MA615 Unit 1 Final Assignment

Rong Li

2020/10/19

1 Summary

In this assignment, we use the data set berries from USDA. We focus on one kind of berries–strawberries. Finish cleaning and organizing the data. Visualize and explore the data. This report can be divided into 2 parts: Data cleaning and EDA.

- Data cleaning: We select the data of strawberries from the raw data. Seperate some columns and eliminate the redundancy.
- EDA: We plot the boxplots for values of every year and of every states. Then we conduct the pca for values.

Besides, we deploy a shiny app Shiny_strawberries for data display.

2 Data cleaning

In this part, we obtain the data set strawberries that we will conduct EDA for later.

2.1 Deal with the raw data set berries

First, we explore the data set berries. Find out the redundent columns and delete them. Then display the data set.

```
## read the data
ag_data <- read_csv("/Users/amelia/Documents/mssp/MA615/Hw_berries/Berries_strawberries/data/berries.cs
```

```
## 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.
```

```
## look at number of unique values in each column
aa <- summarize_all(ag_data, n_distinct)
```

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

## list the 1-unique valu column names
colnames(ag_data)[bb]
```

```
##  [1] "Program"          "Week Ending"      "Geo Level"        "Ag District"
##  [5] "Ag District Code" "County"           "County ANSI"      "Zip Code"
##  [9] "Region"           "watershed_code"   "Watershed"        "CV (%)"
```

```
## remove the 1-unique columns from the dataset
ag_data <- select(ag_data, -all_of(bb))
aa <- select(aa, -all_of(bb))

## State name and the State ANSI code are (sort of) redundant
## Just keep the name
ag_data <- select(ag_data, -4)
aa <- select(aa, -4)

## Display the head of ag_data
head(ag_data)
```

```
## # A tibble: 6 x 8
##    Year Period  State  Commodity `Data Item`       Domain `Domain Categor~ Value
##   <dbl> <chr>   <chr>  <chr>     <chr>             <chr>  <chr>            <chr>
## 1  2019 MARKET~ CALIF~ BLUEBERR~ BLUEBERRIES, TAM~ TOTAL  NOT SPECIFIED    2.85
## 2  2019 MARKET~ CALIF~ BLUEBERR~ BLUEBERRIES, TAM~ TOTAL  NOT SPECIFIED    3.56
## 3  2019 MARKET~ CALIF~ BLUEBERR~ BLUEBERRIES, TAM~ TOTAL  NOT SPECIFIED    0.29
## 4  2019 MARKET~ CALIF~ RASPBERR~ RASPBERRIES - PR~ TOTAL  NOT SPECIFIED    2.69
## 5  2019 MARKET~ CALIF~ RASPBERR~ RASPBERRIES, FRE~ TOTAL  NOT SPECIFIED    (D)
## 6  2019 MARKET~ CALIF~ RASPBERR~ RASPBERRIES, PRO~ TOTAL  NOT SPECIFIED    (D)
```

2.2 Get the raw data of strawberries

Now, we focus on one kind of berries–strawberries.

```
## Strawberries
sberry <- filter(ag_data, (Commodity=="STRAWBERRIES") & (Period=="YEAR"))
sberry <- select(sberry, -c(Period, Commodity))
```

2.3 Separate the column 'Data Item' into 3 columns "type", "meas" and "what".

```
#### Does every Data Item begin with "STRAWBERRIES, "
sum(str_detect(sberry$`Data Item`, "^STRAWBERRIES, ")) == length(sberry$`Data Item`)
```

```
## [1] FALSE
```

```
## Seperate the Data Item
sberry1 <- subset(sberry, str_detect(sberry$`Data Item`, "^STRAWBERRIES, ") == "TRUE")
sberry2 <- subset(sberry, str_detect(sberry$`Data Item`, "^STRAWBERRIES, ") == "FALSE")
sberry1 <- separate(sberry1, `Data Item`, c("B","type", "meas", "what"), sep = ",")
sberry2 <- separate(sberry2, `Data Item`, c("B","todo"), sep = "-")
sberry2 <- separate(sberry2, `todo`, c("type","meas", "what"), sep = ",")
sberry <- rbind(sberry1, sberry2)
sberry <- select(sberry, -B)
```

2.4 Seperate the column “type” into 3 columns “type”, “lab1” and “lab2”.

```
## Seperate the type
sberry1 <- subset(sberry, str_detect(sberry$`type`, " - ") == "TRUE")
sberry1 <- separate(sberry1, type,c("type", "lab1", "lab2"), " - ")
sberry1 <- separate(sberry1, type,c("b1", "type"), " ")

sberry2 <- subset(sberry, type == " FRESH MARKET")
sberry2$type <- "FRESH MARKET"

sberry3 <- subset(sberry, type == " PROCESSING")
sberry3$type <- "PROCESSING"

sberry4 <- subset(sberry, str_detect(sberry$`type`, " - ") == "FALSE" &
                    type != " FRESH MARKET" & type != " PROCESSING")
sberry4 <- separate(sberry4, type,c("b1", "lab1", "lab2"), " ")

sberry <- plyr::rbind.fill(sberry1, sberry2, sberry3, sberry4)

sberry[is.na(sberry)] <- ""

sberry <- select(sberry, -b1)
## OK now Data Item has been split into parts
```

2.5 Separate the column “Domain” into 2 columns “D_left” and “D_right”.

```
## onto Domain
sberry <- separate(sberry, Domain, c("D_left", "D_right"), sep = ", ")

sberry[is.na(sberry)] <- ""
```

2.6 Separate the column “Domain Category” into 4 columns “DC_left_l”, “DC_left_r”, “DC_right_l”, “DC_right_r”.

```
## And now Domain Category
sberry <- separate(sberry, `Domain Category`, c("DC_left", "DC_right"), sep = ", ")

## unique(sberry$DC_left)
## unique(sberry$DC_right)

## looks like DC_left combines labels
sberry$DC_right[which(str_detect(sberry$DC_left,"138831"))]="INSECTICIDE: (CYFLUMETOFEN = 138831)"
sberry$DC_left[which(str_detect(sberry$DC_left,"138831"))]="CHEMICAL"


## work on DC_left first

sberry <- separate(sberry, DC_left, c("DC_left_l", "DC_left_r"), sep = ": ")

## sberry$DC_left_l %>% unique()
## sberry$DC_left_r %>% unique()

## now work on DC_right
```

```r
sberry <- separate(sberry, DC_right, c("DC_right_l", "DC_right_r"), sep = ": ")


sberry[is.na(sberry)] <- ""
```