MidTerm EDA

Tianjian Xie

2022-11-04

Data Input

```
strawb <- read_xlsx("D:/R dataset/615/Midterm/strawberries-2022oct30-a.xlsx", col_names = T)
```

Data Understanding

```
#Columns name and Index them
cnames <- colnames(strawb)
x <- 1:dim(strawb)[2]

#Unique values
unique(strawb[1])
```

```
## # A tibble: 2 x 1
##   Program
##   <chr>
## 1 CENSUS
## 2 SURVEY
```

```
unique(strawb[2])
```

```
## # A tibble: 6 x 1
##    Year
##   <dbl>
## 1  2019
## 2  2016
## 3  2021
## 4  2020
## 5  2018
## 6  2017
```

```
unique(strawb[3])
```

```
## # A tibble: 2 x 1
##   Period
##   <chr>
## 1 YEAR
## 2 MARKETING YEAR
```

```
#Collect number of unique rows in each column
T <- NULL
for(i in x){
  T <- c(T, dim(unique(strawb[i]))[1])
}

#Select drop columns
drop_cols <- cnames[which(T == 1)]

#Drop columns with only one unique row
strawb %<>%
  select(!all_of(drop_cols))

#Arrange the data frame by year and state.
strawb %<>%
  arrange(Year, State)

#New Columns names
colnames(strawb)
```

```
##  [1] "Program"         "Year"            "Period"          "State"
##  [5] "State ANSI"      "Data Item"       "Domain"          "Domain Category"
##  [9] "Value"           "CV (%)"
```

```
#Data Item Column
temp1 <- strawb %>%
  select(`Data Item`) %>%
         distinct()

#Separate to several columns
strawb %<>%
  separate(col=`Data Item`,
                    into = c("Strawberries", "type", "items", "units"),
                    sep = ",",
                    fill = "right")

#Chemicals in Domain Category Column
df_carbendazim <- grep("carbendazim",
                       strawb$`Domain Category`, ignore.case = T)
df_Bifenthrin <- grep("Bifenthrin",
                       strawb$`Domain Category`, ignore.case = T)
df_methyl_bromide <- grep("methyl bromide",
                      strawb$`Domain Category`, ignore.case = T)
df_1_3_dichloropropene <- grep("1,3-dichloropropene",
                          strawb$`Domain Category`,
                          ignore.case = T)
df_chloropicrin <- grep("chloropicrin",
                                strawb$`Domain Category`,
                                ignore.case = T)
df_Telone <- grep("Telone",
                       strawb$`Domain Category`,
                       ignore.case = T)
temp1 <- strawb %>% select(Strawberries) %>%
  distinct()

#Continue Clean Up data
pr_rec <- grep("STRAWBERRIES - PRICE RECEIVED",
                  strawb$Strawberries,
                  ignore.case = T)


#Split this analysis into organic and non organic -- and commercial vs chemicals
#Track down the Organic entries
type_organic <- grep("organic",
                strawb$type,
                ignore.case = T)
items_organic <- grep("organic",
                      strawb$items,
                      ignore.case = T)  ## nothing here
Domain_organic <- grep("organic",
                      strawb$Domain,
                      ignore.case = T)
Domain_Category_organic <- grep("organic",
                       strawb$`Domain Category`,
                       ignore.case = T)
```

```
#Create a Strawb_organic Tibble
same <- (intersect(type_organic, Domain_organic)==
         intersect(type_organic, Domain_organic))
length(same)==length(type_organic)
```

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

```
org_rows <- intersect(type_organic, Domain_organic)
strawb_organic <- strawb %>%
  slice(org_rows, preserve = FALSE)
strawb_non_organic <- strawb %>%
  filter(!row_number() %in% org_rows)

#Separate the Chemical data in non_organic
templ <- strawb_non_organic %>%
  select(type) %>%
  distinct()
chem_rows <- grep("BEARING - APPLICATIONS",
                      strawb_non_organic$type,
                      ignore.case = T)
chem_rows_1 <- grep("chemical",
                    strawb_non_organic$Domain,
                    ignore.case = T)
ins <- intersect(chem_rows, chem_rows_1)

#Examine the Domain Category Column
chem_rows_2 <- grep("chemical",
                     strawb_non_organic$`Domain Category`,
                     ignore.case = T)
ins_2 <- intersect(chem_rows, chem_rows_2)

#Create a Strawb_chem tibble
strawb_chem <- strawb_non_organic %>%
  slice(chem_rows, preserve = FALSE)

#Clean up the workspace before tackling the three tibbles just created.
rm(x, T, drop_cols, templ, df_carbendazim,
   df_Bifenthrin, df_methyl_bromide, df_1_3_dichloropropene,
   df_chloropicrin, df_Telone,
   pr_rec, type_organic, items_organic, Domain_organic,
   Domain_Category_organic, same, org_rows, chem_rows,
   chem_rows_1, chem_rows_2, ins, ins_2, cnames, i)

#Function that drop the "no info" columns
before_cols = colnames(strawb_chem)
  T = NULL
  x = length(before_cols)

   for(i in 1:x){
    b <- length(unlist(strawb_chem[,i] %>% unique()) )
    T <- c(T,b)
   }

  drop_cols <- before_cols[which(T == 1)]
  strawb_chem %<>%
    select(!all_of(drop_cols))
  after_cols = colnames(strawb_chem)

templ <- strawb_chem %>% select(units) %>% distinct()
```

```
## in units rows are either NA or AVG

strawb_chem %<>% separate(col=`Domain Category`,
                                  into = c("dc1", "chem_name"),
                                  sep = ":",
                                  fill = "right")
temp1 <- strawb_chem %>% select(chem_name) %>% unique()
length(unlist(temp1))
```

```
## [1] 172
```

```
aa  <- grep("measured in",
                   strawb_chem$items,
                   ignore.case = T)
length(aa)
```

```
## [1] 2112
```

```
#Drop State ANSI
strawb_chem %<>% select(Year, State, items, units, dc1, chem_name, Value)

#Rename Unites to Category
strawb_chem %<>% rename(category = units)

#Remove "MEASURED IN "
strawb_chem$items <- str_remove_all(strawb_chem$items, "MEASURED IN ")

#Rename items to units
strawb_chem %<>% rename(units = items)

#Check if all dc1 start with "Chemical,"
bb  <- grep("CHEMICAL, ",
            strawb_chem$dc1,
            ignore.case = T)
length(bb)
```

```
## [1] 2067
```

```
chem <- 1:2112

non_chem_rows <- setdiff(chem, bb)
length(non_chem_rows)
```

```
## [1] 45
```

```
temp1 <- strawb_chem %>% slice(non_chem_rows)
fertilizers <- temp1

#Clean up
rm(temp1, temps, temp3, aa, bb)
```

```
## Warning in rm(temp1, temps, temp3, aa, bb): object 'temps' not found
```

```
## Warning in rm(temp1, temps, temp3, aa, bb): object 'temp3' not found
```

```
#Remove "CHEMICAL, " from the entries in the dc1 and rename the column chem_types
strawb_chem$dc1 <- str_remove_all(strawb_chem$dc1, "CHEMICAL, ")
strawb_chem$dc1 %>% unique()
```

```
## [1] "FUNGICIDE"   "HERBICIDE"   "INSECTICIDE" "OTHER"       "FERTILIZER"
```

```
strawb_chem %<>% rename(chem_types = dc1)

#Get the units and categories sorted out
bb  <- grep("BIFENTHRIN",
            strawb_chem$chem_name,
            ignore.case = T)

bifen <- strawb_chem %>% slice(bb)
#Now fix the chem_name column and Remove the parens
strawb_chem$chem_name <- str_remove_all(strawb_chem$chem_name, "\\(")
strawb_chem$chem_name <- str_remove_all(strawb_chem$chem_name, "\\)")

#Separate chem_name and chem_code
strawb_chem %<>% separate(col = chem_name,
                          into = c("chem_name","chem_code"),
                          sep = "=",
                          fill = "right")
```

Compute a 95% confidence interval for California organic strawberry sales in 2016.

```
# California_Organic <- grep("CALIFORNIA",
#                                 strawb_organic$State,
#                                 ignore.case = T)
# Year_Organic <- grep("2016",
#                      strawb_organic$Year,
#                      ignore.case = T)
# ins <- intersect(California_Organic, Year_Organic)
# CA2016 <- as.integer(strawb_organic$Value[1:6])
# sample.sum <- sum(CA2016)
# sample.mean <- mean(CA2016)
# sample.n <- length(CA2016)
# sample.sd <- sd(CA2016)
# sample.se <- sample.sd/sqrt(sample.n)
# #T-score
# alpha = 0.5
# degrees.freedom <- sample.n - 1
# t.score <- qt(p=alpha/2, df=degrees.freedom,lower.tail=F)
# #CI
# margin.error <- t.score * sample.se
# lower.bound <- sample.mean - margin.error
# upper.bound <- sample.mean + margin.error
# print(c(lower.bound,upper.bound))
#CI in $
  # CV = sd/mean
  # mean = strawb_organic$Value[1]
  # sd = strawb_organic$`CV (%)`[1] * mean
  # 95%CI = (mean - 1.96 * sd, mean + 1.96 *sd)
231304956 - 1.96 * 0.137 * 231304956
```

```
## [1] 169194949
```

```
231304956 + 1.96 * 0.137 * 231304956
```

```
## [1] 293414963
```

Compute a 95% confidence interval for California non-organic strawberry sales in 2016.

```
# #Drop NAs and Ds
# NA_rows <- grep("(NA)",
#                         strawb_non_organic$Value,
#                         ignore.case = T)
# D_rows <- grep("(D)",
#                         strawb_non_organic$Value,
#                         ignore.case = T)
# California_non_Organic <- grep("CALIFORNIA",
#                                 strawb_non_organic$State,
#                                 ignore.case = T)
# Year_non_Organic <- grep("2016",
#                       strawb_non_organic$Year,
#                       ignore.case = T)
# ins <- intersect(California_non_Organic, Year_non_Organic)
# used_NA_rows <- intersect(ins, NA_rows)
# used_D_rows <- intersect(ins, D_rows)
# uncleaned_rows <- sort(c(used_NA_rows, used_D_rows), decreasing = FALSE)
# #Clean Dataset
# cleaned_rows <- setdiff(ins, uncleaned_rows)
# cleaned_strawb_non_organic <- strawb_non_organic %>% slice(cleaned_rows)
# CA2016_non <- as.integer(cleaned_strawb_non_organic$Value[1:172])
# sample.sum <- sum(CA2016_non)
# sample.mean2 <- mean(CA2016_non)
# sample.n2 <- length(CA2016_non)
# sample.sd2 <- sd(CA2016_non)
# sample.se2 <- sample.sd2/sqrt(sample.n2)
# #T-score
# alpha = 0.5
# degrees.freedom2 <- sample.n2 - 1
# t.score2 <- qt(p=alpha/2, df=degrees.freedom2,lower.tail=F)
# #CI
# margin.error2 <- t.score2 * sample.se2
# lower.bound2 <- sample.mean2 - margin.error2
# upper.bound2 <- sample.mean2 + margin.error2
# print(c(lower.bound2,upper.bound2))
```

##In the data set for the MA615 Strawberry project, how many different chemicals are listed?

```
# chemicals_types <- unique(strawb_chem$chem_types)
# length(chemicals_types)
chemical_names <- unique(strawb_chem$chem_name)
length(chemical_names)
```

```
## [1] 172
```

```
# unique(fertilizers$chem_name)
# unique(strawb_chem$chem_name)
length(chemical_names) + 3
```

```
## [1] 175
```

##On the basis of the data set for the MA615 Strawberry project, how many more chemicals have been used in California than in Florida?

```
CA_chem <- grep("CALIFORNIA",
                strawb_chem$State,
                ignore.case = T)
FL_chem <- grep("FLORIDA",
                strawb_chem$State,
                ignore.case = T)
CA_strawb_chem <- strawb_chem %>% slice(CA_chem)
FL_strawb_chem <- strawb_chem %>% slice(FL_chem)
CA_chemicals_types <- unique(CA_strawb_chem$chem_types)
CA_Types_Len <- length(CA_chemicals_types)
FL_chemicals_types <- unique(FL_strawb_chem$chem_types)
FL_Types_Len <- length(FL_chemicals_types)
Types_more <- CA_Types_Len - FL_Types_Len
Types_more
```

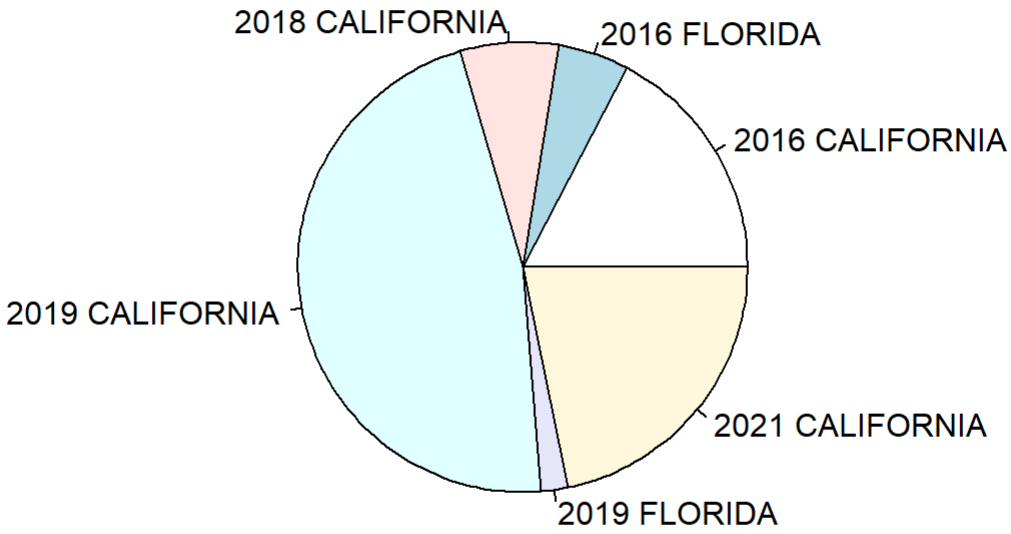
```
## [1] 0
```

```
CA_chemical_names <- unique(CA_strawb_chem$chem_name)
CA_Names_Len <- length(CA_chemical_names)
FL_chemical_names <- unique(FL_strawb_chem$chem_name)
FL_Names_Len <- length(FL_chemical_names)
Names_more <- CA_Names_Len - FL_Names_Len
Names_more
```

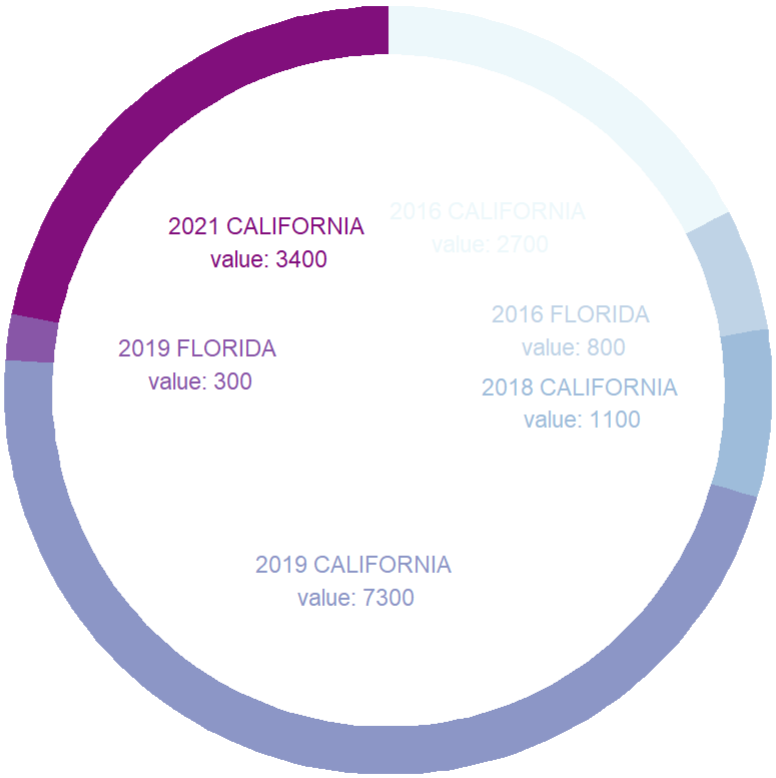
```
## [1] 23
```

##Bifenthrin detected

```
Bife_used <- grep("BIFENTHRIN",
                  strawb_chem$chem_name,
                  ignore.case = T)
df_Bifenthrin_used <- strawb_chem %>% slice(Bife_used)
Year <- c(df_Bifenthrin_used$Year[1],df_Bifenthrin_used$Year[4],df_Bifenthrin_used$Year[10],df_Bif
enthrin_used$Year[16],df_Bifenthrin_used$Year[19],df_Bifenthrin_used$Year[22])

State <- c(df_Bifenthrin_used$State[1],df_Bifenthrin_used$State[4],df_Bifenthrin_used$State[10],df
_Bifenthrin_used$State[16],df_Bifenthrin_used$State[19],df_Bifenthrin_used$State[22])

Value <- c(df_Bifenthrin_used$Value[1],df_Bifenthrin_used$Value[4],df_Bifenthrin_used$Value[10],df
_Bifenthrin_used$Value[16],df_Bifenthrin_used$Value[19],df_Bifenthrin_used$Value[22])

comb_Year_State <- paste(Year, State)

cldf_Bifenthrin_used <- data.frame(comb_Year_State,Value)


#Pie Plot of Usage of Bifenthrin by Year and State
pie(as.integer(Value), labels = comb_Year_State)
```



```
#Donut Chart of Biefenthrin by Year and State
sumvalue <- sum(as.integer(Value))
cldf_Bifenthrin_used$fractionvalue <- as.integer(Value) / sumvalue
cldf_Bifenthrin_used$ymax <- cumsum(cldf_Bifenthrin_used$fractionvalue)
cldf_Bifenthrin_used$ymin <- c(0, head(cldf_Bifenthrin_used$ymax, n=-1))
cldf_Bifenthrin_used$labelPosition <- (cldf_Bifenthrin_used$ymax + cldf_Bifenthrin_used$ymin) / 2
cldf_Bifenthrin_used$label <- paste0(cldf_Bifenthrin_used$comb_Year_State, "\n value: ", cldf_Bife
nthrin_used$Value)
ggplot(cldf_Bifenthrin_used, aes(ymax=ymax, ymin=ymin, xmax=7, xmin=6, fill=comb_Year_State)) +
  geom_rect() +
  geom_text( x=3, aes(y=labelPosition, label=label, color=comb_Year_State), size=3) + # x here con
trols label position (inner / outer)
  scale_fill_brewer(palette=3) +
  scale_color_brewer(palette=3) +
  coord_polar(theta="y") +
  xlim(c(-1, 7)) +
  theme_void() +
  theme(legend.position = "none")
```



##California non organic strawberry messured in LB/Year/Application

```
#Drop NAs and Ds
NA_rows <- grep("(NA)",
                      strawb_non_organic$Value,
                      ignore.case = T)
D_rows <- grep("(D)",
                      strawb_non_organic$Value,
                      ignore.case = T)
California_non_Organic <- grep("CALIFORNIA",
                               strawb_non_organic$State,
                               ignore.case = T)
Year_non_Organic <- grep("2016",
                     strawb_non_organic$Year,
                     ignore.case = T)
ins <- intersect(California_non_Organic, Year_non_Organic)
used_NA_rows <- intersect(ins, NA_rows)
used_D_rows <- intersect(ins, D_rows)
uncleaned_rows <- sort(c(used_NA_rows, used_D_rows), decreasing = FALSE)
#Clean Dataset
cleaned_rows <- setdiff(ins, uncleaned_rows)
cleaned_strawb_non_organic <- strawb_non_organic %>% slice(cleaned_rows)
LB_Year_App <- grep("MEASURED IN LB / ACRE / APPLICATION",
                    cleaned_strawb_non_organic$items,
                    ignore.case = T)
LB_Year_App_data <- cleaned_strawb_non_organic %>% slice(LB_Year_App)
#Circular Barplot
#Form new Dataset
lyadata <- data.frame(
  id <- seq(1,55),
  category <- c(LB_Year_App_data$`Domain Category`),
  value <- round(as.numeric(LB_Year_App_data$Value),4)
)
label_data <- lyadata
number_of_bar <- nrow(label_data)
angle <-  90 - 360 * (label_data$id-0.5) /number_of_bar
label_data$hjust<-ifelse( angle < -90, 1, 0)
label_data$angle<-ifelse(angle < -90, angle+180, angle)



ggplot(lyadata, aes(x=as.factor(id), y=value)) +
  geom_bar(stat="identity", fill=alpha("skyblue", 0.7)) +
  ylim(-100,120) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    plot.margin = unit(rep(-1,4), "cm")
  ) +
  coord_polar(start = 0) +
  geom_text(data=label_data, aes(x=id, y=value+10, label=category, hjust=hjust), color="black", fo
ntface="bold",alpha=0.6, size=1, angle= label_data$angle, inherit.aes = FALSE )
```

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

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