Report for MA615 Assignment-EDA for Strawberries

Chenxun Li

10/18/2020

1.Background

This is an assignment to help us know how data cleaning, data organization and EDA are generally happens. I use the strawberries for this assignment.

2.Data Cleaning

(1)Read Data

At first, we read the data.

These data were collected from the USDA database selector: https://quickstats.nass.usda.gov

The data were stored online and then downloaded as a CSV file.

```
original_berries <- read_csv("~/Desktop/615/assignment/berry/berries(3).csv", col_names = TRUE)
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_double(),
##   `Week Ending` = col_logical(),
##   `Ag District` = col_logical(),
##   `Ag District Code` = col_logical(),
##   County = col_logical(),
##   `County ANSI` = col_logical(),
##   `Zip Code` = col_logical(),
##   Region = col_logical(),
##   Watershed = col_logical(),
##   `CV (%)` = col_logical()
## )
```

```
## See spec(...) for full column specifications.
```

(2)Data Preparing

We can find that the data selected from the NASS database often has columns without any data or with a single repeated Values, and the berries data had only 8 out of 21 columns containing meaningful data.

So, we need to remove some columns without any data.

```
## look at number of unique values in each column
original_berries %>% summarize_all(n_distinct) -> aa

## make a list of the columns with only one unique value
bb <- which(aa[1,]==1)
```

```r
## list the 1-unique value column names
cn <- colnames(original_berries)[bb]

## remove the 1-unique columns from the data set
original_berries  <-  original_berries[,-all_of(bb)]
aa <-  aa[,-all_of(bb)]

## State name and the State ANSI code are (sort of) redundant
## Just keep the name
original_berries <-  original_berries[,-4]
aa <- aa[,-4]
```

(3)Choose commodity

In this assignment, I choose Strawberries as the commodity.

And, I find that some values are undisclosed and none, so I discard them.

```r
#choose STRAWBERRIES
original_STRAWBERRIES <- filter(original_berries, Commodity=="STRAWBERRIES")

#discard useless values
original_STRAWBERRIES <- filter(original_STRAWBERRIES, Value != "(D)")
original_STRAWBERRIES <- filter(original_STRAWBERRIES, Value != "(NA)")
```

(4)Data Processing- 'Data Item'

I find there are much information in the column 'Data Item' and I only need the unit associated with the statistical category, so I use regular expression to gain the 'unit_desc'.

```r
#read the 'Data Item'
dt_item <- original_STRAWBERRIES$`Data Item`

#replace the '-' with ',' to prepare for spliting
dt_item_with_comma <- gsub(" - ",",",dt_item)

#extract 'MEASURED IN'
original_STRAWBERRIES$unit_desc <- str_extract_all(dt_item_with_comma,"MEASURED.*[^./AVG]|ACRES.*")
```

Now, we can see the categories of measurement.

```r
unique(original_STRAWBERRIES$unit_desc)
```

```
## [[1]]
## [1] "MEASURED IN $ / CWT"
##
## [[2]]
## [1] "ACRES HARVESTED"
##
## [[3]]
## [1] "ACRES PLANTED"
##
## [[4]]
## [1] "MEASURED IN $"
##
## [[5]]
```

```
## [1] "MEASURED IN CWT"
##
## [[6]]
## [1] "MEASURED IN CWT / ACRE"
##
## [[7]]
## [1] "MEASURED IN LB"
##
## [[8]]
## [1] "MEASURED IN LB / ACRE / APPLICATION, "
##
## [[9]]
## [1] "MEASURED IN LB / ACRE / YEAR, "
##
## [[10]]
## [1] "MEASURED IN NUMBER, "
##
## [[11]]
## [1] "MEASURED IN PCT OF AREA BEARING, "
##
## [[12]]
## [1] "MEASURED IN $ / TON"
##
## [[13]]
## [1] "MEASURED IN TONS"
```

(5)Data Processing- 'Domain'

I find that 'Domain' includes both characteristic of operations that produce a particular commodity and some details, so I separate it into two parts.

```
original_STRAWBERRIES <- separate(data=original_STRAWBERRIES, col =6,
                          into=c("Domain", "Domain.Detail"), sep=",")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with 'NA' in 579 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 83, 84, 85, 86, 156, 157, 158, 159, 229, 230, 231, ...].
```

After that, I find some pieces are filled with 'NA', that is the domain=TOTAL, which means there are no further breakouts and the domain=FERTILIZER. So we supplement them.

```
for(i in 1:length(original_STRAWBERRIES$Domain)){
  if(is.na(original_STRAWBERRIES$Domain.Detail[i]) == T){
    original_STRAWBERRIES$Domain.Detail[i] = original_STRAWBERRIES$Domain[i]
  }
}
```

(6)Delete duplicates

Now, we have the data set with 10 columns, but there also some variables are repetitive and useless.

i.'category'

I have choosn strawberries, so this column is useless.

ii.'Data Item'

I have gained the information that I am interested in from it, so I discard this column.

iii.'Domain Category'

I have separated the Domain into two parts, and one of them can reveal some information in Domain Category. Besides, I am not care about some other information in 'Domain Category' such as the price of chemicals.

So, I delete these variables.

```
original_STRAWBERRIES <- original_STRAWBERRIES[,-8]
original_STRAWBERRIES <- original_STRAWBERRIES[,-5]
original_STRAWBERRIES <- original_STRAWBERRIES[,-4]
```

(7)Change types of variables

I would like to change 'Value' from char to numeric for computing in EDA. However, there are some comma symbols in it(these data will turn to NA when use as.numeric), so I try to delete the comma symbols.

```
#delete the comma symbols in Values
for(i in 1:length(original_STRAWBERRIES$Value)){
  original_STRAWBERRIES$Value[i] <- gsub(pattern = ",", replacement = "",
                                 original_STRAWBERRIES$Value[i])
}

#change the type
original_STRAWBERRIES$Year <- as.integer(original_STRAWBERRIES$Year)
original_STRAWBERRIES$Value <- as.numeric(original_STRAWBERRIES$Value)
```

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

The NAs are value=0, and I will process them in EDA.

(8)Finish cleaning

```
#get the new data set
STRAWBERRIES <- original_STRAWBERRIES
```

3. Exploratory Data Analysis

(1)Count types of units

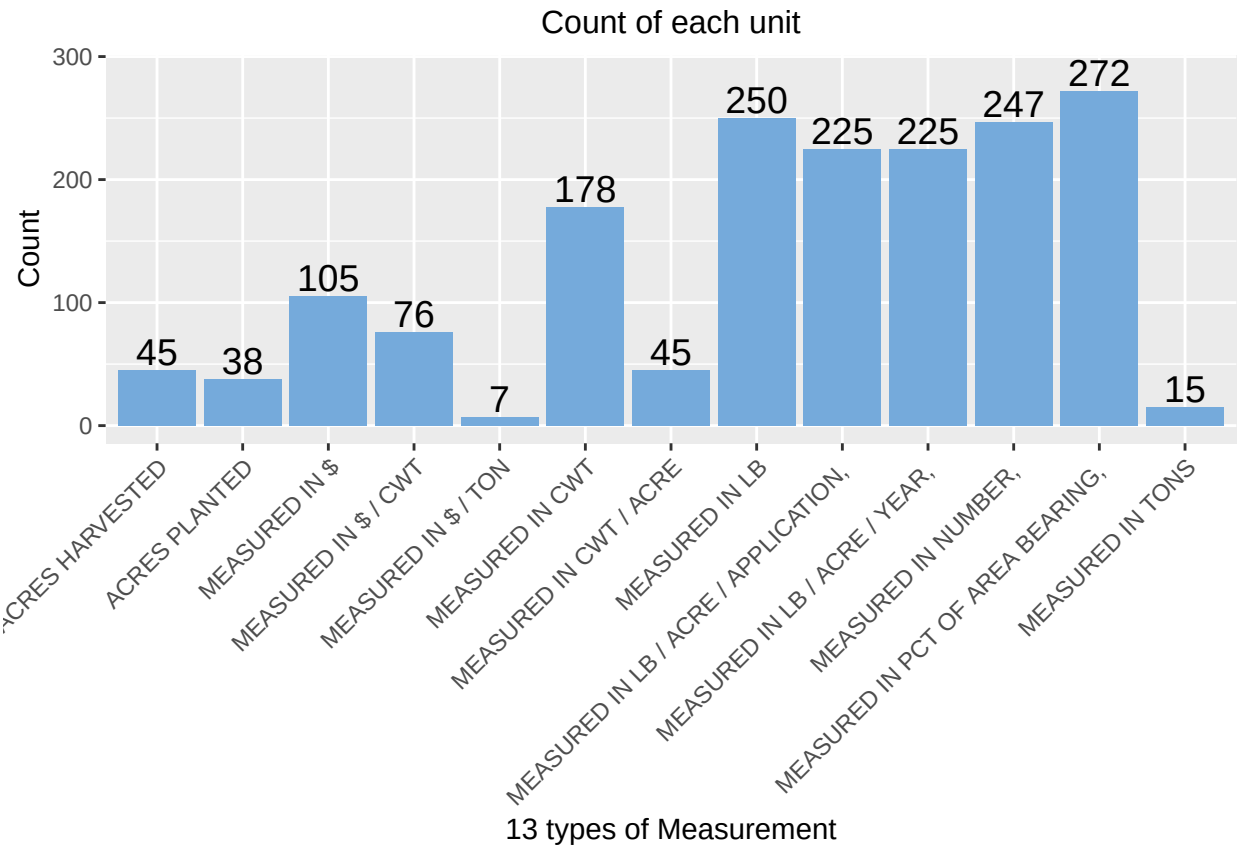
At first, I would like to take a look at how many units in each type. So, I group by unit.

```
#group by unit and summarize
strawberry_for_count_unit <- STRAWBERRIES %>% group_by(unit_desc) %>%
  summarize(
    numbers=n(),
    sum_value=sum(Value)
  )
```

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

```
#Use bar plot to see the difference between each unit
ggplot(data=strawberry_for_count_unit, mapping=aes(x = as.character(unit_desc), y = numbers))+
  geom_bar(stat = "identity", fill = "#75AADB", border = "white")+
  ggtitle("Count of each unit")+xlab("13 types of Measurement")+ylab("Count")+
  geom_text(aes(label=numbers,y=numbers+15),size=5,color="black") +
    theme(plot.title = element_text(hjust = 0.5, size = 12))+
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: border
```



Now, we can see from the plot that these measurements differs large, for example, the 'MEASURED IN LB' is up to 250 but the 'MEASURED IN $ / TON' only has 7.

(2)Trend from 2015 to 2019 betweeen different states

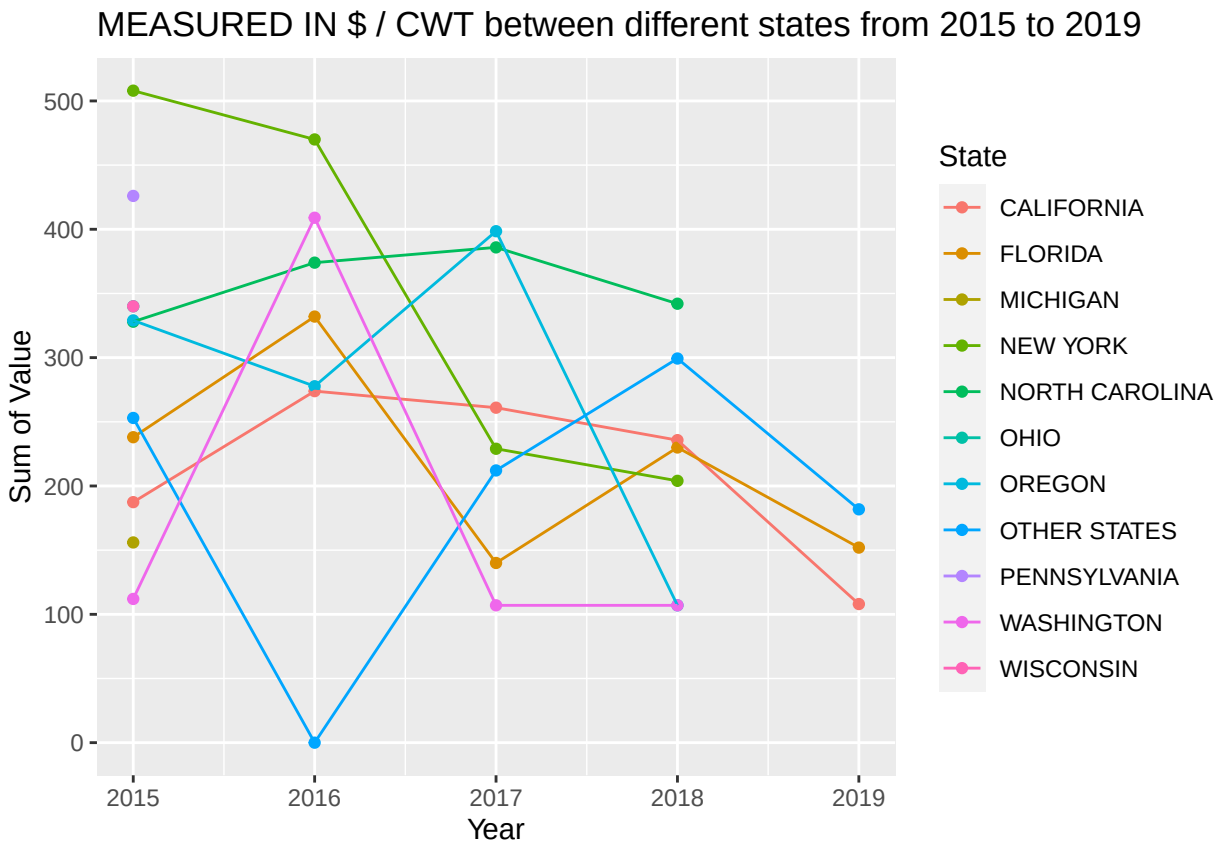
Now, I would like to see the trend between different states from 2015 to 2019 in different measurements.

```
#prepare the data set
strawberry_for_state_change <- STRAWBERRIES %>% group_by(unit_desc,State,Year) %>%
  summarize(
    sum_value=sum(Value)
  )
```

```
## `summarise()` regrouping output by 'unit_desc', 'State' (override with `.groups` argument)
```

```
#choose 'MEASURED IN $ / CWT'
strawberry_measure_in_dollar_CWT <- filter(strawberry_for_state_change,
                                unit_desc=='MEASURED IN $ / CWT')

#plot
ggplot(strawberry_measure_in_dollar_CWT, aes(x=Year, y=sum_value,color=State))+
  geom_point()+geom_line()+ylab('Sum of Value')+
  ggtitle("MEASURED IN $ / CWT between different states from 2015 to 2019")
```



As we can see, all states except 'other states' have a decrease in 2015-2016, but they all fall even sharp drop(New York and Wisconsin) in 2016-2017.

(3)Mean value grouped by unit in years&states

At first, we also need to select the information are interested in. I would like to research value in different years in different states. However, the unit is not uniform, so we group by the unit.

```
strawberry_unit_desc <- STRAWBERRIES %>% group_by(unit_desc) %>%
  summarize(
    state=State,
    year=Year,
    numbers=n(),
    value=Value
  )
```

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

Now, we can EDA for the new data set.

```
#create a data frame of different units
dt <- unique(strawberry_unit_desc$unit_desc)
dt <- do.call(rbind,dt)

## create the loop to perform 13 plots
plot_list <- list()
for(i in 1:length(dt)){
  #create flag variables
  strawberry_unit_desc_for_loop <- data.frame(strawberry_unit_desc)
  #extract specific unit from data set
```

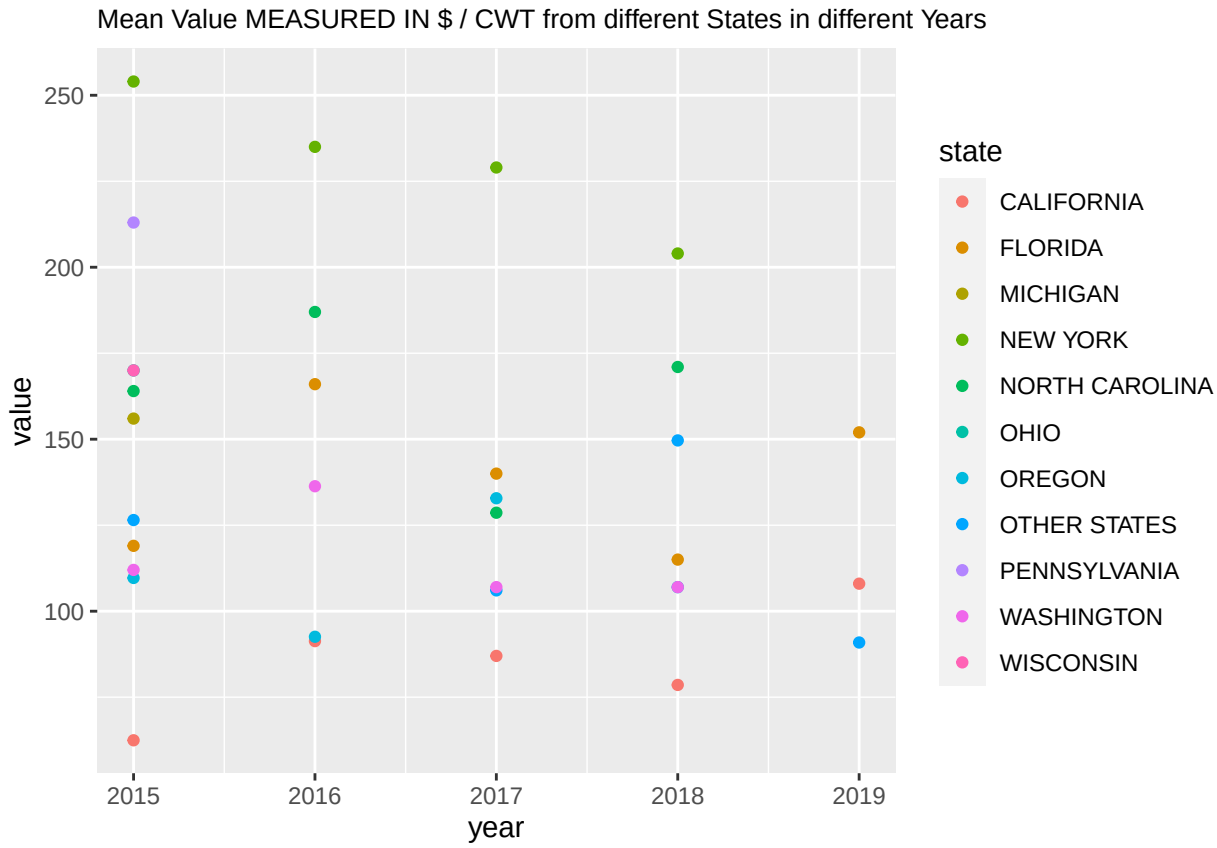
```
  strawberry_unit_desc_specific <- filter(strawberry_unit_desc_for_loop, unit_desc==dt[i])
  #replace the value=0 to NA
  strawberry_unit_desc_specific$value[strawberry_unit_desc_specific$value==0] <- NA
  #I would like to research values from different States in different Years
  #so I choose group by year and state.
  strawberry_unit_desc_specific_new <- group_by(strawberry_unit_desc_specific, year,state)
  strawberry_specific <- summarize(strawberry_unit_desc_specific_new, value= mean(value, na.rm=T))
  #plot the mean value from different States in different Years
  plot_list[[i]] <- ggplot(strawberry_specific, aes(x=year, y=value))+geom_point(aes(color=state))+
    ggtitle(paste("Mean Value",dt[i],"from different States in different Years"))+
    theme(plot.title = element_text( size = 10))+
    scale_x_continuous(breaks=c(2015,2016,2017,2018,2019))
}
```

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

There are 13 plots, and I choose three of them for showing.

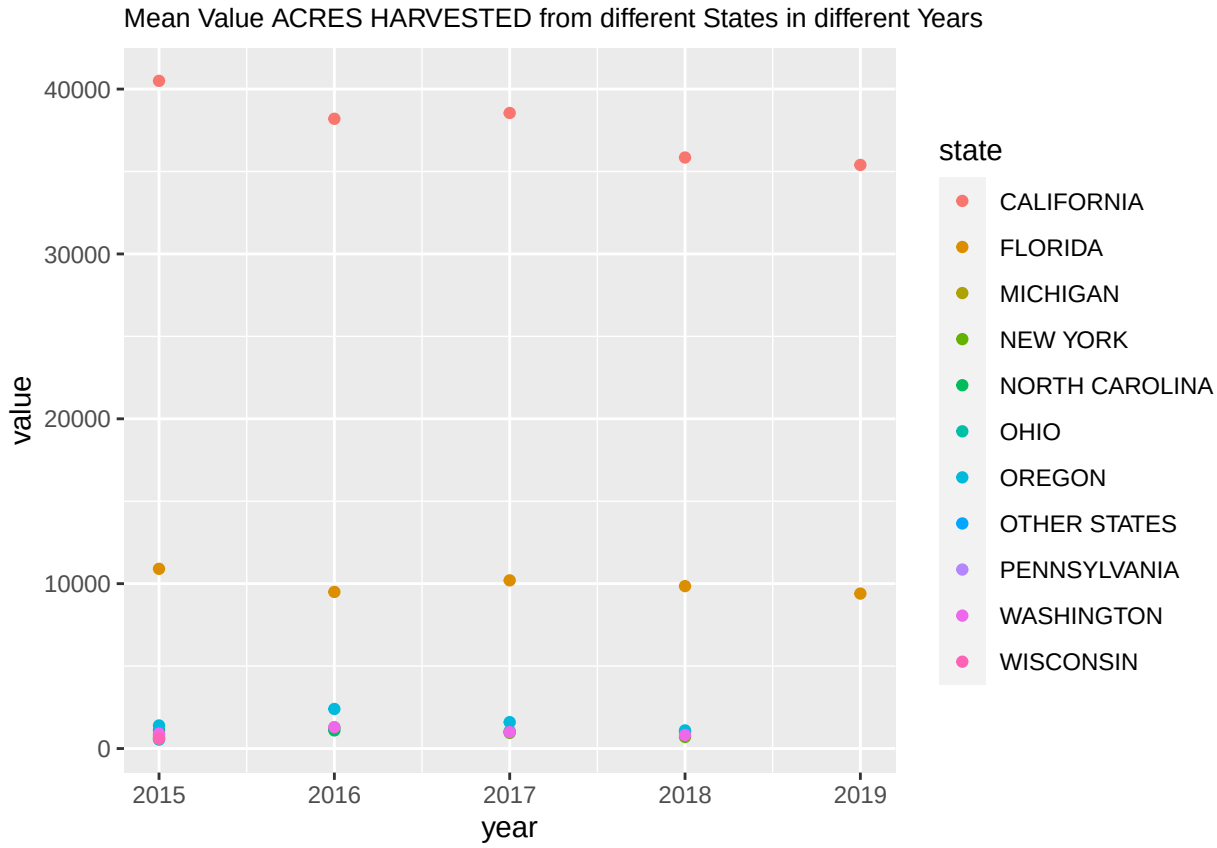
```
plot_list[[1]]
```

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



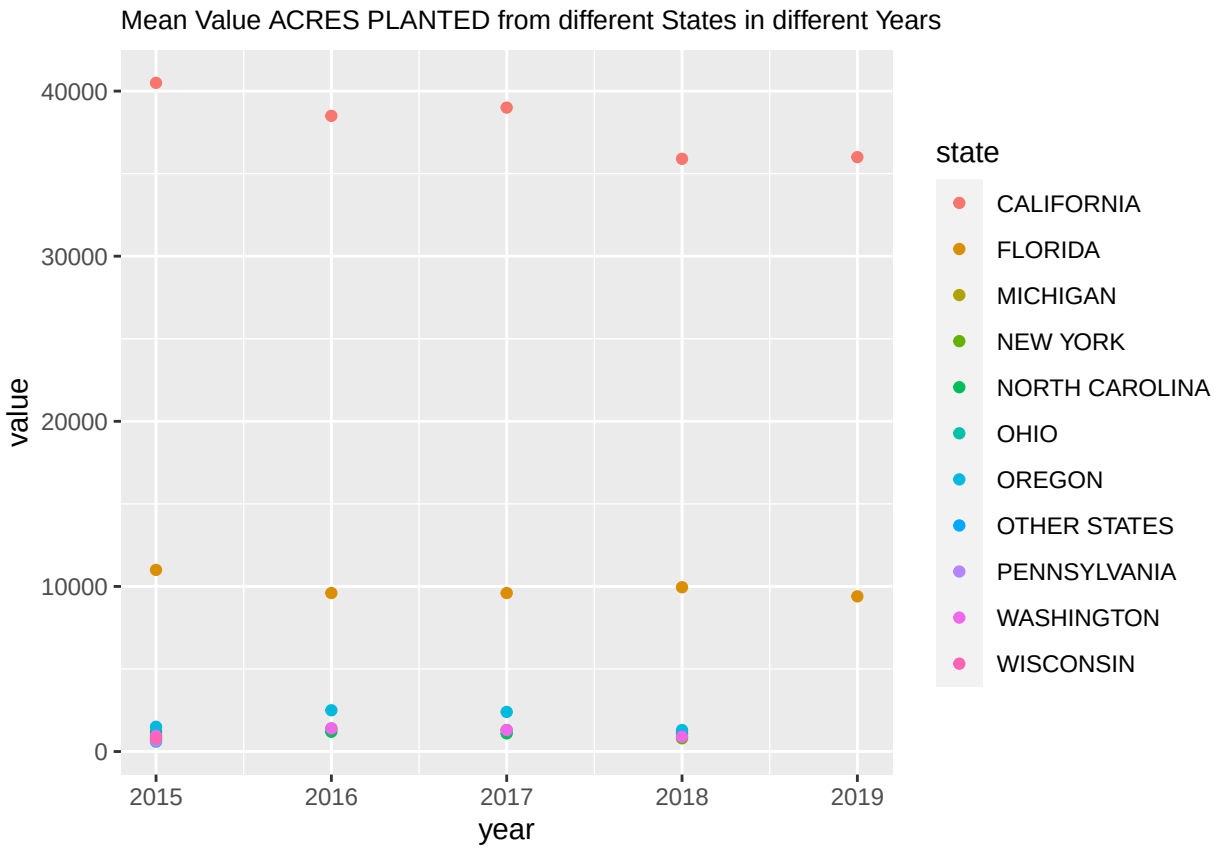

```
plot_list[[2]]
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
plot_list[[3]]
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



From the first picture, we can find the mean value in New York state is much higher than the other states, and I think probably because of its developed economics. And from the other pictures, we can find mean value in California is also higher than the other states.

(4)Mean value grouped by unit in years&domains

Now, I would like to research the relationship in years and domains grouped by unit.

At first, I also group by the unit, but I choose domain as variable.

```
strawberry_unit_desc_2 <- STRAWBERRIES %>% group_by(unit_desc) %>%
  summarize(
    domain=Domain,
    year=Year,
    numbers=n(),
    value=Value
  )
```

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

Then, we do similar as before.

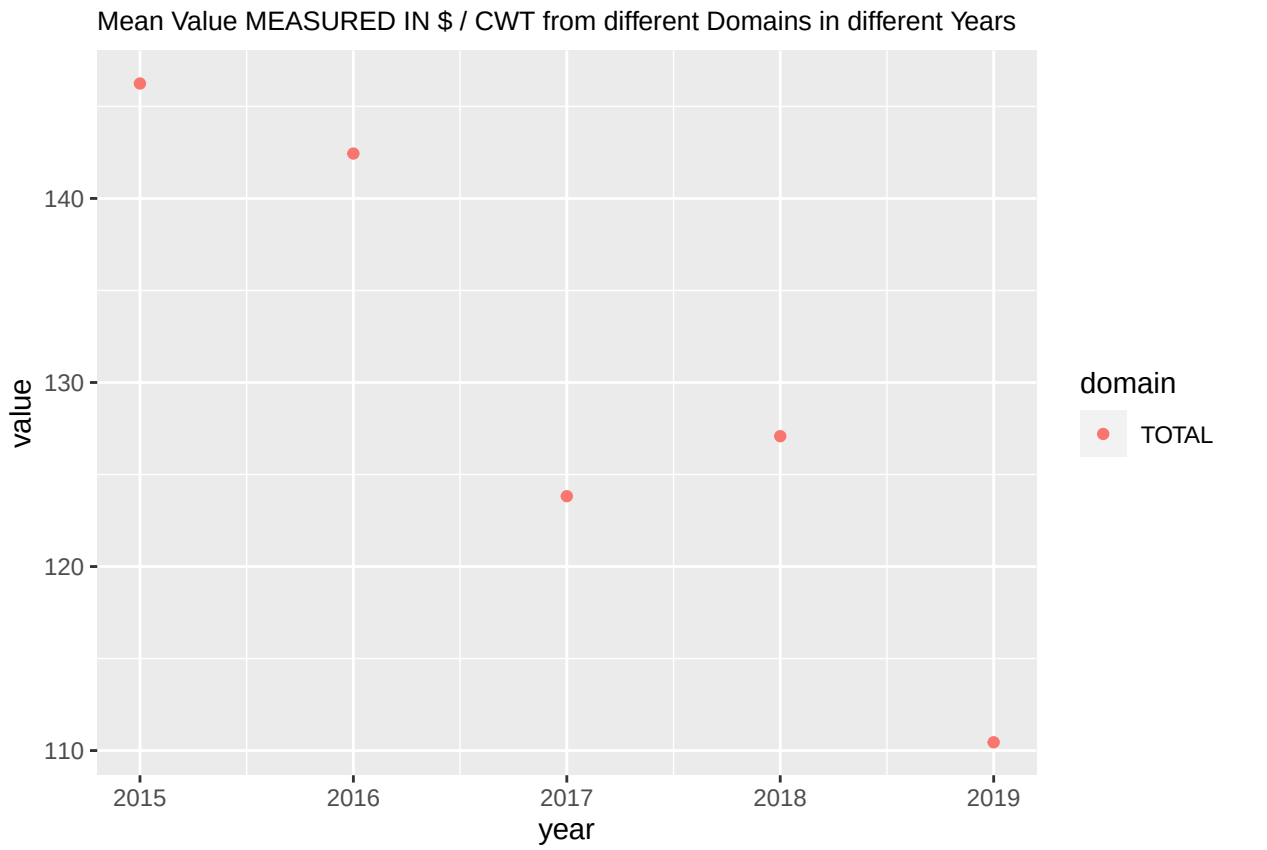
```
dt <- unique(strawberry_unit_desc$unit_desc)
dt <- do.call(rbind,dt)
## create the loop to perform 13 plots
plot_list2 <- list()
for(i in 1:length(dt)){
  #create flag variables
  strawberry_unit_desc_for_loop <- data.frame(strawberry_unit_desc_2)
  #extract specific unit from data set
```

```
  strawberry_unit_desc_specific <- filter(strawberry_unit_desc_for_loop, unit_desc==dt[i])
  #replace the value=0 to NA
  strawberry_unit_desc_specific$value[strawberry_unit_desc_specific$value==0] <- NA
  #I would like to research values from different Domains in different Years
  #so I choose group by year and domain.
  strawberry_unit_desc_specific_new <- group_by(strawberry_unit_desc_specific, year,domain)
  strawberry_specific <- summarize(strawberry_unit_desc_specific_new, value= mean(value, na.rm=T))
  #plot the mean value from different Domains in different Years
  plot_list2[[i]] <- ggplot(strawberry_specific, aes(x=year, y=value))+geom_point(aes(color=domain))+
    ggtitle(paste("Mean Value",dt[i],"from different Domains in different Years"))+
    theme(plot.title = element_text( size = 10))+
    scale_x_continuous(breaks=c(2015,2016,2017,2018,2019))
}
```

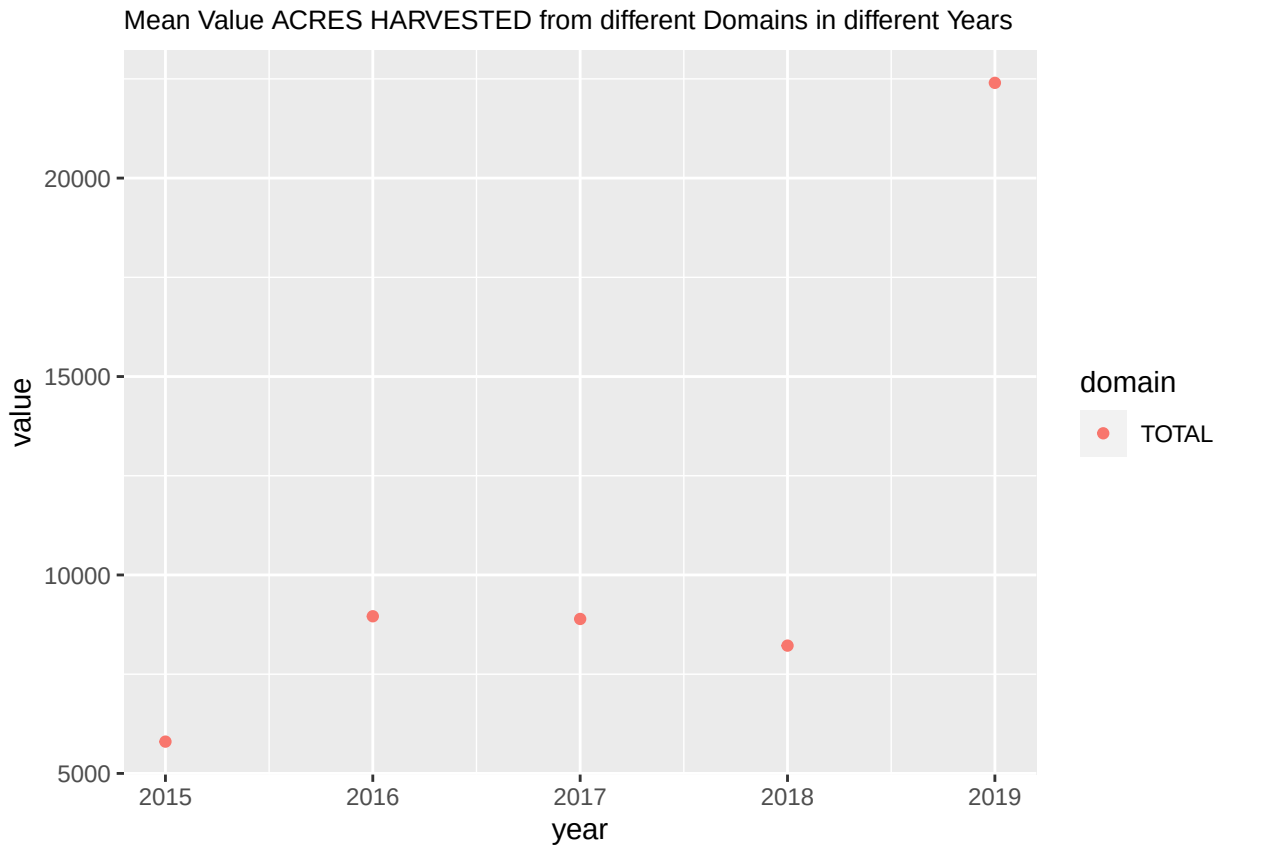
```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
## `summarise()` regrouping output by 'year' (override with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

As the same, I select 3 of them to show.

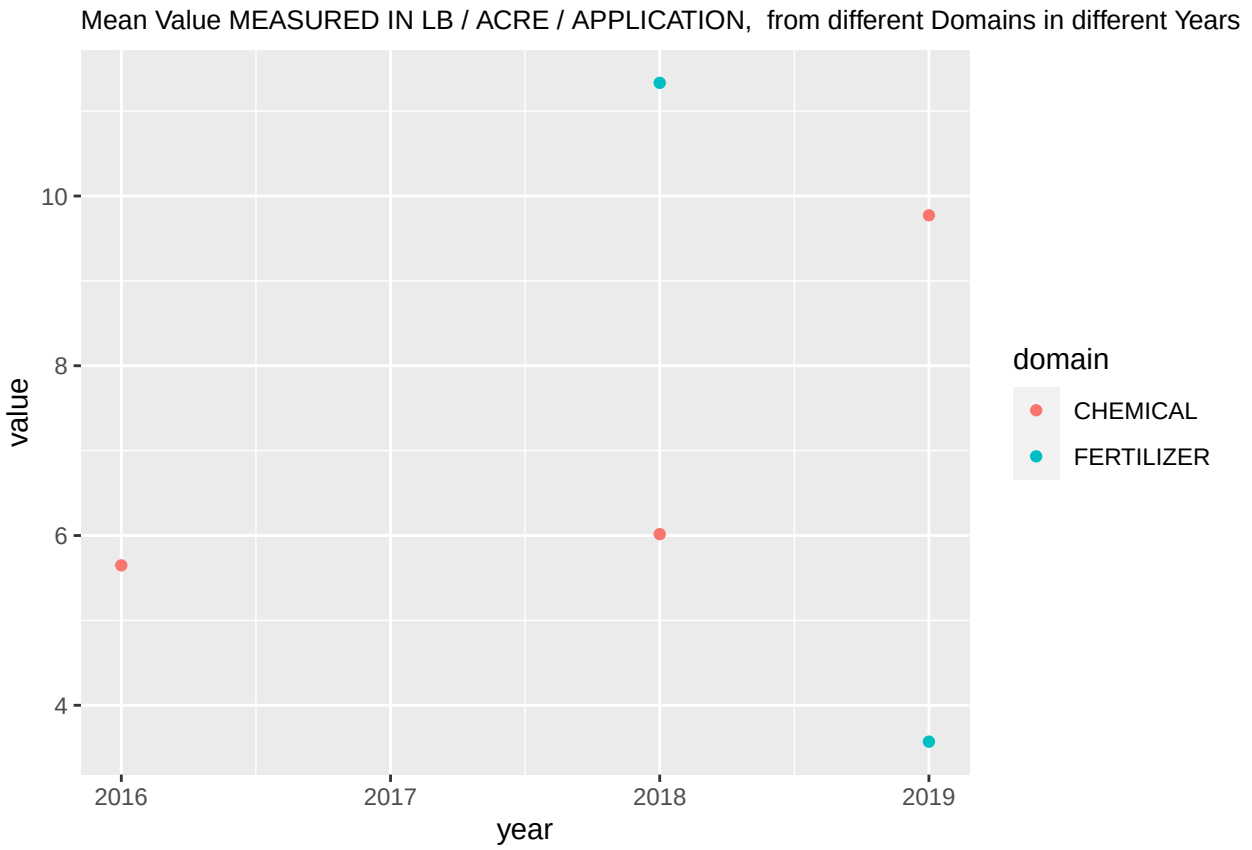
```
plot_list2[[1]]
```


Mean Value MEASURED IN $ / CWT from different Domains in different Years
value
140
130
120
110
2015
2016
2017
2018
2019
year
domain
TOTAL

```
plot_list2[[2]]
```


Mean Value ACRES HARVESTED from different Domains in different Years
value
20000
15000
10000
5000
2015
2016
2017
2018
2019
year
domain
TOTAL

```
plot_list2[[8]]
```



What I find:

At first, I find that total is used the most, that means in most situations, producing strawberries need both chemicals and fertilizers. Besides, I find that the value in 2019 is biggest in most situations.

(5)PCA

Principal Component analysis (PCA) is a multivariate statistical analysis method that USES linear transformation of multiple variables to select a small number of important variables. It is a common data dimension reduction method. Through linear transformation, the original N dimensions become the new N linearly independent principal components sorted by variance interpretation (PC).

```
strawberry_for_pca <- STRAWBERRIES %>% group_by(unit_desc,Domain.Detail) %>%
  summarize(
    year=Year,
    count=n(),
    value=Value
  )
```

```
## `summarise()` regrouping output by 'unit_desc', 'Domain.Detail' (override with `.groups` argument)
```

```
strawberry_for_pca2 <- filter(strawberry_for_pca,unit_desc == 'MEASURED IN LB')
strawberry_for_pca3 <- filter(strawberry_for_pca2,year==2019)

FUNGICIDE <- filter(strawberry_for_pca3,Domain.Detail==" FUNGICIDE")$value
HERBICIDE <- filter(strawberry_for_pca3,Domain.Detail==" HERBICIDE")$value
INSECTICIDE <- filter(strawberry_for_pca3,Domain.Detail==" INSECTICIDE")$value
OTHER <- filter(strawberry_for_pca3,Domain.Detail==" OTHER")$value
FERTILIZER <- filter(strawberry_for_pca3,Domain.Detail=="FERTILIZER")$value
```

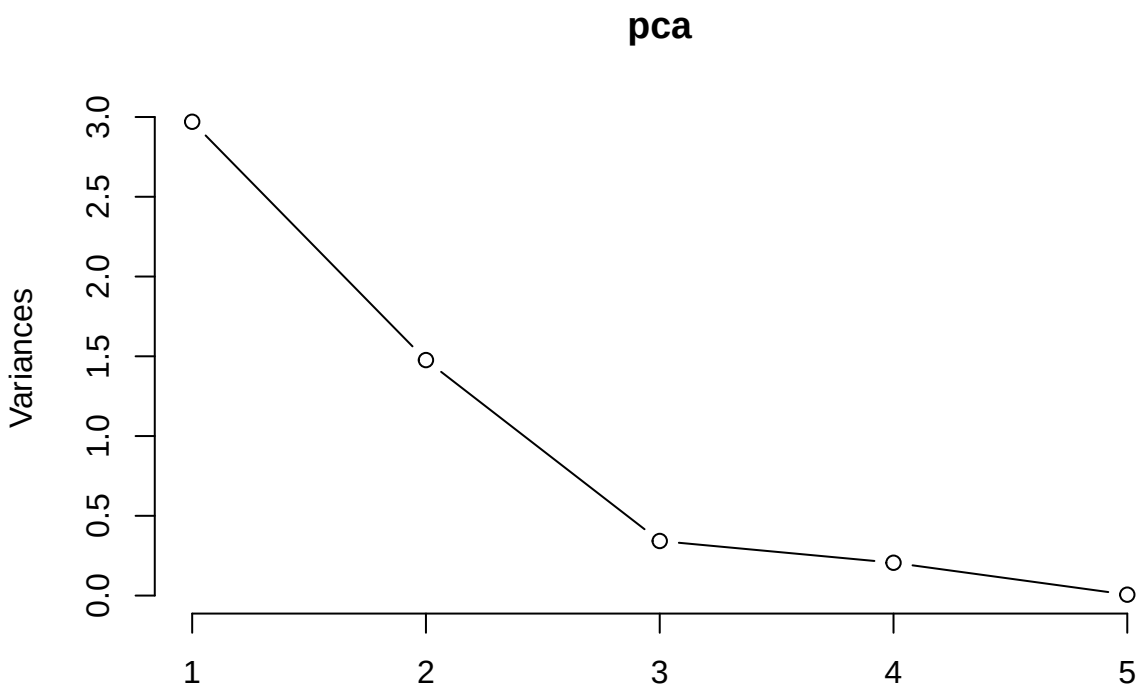
```r
pca_data <- data.frame(FUNGICIDE=FUNGICIDE[1:6],HERBICIDE=HERBICIDE[1:6],
 INSECTICIDE=INSECTICIDE[1:6],OTHER=OTHER[1:6],FERTILIZER=FERTILIZER[1:6])

pca <- prcomp(pca_data,center = T,scale. = T)
summary(pca,center = T,scale. = T)
```

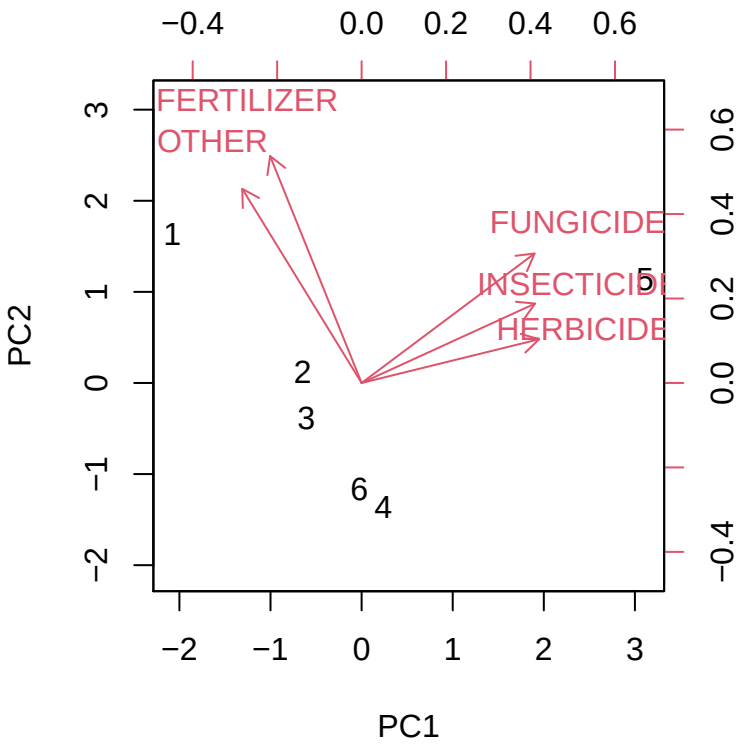
```
## Warning: In summary.prcomp(pca, center = T, scale. = T) :
##  extra arguments 'center', 'scale.' will be disregarded
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     1.723 1.2150 0.58494 0.45360 0.07523
## Proportion of Variance 0.594 0.2952 0.06843 0.04115 0.00113
## Cumulative Proportion  0.594 0.8893 0.95772 0.99887 1.00000
```

```r
plot(pca,type="line")
```




```r
biplot(pca,scale = 0)
```



In the pca, there are 5 components and the first three of them have large variances.

And in the biplot, the smaller angles between each variables, the less independent between them. So, for examples, we can see “FUNGICIDE”, “INSECTICIDE” and “HERBICIDE” have a positive correlation, and “HERBICIDE” and “FERTILIZER” have nealy no correlation.

4.Conclutions

Through this assignment, I completed data cleaning and organization and EDA, detail are as below.

(1)I have learned how to use package ‘stringr’ to process characters and strings such as deleting empty columns, separating strings into parts (two ways: using regular expression or loop) and changing types of variables.

(2)I have learned how to use package ‘srvyr’ to group by and summarize for plotting at last.

(3)I have learned how to use prcomp() to analyze data.

5.Reference

(1)R packages Documentation:tidyverse, magrittr, plyr, stringr, srvyr (2)Hadley Wickham & Garrett Grolemund(2017). R for data science (3)PCA Methods form: https://www.jianshu.com/p/8994afcaa757 (4)Data from USDA