement. The huge difference can show why I need to separate the “item” variable.

2.2.1 Creat a new frame in order to exact data to do the further EDA

```
# Creat a new data.frame in order to exact data to do the further EDA
strawberry_unit <- strawberry_raw2 %>%
  group_by(unit)%>%
  summarize(
    state=State,
    year= Year,
    count=n(),
    value=Value
    )
```

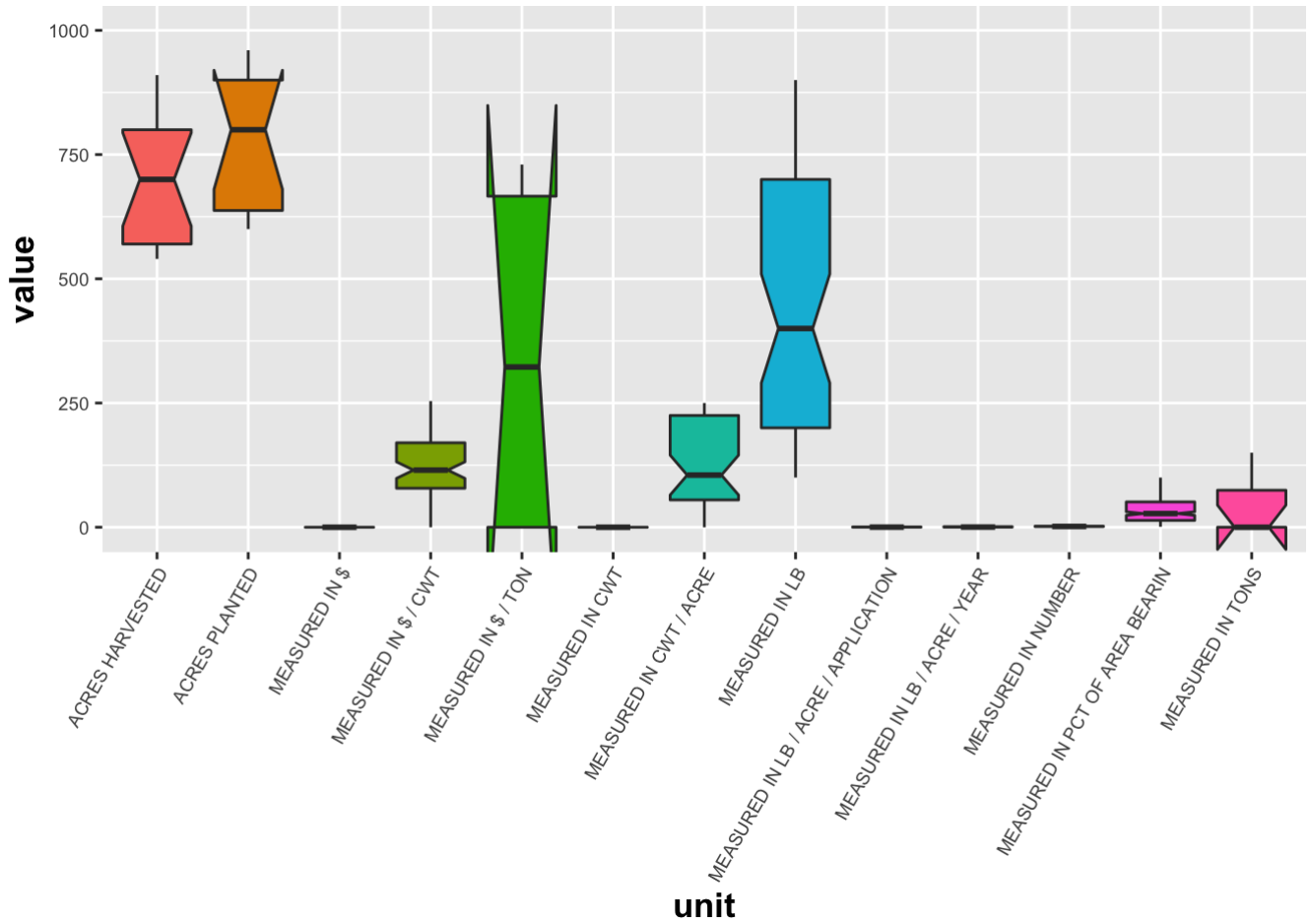
```
## `summarise()` regrouping output by 'unit' (override with `.groups` argument)
```

```
view(strawberry_unit)
```

To make a box plot This plot is just let us have a knowledge that which measurement can be use as a variable in the futher exploring analysis. Since the data that are have much outliers or the range of the data are not great can make a misunderstanding of the model.

```
# excluding outliers
boxplot <- ggplot(strawberry_unit, aes(x = unit, y = value, fill=unit))+
  geom_boxplot(outlier.colour = NA,notch = TRUE) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1),
        axis.text = element_text(size = 7),
        axis.title = element_text(size = 13, face = "bold")) +
  coord_cartesian(ylim = c(0, 1000)) +
  guides(fill=FALSE)
print(boxplot)
```

```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



2.2.2 Creat the data.frame that only contain the $/CWT

```
strawberry_unit_CWT <- filter(strawberry_unit,unit=="MEASURED IN $ / CWT" )
strawberry_unit_CWT$value <- as.numeric(strawberry_unit_CWT$value)
strawberry_unit_CWT$value[strawberry_unit_CWT$value ==0] <- NA # Replace 0 with NA
strawberry_unit_CWT_new <- group_by(strawberry_unit_CWT,year,state)
strawberry_CWT <- summarize(strawberry_unit_CWT_new, value = mean(value, na.rm = TRUE
))
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

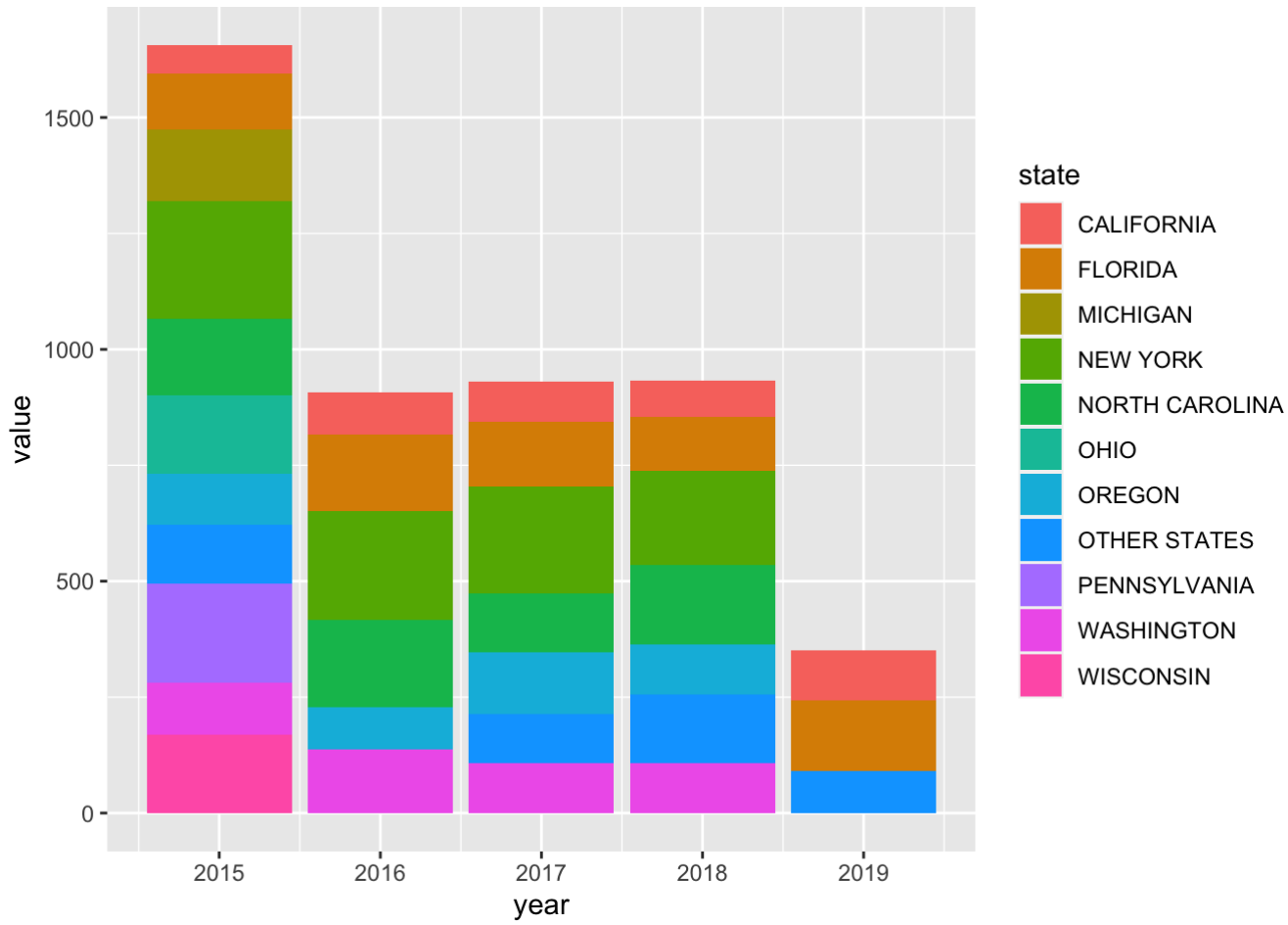
```
summary(strawberry_CWT)
```

```
##       year         state               value
##  Min.   :2015   Length:35          Min.   : 62.47
##  1st Qu.:2015   Class :character   1st Qu.:107.00
##  Median :2016   Mode  :character   Median :130.73
##  Mean   :2017                      Mean   :140.54
##  3rd Qu.:2018                      3rd Qu.:169.00
##  Max.   :2019                      Max.   :254.00
##                                    NA's   :1
```

Making a plot that the x-label is year, the y-label is value and different color means difference states.

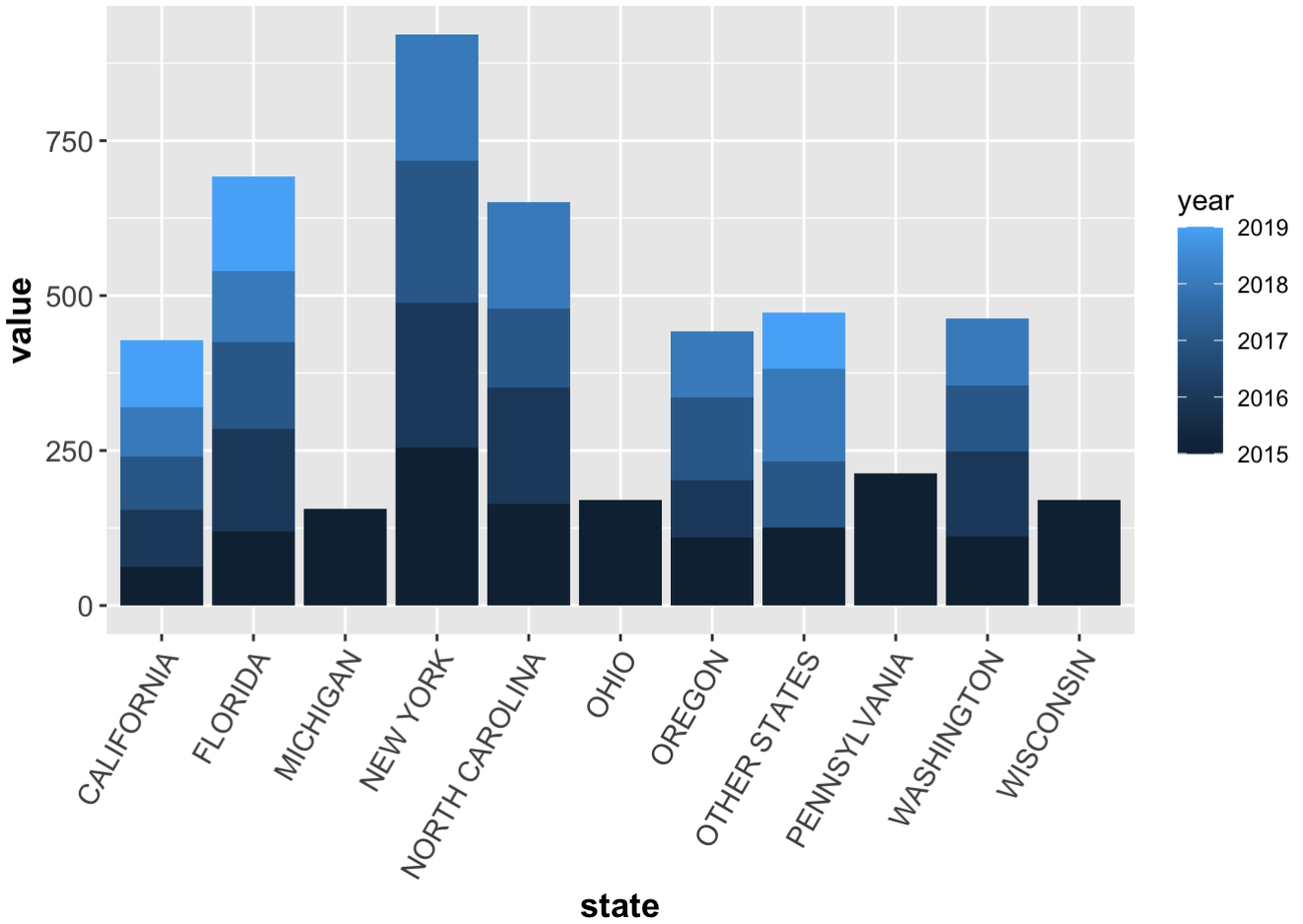
```
# Plot $/ CWT plot
ggplot(strawberry_CWT, aes(x = year, y = value, fill=state)) +
   geom_bar(stat = "identity")
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



```
# We can also transfer the plot in other way
ggplot(strawberry_CWT, aes(x =state , y = value, fill=year)) +
   geom_bar(stat = "identity")+
   theme(axis.text.x = element_text(angle = 60, hjust = 1),
         axis.text = element_text(size = 11),
         axis.title = element_text(size = 13, face = "bold"))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```

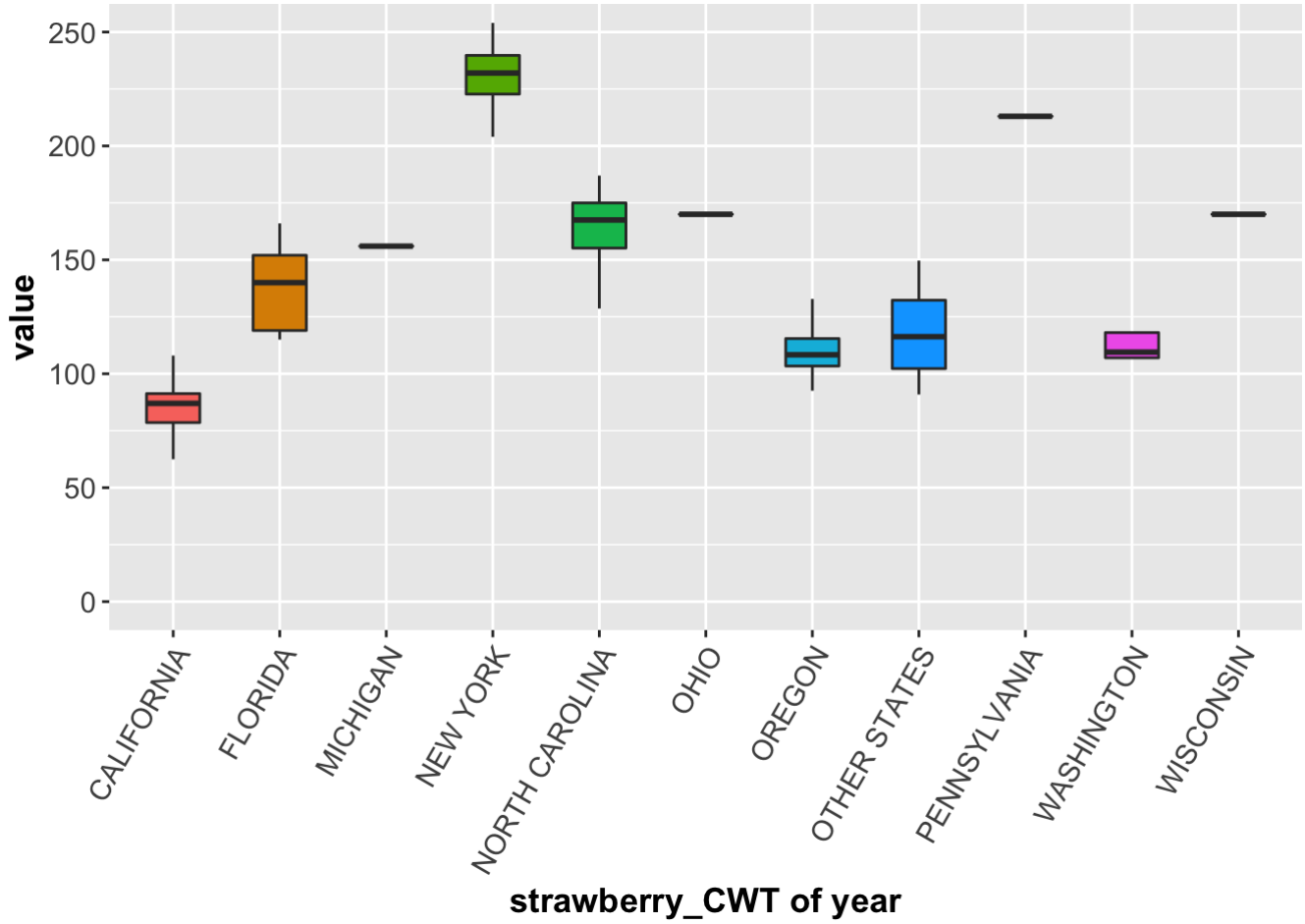


By looking at these two plots, the we can have more insight of the data that use $ /cwt as measurement.

2.2.3 To find the outliers of the $/CWT

```
# excluding outliers
outliers <- ggplot(strawberry_CWT, aes(x =state, y = value,fill=state))
outliers <- outliers +
  geom_boxplot(width=0.5, outlier.colour = NA) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1),
        axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  coord_cartesian(ylim = c(0, 250)) +
  labs(x = "strawberry_CWT of year")+
  guides(fill= FALSE)
print(outliers)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



By looking at this plot, there are servel states that the data of tem are not very great. However, since I have not separate the year of the variable.

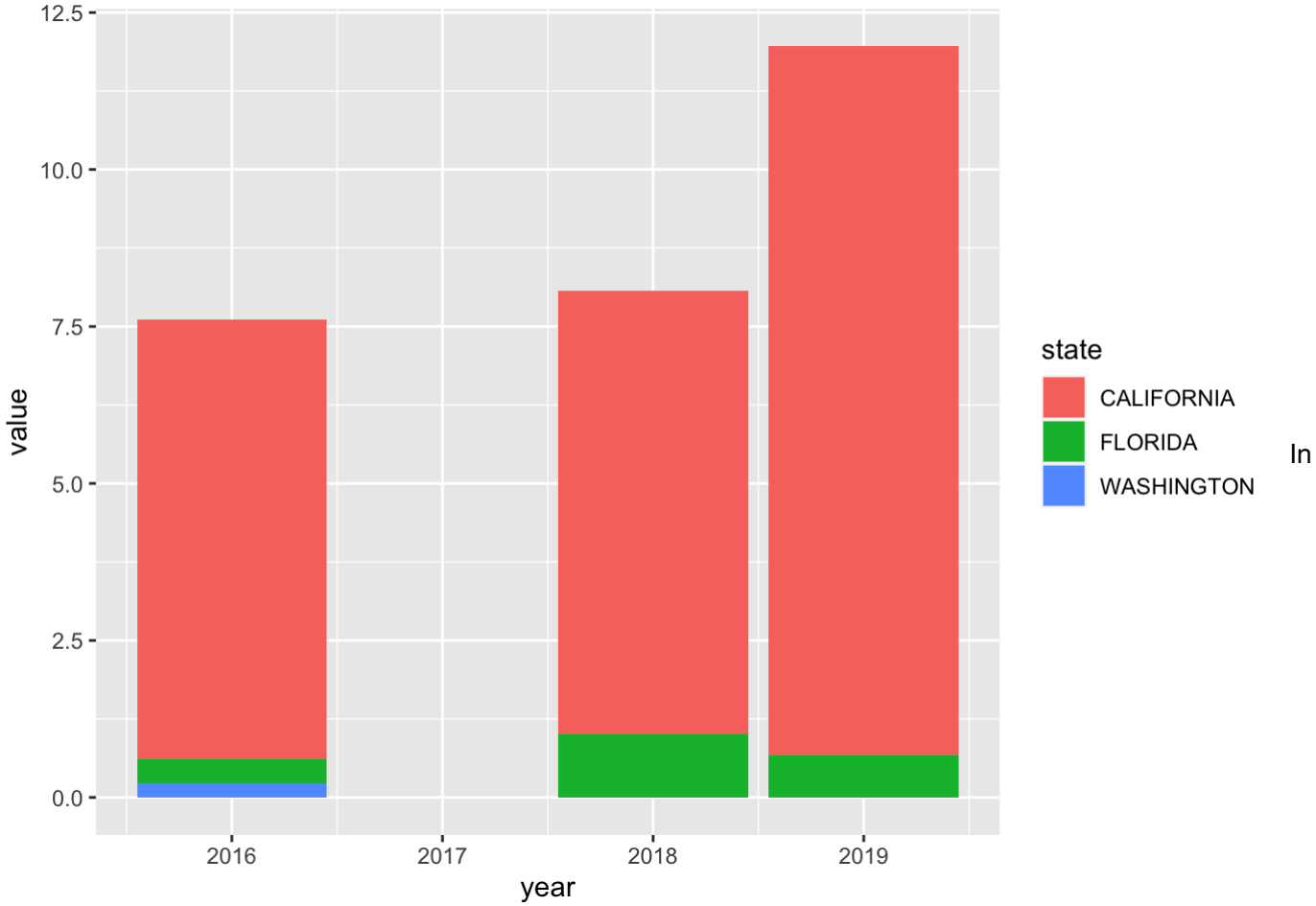
2.3 After analysising the dwt variable, the MEASURED IN LB / ACRE / APPLICATION is also important, so I make a plot.

2.3.1 tidy data

```
# tidy the data
strawberry_unit_APPLICATION <- filter(strawberry_unit,unit=="MEASURED IN LB / ACRE /
 APPLICATION" )
strawberry_unit_APPLICATION$value <- as.numeric(strawberry_unit_APPLICATION$value)
strawberry_unit_APPLICATION$value[strawberry_unit_APPLICATION$value ==0] <- NA # Repl
ace 0 with NA
strawberry_unit_APPLICATION_new <- group_by(strawberry_unit_APPLICATION,year,state)
strawberry_APPLICATION <- summarize(strawberry_unit_APPLICATION_new, value = mean(val
ue, na.rm = TRUE))
```

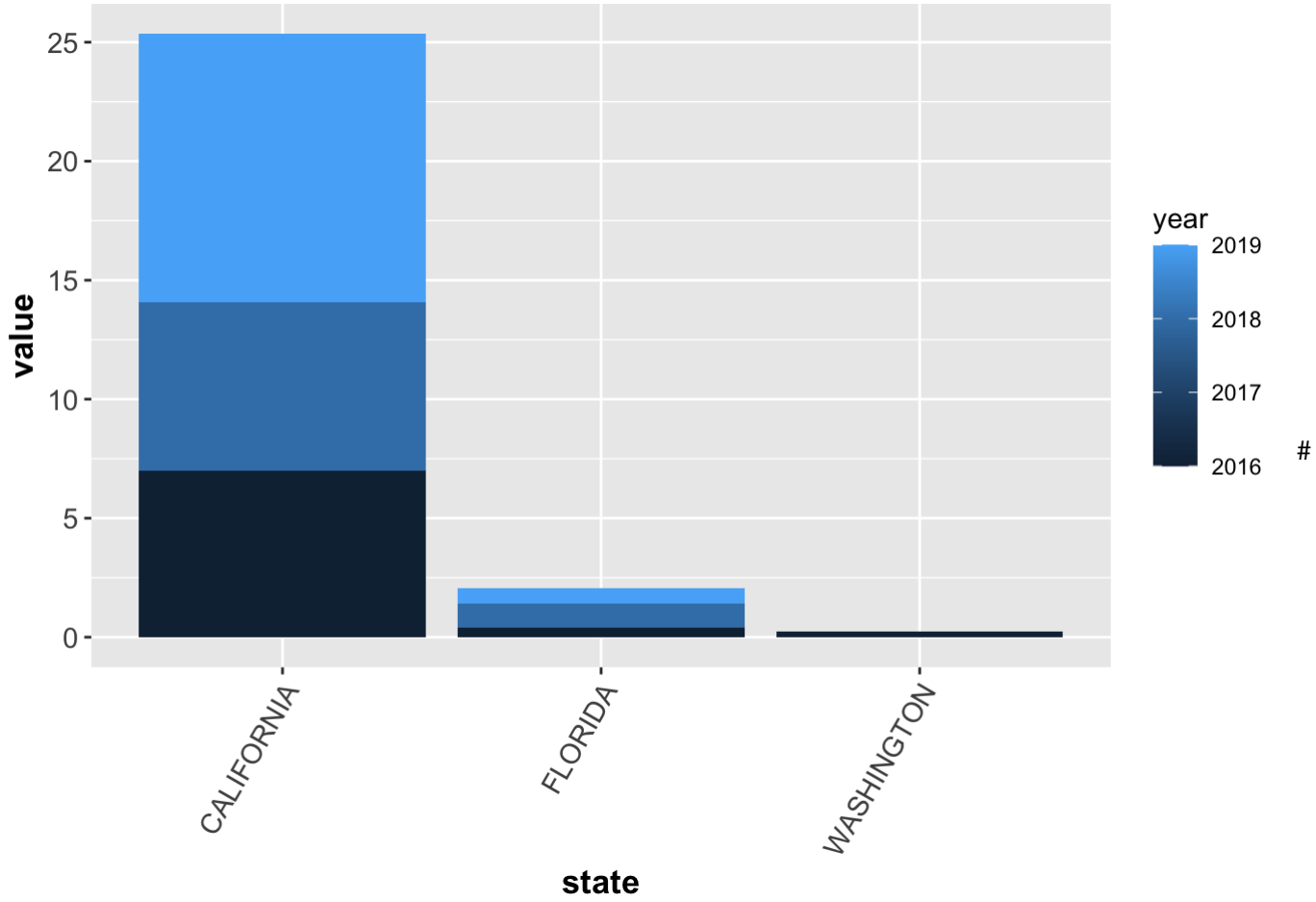
```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
# Plot MEASURED IN LB / ACRE / APPLICATION plot
ggplot(strawberry_APPLICATION, aes(x = year, y = value, fill=state)) +
   geom_bar(stat = "identity")
```



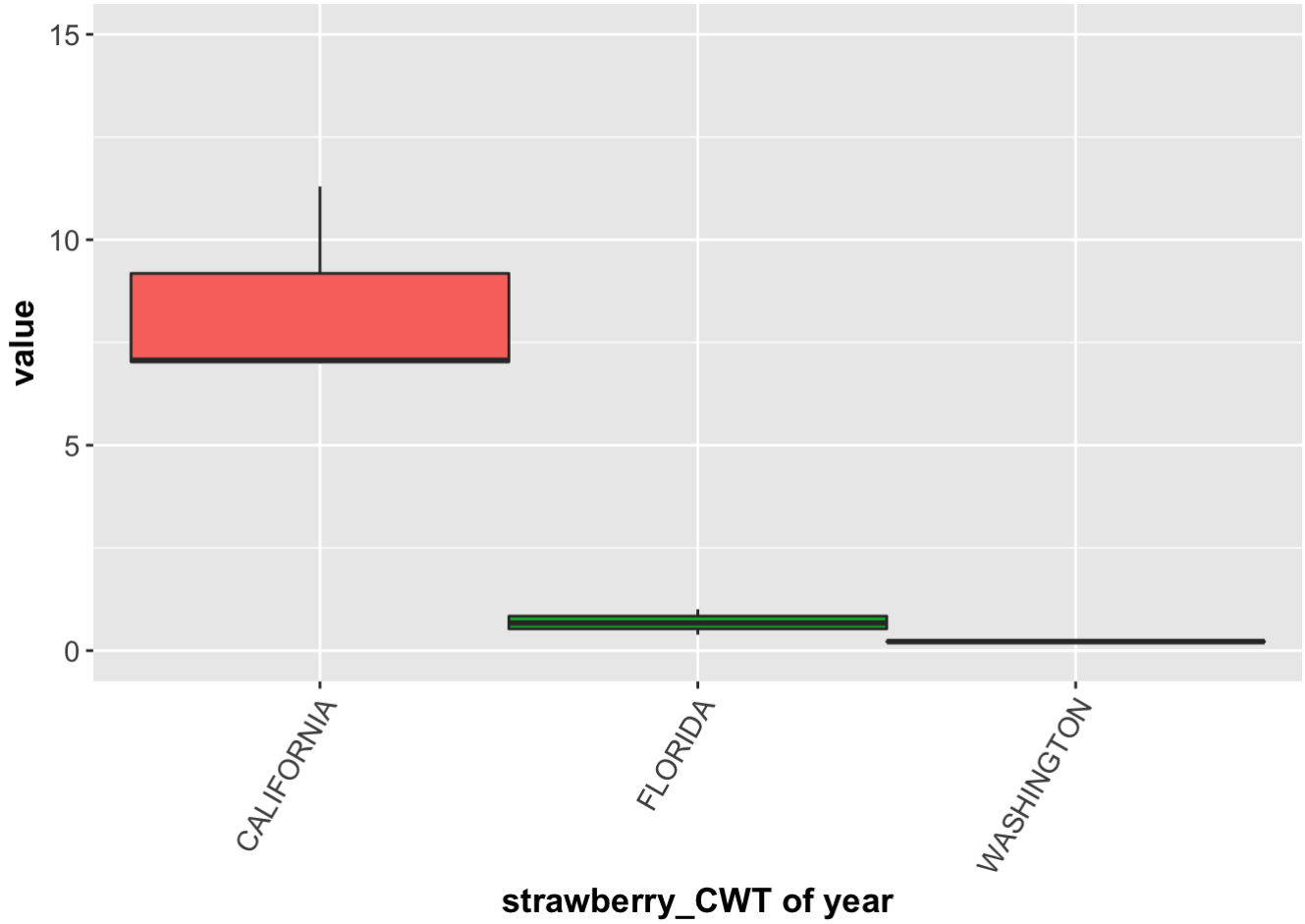
In this plot, we can find that there are no data in 2017, so we would better not to use year as a main variable.

```
# We can also transfer the plot in other way
ggplot(strawberry_APPLICATION, aes(x =state , y = value, fill=year)) +
   geom_bar(stat = "identity")+
   theme(axis.text.x = element_text(angle = 60, hjust = 1),
        axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold"))
```



In this plot we can just have a view of these three states, since other states are not involved in the data set for MEASURED IN LB / ACRE / APPLICATION.

```
# excluding outliers
outliers <- ggplot(strawberry_APPLICATION, aes(x =state, y = value,fill=state))
outliers <- outliers +
  geom_boxplot(width=1, outlier.colour = NA) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1),
        axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  coord_cartesian(ylim = c(0, 15)) +
  labs(x = "strawberry_CWT of year")+
  guides(fill= FALSE)
print(outliers)
```



We can find whether there are outliers in the plot.

3 Shiny part

3.1 Using shiny to make the strewberry data as a chart

```
library(shiny)
library(tidyverse)

dt <- read.csv(file="/Users/mac/Desktop/berries.csv",header=T)
berry_raw <- dt %>%
    select(Year,Period,State,Commodity,Data.Item,Domain,Domain.Category,Value)

berry_raw$Value <- as.numeric(berry_raw$Value)
```

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

```
# Replace (D),(NA),(X) and (Z) with NA
berry_raw[berry_raw =="(D)"] <- NA
berry_raw[berry_raw =="(NA)"] <- NA
berry_raw[berry_raw =="(X)"] <- NA
berry_raw[berry_raw =="(Z)"] <- NA
strawberry_raw <- berry_raw %>% filter(Commodity=="STRAWBERRIES")
strawberry_raw2 <- strawberry_raw %>% drop_na()
item_pre <- strawberry_raw2$Data.Item
# Replace "-" with "," for the convenience of spliting
item <- gsub(" - ",",",item_pre)
# Type of the strawberry
type_stberry <- str_extract_all(item,"(BEARING){1}")
strawberry_raw2$unit <- str_extract_all(item,"MEASURED IN.*[^, /AVG]|ACRES.*")
# Also, we have to delate the comma and space
strawberry_raw2$unit <- str_replace(strawberry_raw2$unit,",","")
strawberry_raw2$unit <- trimws(strawberry_raw2$unit)
strawberry_raw2$unit <- as.character(strawberry_raw2$unit)
strawberry_sum <- strawberry_raw2 %>%
    group_by(unit)%>%
    summarize(
        count=n(),
        value=sum(Value)
    )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# Shiny app
strawberry_unit <- as.data.frame(strawberry_unit)
ui <- fluidPage(
    title = "Strawberry Data Tables",
    mainPanel(
        tabsetPanel(
            id = 'dataset',
            tabPanel("strawberry data",

                     # Create a new Row in the UI for selectInputs
                     fluidRow(
                         column(4,
                                selectInput("measurement",
                                            "kinds of measurement:",
                                            c("All",
                                              unique(as.character(strawberry_unit$uni
t))))
                         ),
                         column(4,
                                selectInput("year",
                                            "Year:",
                                            c("All",
                                              unique(strawberry_unit$year)))
                         ),
                         column(4,
                                selectInput("state",
                                            "State:",
                                            c("All",
                                              unique(strawberry_unit$state)))
                         ),
                     ),
                     # Create a new row for the table.
                     DT::dataTableOutput("table1")))

    )
)


server <- function(input, output) {
    strawberry_unit$unit <- as.character(strawberry_unit$unit)
    strawberry_unit$year <- as.numeric(strawberry_unit$year)
    strawberry_unit$state <- as.character(strawberry_unit$state)
    # Filter data based on selections
    output$table1 <- DT::renderDataTable(
        DT::datatable({
            data <- strawberry_unit

            if (input$year != "All") {
                data <- data[data$year == input$year,]
            }



            data
        }))
```

```
}

# Run the application
shinyApp(ui = ui, server = server)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it i
s installed, please make sure the phantomjs executable can be found via the PATH vari
able.
```

Shiny applications not supported in static R Markdown documents

3.2 Using shiny to plot the EDA process.

```
library(shiny)
library(tidyverse)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
Sys.setlocale("LC_TIME", "English")
```

```
## Warning in Sys.setlocale("LC_TIME", "English"): OS reports request to set locale
## to "English" cannot be honored
```

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

```
dt <- read.csv(file="/Users/mac/Desktop/berries.csv",header=T)
berry_raw <- dt %>%
    select(Year,Period,State,Commodity,Data.Item,Domain,Domain.Category,Value)

berry_raw$Value <- as.numeric(berry_raw$Value)
```

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

```
# Replace (D),(NA),(X) and (Z) with NA
berry_raw[berry_raw =="(D)"] <- NA
berry_raw[berry_raw =="(NA)"] <- NA
berry_raw[berry_raw =="(X)"] <- NA
berry_raw[berry_raw =="(Z)"] <- NA
strawberry_raw <- berry_raw %>% filter(Commodity=="STRAWBERRIES")
strawberry_raw2 <- strawberry_raw %>% drop_na()
item_pre <- strawberry_raw2$Data.Item
# Replace "-" with "," for the convenience of spliting
item <- gsub(" - ",",",item_pre)
# Type of the strawberry
type_stberry <- str_extract_all(item,"(BEARING){1}")
strawberry_raw2$unit <- str_extract_all(item,"MEASURED IN.*[^, /AVG]|ACRES.*")
# Also, we have to delate the comma and space
strawberry_raw2$unit <- str_replace(strawberry_raw2$unit,",","")
strawberry_raw2$unit <- trimws(strawberry_raw2$unit)
strawberry_raw2$unit <- as.character(strawberry_raw2$unit)
strawberry_sum <- strawberry_raw2 %>%
    group_by(unit)%>%
    summarize(
        count=n(),
        value=sum(Value)
    )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
strawberry_unit <- strawberry_raw2 %>%
    group_by(unit)%>%
    summarize(
        state=State,
        year= Year,
        count=n(),
        value=Value
    )
```

```
## `summarise()` regrouping output by 'unit' (override with `.groups` argument)
```

```
strawberry_unit_CWT <- filter(strawberry_unit,unit=="MEASURED IN $ / CWT" )
strawberry_unit_CWT$value <- as.numeric(strawberry_unit_CWT$value)
strawberry_unit_CWT$value[strawberry_unit_CWT$value ==0] <- NA # Replace 0 with NA
strawberry_unit_CWT_new <- group_by(strawberry_unit_CWT,year,state)
strawberry_CWT <- summarize(strawberry_unit_CWT_new, value = mean(value, na.rm = TRUE
))
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
strawberry_unit_APPLICATION <- filter(strawberry_unit,unit=="MEASURED IN LB / ACRE /
 APPLICATION" )
strawberry_unit_APPLICATION$value <- as.numeric(strawberry_unit_APPLICATION$value)
strawberry_unit_APPLICATION$value[strawberry_unit_APPLICATION$value ==0] <- NA # Repl
ace 0 with NA
strawberry_unit_APPLICATION_new <- group_by(strawberry_unit_APPLICATION,year,state)
strawberry_APPLICATION <- summarize(strawberry_unit_APPLICATION_new, value = mean(val
ue, na.rm = TRUE))
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
# Define UI ----
ui <- fluidPage(
    titlePanel("Analysis of Berries Dataset"),
    verticalLayout(
        mainPanel(
            h3("1. Overview of the classification of the measurement."),
            h4("1.1 two point plot of the measurement"),
            h5("Having a roughly knowledge of the value of each measurement after gro
uping the measurement"),
            plotOutput("p1"),
            plotOutput("p2"),
            h5("The first plot shows that the number of each measurement. While the s
econd plot show that the value of each measurement. this has a little meaning, but I
 think we have to have a roughly knowledge of the value of each measurement. The huge
difference can show why I need to separate the 'item' variable."),

            h4("1.2 Box plot of each knid of measurement"),
            h5("A box plot."),
            plotOutput("p3"),
            h5("This plot is just let us have a knowledge that which measurement can
 be use as a variable in the futher exploring analysis. Since the data that are have
 much outliers or the range of the data are not great can make a misunderstanding of
 the model."),

            h3("2. EDA for the measurement is  $/CWT"),
            h4("2.1 Bar plot of Year"),
            plotOutput("p4"),
            h5("Making a plot that the x-label is year, the y-label is value and diff
erent color means difference states."),

            h4("2.2  Bar plot of state"),
            plotOutput("p5"),
            h5("Making a plot that the x-label is each state, the y-label is value an
d different color means difference years."),

            h4("2.3 box plot of state"),
            plotOutput("p6"),
            h5("By looking at this plot, there are servel states that the data of tem
are not very great. However, since I have not separate the year of the variable.")
        )
    )
)

# Define server logic required to draw a plot
server <- function(input, output) {
    output$p1 <- renderPlot({
        ggplot(data = strawberry_sum, mapping = aes(x = unit, y = count))+
            geom_point(shape=21,fill="black", alpha = 1/3)
    })

    output$p2 <- renderPlot({
        ggplot(data = strawberry_sum, mapping = aes(x = unit, y = value, size = coun
t)) +
            geom_point(shape=21,fill="cornsilk", alpha = 1/3)
    })
    output$p3 <- renderPlot({
        ggplot(strawberry_unit, aes(x = unit, y = value, fill=unit))+
```

```
            geom_boxplot(outlier.colour = NA,notch = TRUE) +
            theme(axis.text.x = element_text(angle = 60, hjust = 1),
                  axis.text = element_text(size = 7),
                  axis.title = element_text(size = 13, face = "bold")) +
            coord_cartesian(ylim = c(0, 1000)) +
            guides(fill=FALSE)
    })

    output$p4 <- renderPlot({
        ggplot(strawberry_CWT, aes(x = year, y = value, fill=state)) +
            geom_bar(stat = "identity")
    })
    output$p5 <- renderPlot({
        ggplot(strawberry_CWT, aes(x =state , y = value, fill=year)) +
            geom_bar(stat = "identity")+
            theme(axis.text.x = element_text(angle = 60, hjust = 1),
                  axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold"))
    })
    output$p6 <- renderPlot({
        ggplot(strawberry_CWT, aes(x =state, y = value,fill=state))+
            geom_boxplot(width=0.5, outlier.colour = NA) +
            theme(axis.text.x = element_text(angle = 60, hjust = 1),
                  axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            coord_cartesian(ylim = c(0, 250)) +
            labs(x = "strawberry_CWT of year")+
            guides(fill= FALSE)
    })

}

# Run the application
shinyApp(ui = ui, server = server)
```

Shiny applications not supported in static R Markdown documents