Berry Report

Yuxin Zeng

2020/10/19

###1.Introduction The NASS website has a large amount of berries data from many states in recent years, but these data are very messy, with mixed variables and different dimensions. After some simple cleaning and organization, I chose blueberry for EDA and present it in a shiny app.

###2.Data Peparation First of all, read in the data and remove those columns with single repeated values which are meaningless. "State ANSI" is also removed from the dataset since the "State" is clear enough. We find that berries data had only 8 out of 21 columns containing meaningful data. But some of the variables in these 8 columns are not clear enough, and we need to deal with them later. When it comes to "Period" column."Year" generally refers to calendar year while the definition of "Marketing year" varies by commodity. For Prices Received data, they both refers to an unweighted average (by month). Only consider the "YEAR" period. Filter out berries with specific numbers in "Value" in order to do further analysis. Divide the data into three parts according to the type of berries, so we get blueberries dataset, raspberries dataset, and strawberries dataset.

```
#Read the data
berries=read_csv("berries.csv",col_names=T)

#Remove columns with single value
col=berries%>%summarize_all(n_distinct)
single=which(col[1,]==1)
berries%<>%select(-all_of(single))
#Remove State ANSI
berries%<>%select(-4)
#Period="Year"
berries=berries%>%filter(Period=="YEAR")
#Filter out berries with specific numbers in "Value"
berries%<>%filter(Value!="(D)")
berries%<>%filter(Value!="(NA)")

#Group the data by commodity
bberry=berries%>%filter(Commodity=="BLUEBERRIES")
rberry=berries%>%filter(Commodity=="RASPBERRIES")
sberry=berries%>%filter(Commodity=="STRAWBERRIES")
```

Focus on blueberries. Separate multiple variables from the same column, merge variables of the same type into a new column, and delete duplicate columns.

```
#Separate "Data Item"
bberry%<>%separate('Data Item',c("B","type","meas","what"), sep=",")
bberry%<>%separate(type,c("b1","type","b2","lab1", "lab2"),sep="")
bberry%<>%mutate(label=paste(lab1,lab2))
bberry%<>%select(-c(B,b1,b2))

##Domain & Domain Category
```

```
bberry%<>%separate(Domain,c("D_left","D_right"),sep=",")
bberry%<>%mutate(D_left="CHEMICAL",D_left="")
bberry%<>%mutate(Chemical=paste(D_left,D_right))
bberry%<>%select(-c(D_left,D_right))

bberry%<>%separate('Domain Category',c("DC_left","DC_right"), sep=",")
bberry%<>%separate(DC_left,c("DC_left_l","DC_left_r"),sep=":")
bberry%<>%separate(DC_right,c("DC_right_l","DC_right_r"),sep=":")
bberry%<>%select(-c(DC_left_l,DC_right_l))

bberry%<>%select(Year,State,what,meas,label,Chemical,DC_left_r,DC_right_r,Value)
```

Some variables are not properly separated. We have entries in both the “what” and “meas” columns that begin with “MEASURED IN”. Separate them from their current columns and then merge them to unit column.

```
#Write a function
f1 <- function(a,b){
  if(a){
    return(b)
  }else{
    return("")
  }
}

f1_log=c(F,T,T)
f1_str=c("one","two","three")
map2(f1_log,f1_str,f1)
```

```
## [[1]]
## [1] ""
##
## [[2]]
## [1] "two"
##
## [[3]]
## [1] "three"
```

```
#Replace "NA" with blank before using the function
bberry[is.na(bberry)]=""

#"Meas"
detect.meas=str_detect(bberry$meas,"MEASURED IN")
bberry%<>%mutate(new_col1=unlist(map2(detect.meas,bberry$meas,f1)))
bberry%<>%mutate(meas=str_replace(bberry$meas,"MEASURED IN.*$", ""))

#"What"
detect.what=str_detect(bberry$what,"MEASURED IN")
bberry%<>%mutate(new_col2=unlist(map2(detect.what,bberry$what,f1)))
bberry%<>%mutate(what=str_replace(bberry$what,"MEASURED IN.*$", ""))

#Units
bberry%<>%mutate(units=str_trim(paste(new_col1,new_col2)))
```

Finally organize the columns and rename them.

```
#Rename the columns
bberry%<>%rename(Marketing=meas,Avg=what,Harvest=label,Chem_family=DC_left_r,Materials=DC_right_r,Measu
#Joint some columns
bberry%<>%mutate(production=str_trim(paste(Marketing,Harvest)))
bberry%<>%mutate(Chemical=str_trim(paste(Chem_family,Chemical)))

bberry%<>%select(Year,State,production,Avg,Measures,Materials,Chemical,Value)
write.csv(bberry,file="C:/Users/lenovo/Desktop/615 R/berry/bberry.csv")
```

###3.EDA Values are measured in different way, for example, some are measured in dollars, some are measured in LB. Only choose the blueberries that are measured in LB. Explore and visualize the relationship between Values and Year, State, production.

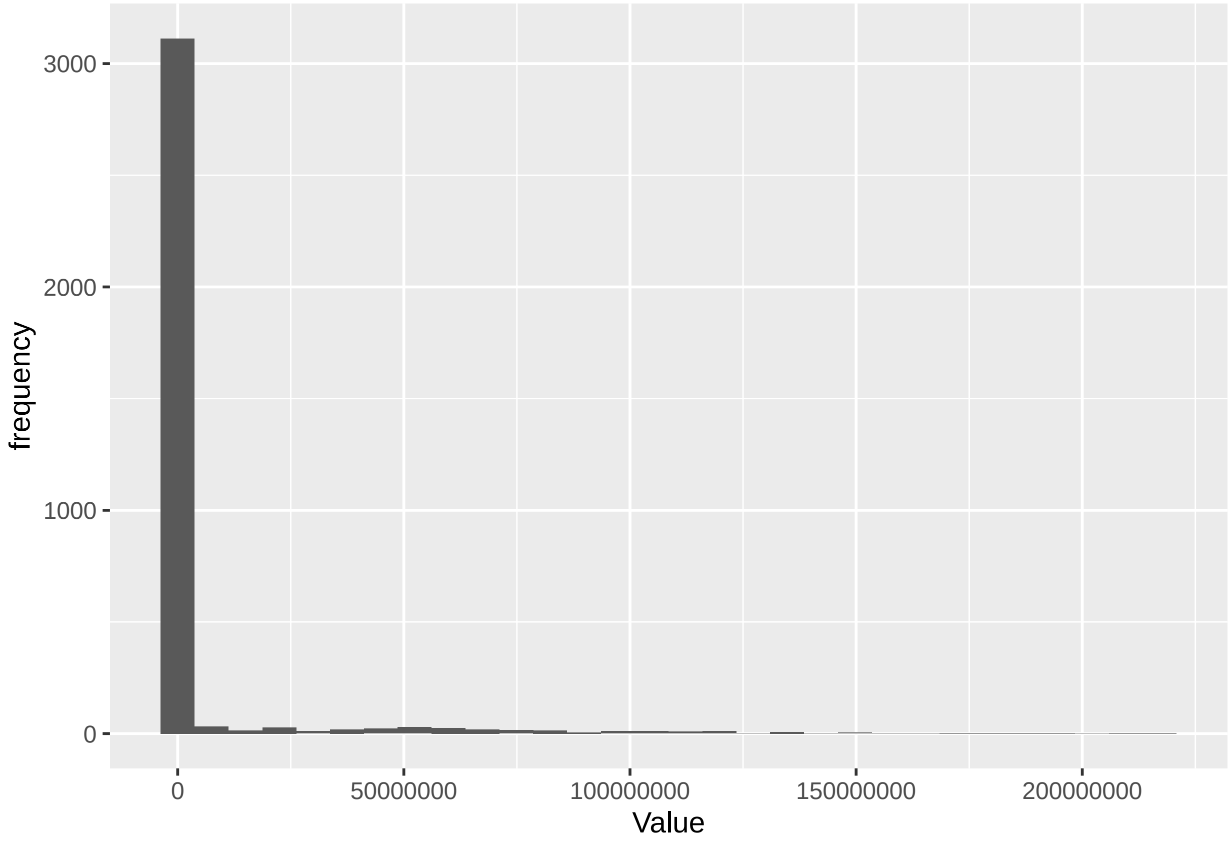
```
#Summary
options(scipen=200)
bberry$Value=as.numeric(gsub(",","",bberry$Value))
dim(bberry)
```

```
## [1] 3431    8
```

```
summary(bberry)
```

```
##       Year         State            production            Avg
##  Min.   :2015   Length:3431        Length:3431        Length:3431
##  1st Qu.:2015   Class :character   Class :character   Class :character
##  Median :2017   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2017
##  3rd Qu.:2019
##  Max.   :2019
##
##    Measures          Materials          Chemical              Value
##  Length:3431        Length:3431        Length:3431        Min.   :        0
##  Class :character   Class :character   Class :character   1st Qu.:        1
##  Mode  :character   Mode  :character   Mode  :character   Median :       10
##                                                           Mean   :  5363554
##                                                           3rd Qu.:     1300
##                                                           Max.   :217106000
##                                                           NA's   :22
```

```
p=qplot(x=Value,data=bberry,ylab='frequency')
p
```



```
#Choose ine measurement
sum(bberry$Measures=="MEASURED IN LB")
```

```
## [1] 866
```

```
sum(bberry$Measures=="MEASURED IN $")
```

```
## [1] 138
```

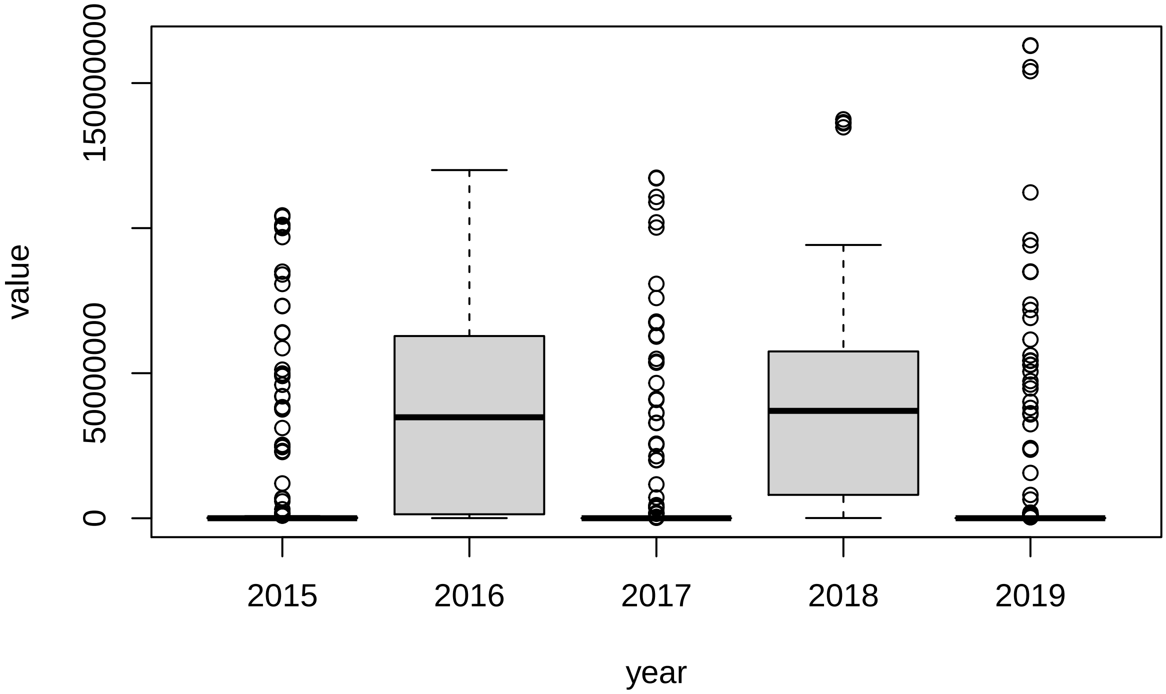
```
sum(bberry$Measures=="MEASURED IN LB / ACRE / APPLICATION")
```

```
## [1] 552
```

```
sum(bberry$Measures=="MEASURED IN LB / ACRE / YEAR")
```

```
## [1] 552
```

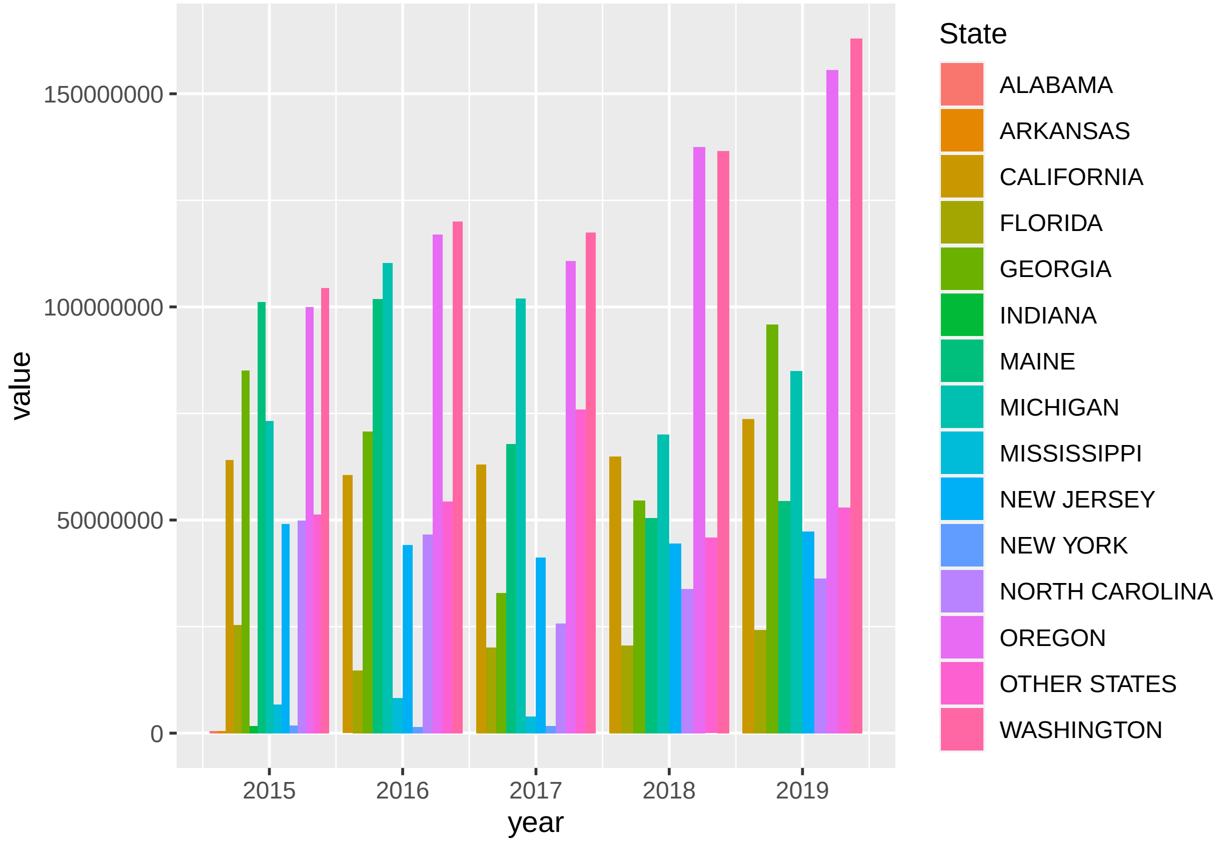
```
bberry%<>%filter(Measures=="MEASURED IN LB")

#Year
p1=boxplot(Value~Year,data=bberry,xlab='year',ylab='value')
```



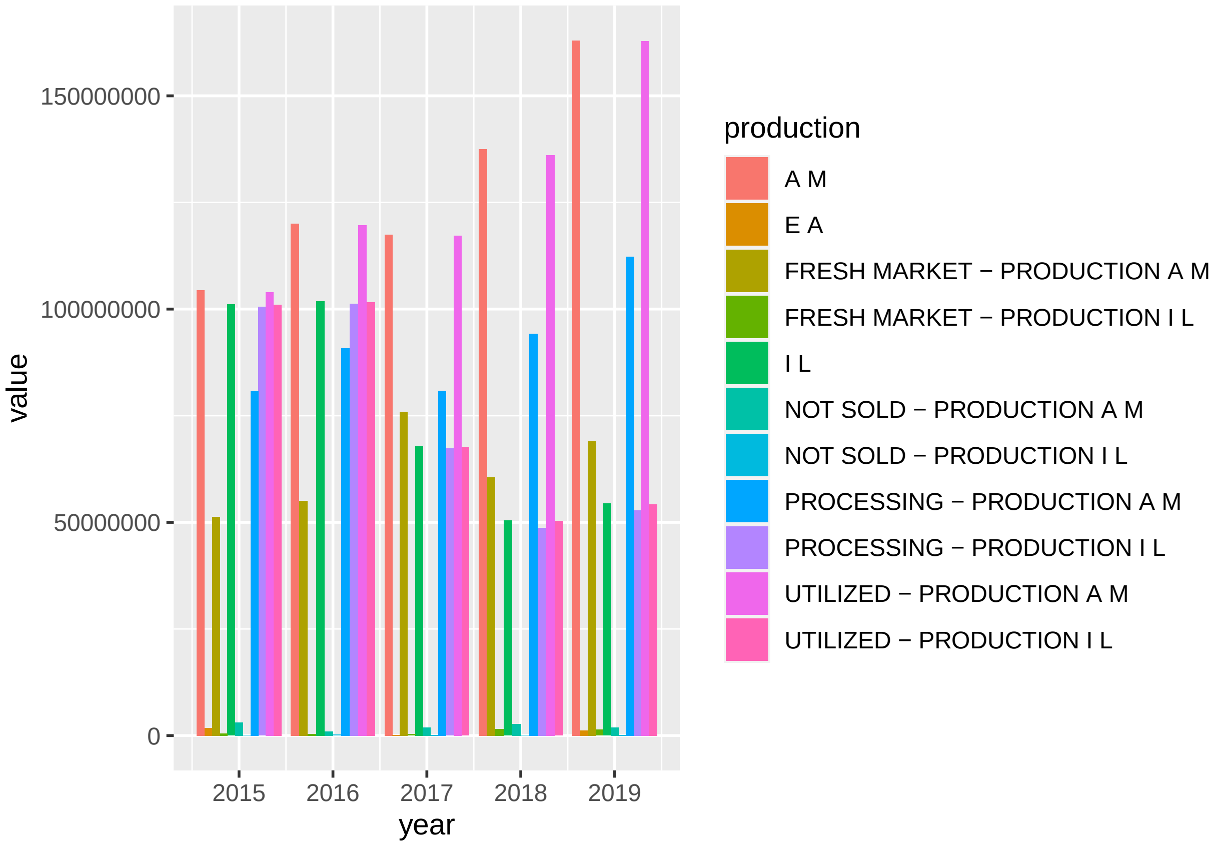
```
p1
```

```
## $stats
##        [,1]      [,2]   [,3]     [,4]   [,5]
## [1,]      0     40000    100    70000      0
## [2,]   1000   1370000   1100  8055000   1000
## [3,]   7450  34800000   3750 37010000   4000
## [4,] 303000  62800000  45800 57500000 123050
## [5,] 702000 120000000 107500 94190000 238000
##
## $n
## [1] 266  51 230  43 256
##
## $conf
##           [,1]     [,2]      [,3]     [,4]      [,5]
## [1,] -21806.56 21208966 -906.9405 25096339 -8052.438
## [2,]  36706.56 48391034 8406.9405 48923661 16052.438
##
## $out
##   [1]  64100000  63900000  25300000  24500000  24800000   1784000    906000
##   [8]   1124000  85000000  46000000   1000000  38000000  84000000   1610000
##  [15]   1600000 101110000 100500000 101000000    983000  73200000  42000000
##  [22]  31100000  73100000   6700000    900000   5800000  49080000  42100000
##  [29]   6930000  49030000   1790000   1720000  49900000  37500000  12000000
##  [36]  49500000 100000000  38300000   3100000  58600000  96900000  51220000
##  [43]  22820000 104400000  23200000  80750000 103950000  63030000    430000
##  [50]  62600000  20070000  19990000  32910000  25650000   7160000  32810000
##  [57]  67800000    350000    150000  67300000  67650000 102000000  53600000
##  [64]   1800000  46600000 100200000   3870000    420000   3450000  41180000
##  [71]  36250000    410000   4520000  40770000   1620000   1540000  25700000
##  [78]  21400000    370000   3930000  25330000 110780000  54950000   1880000
##  [85]  53950000 108900000  75910000  11670000 117380000  36350000    230000
##  [92]  80800000 117150000 137500000 134750000 136500000 136100000  73700000
##  [99]  56160000   1920000  15620000  71780000  24200000    580000  23620000
## [106]    667000    567000    612000  95900000  61570000   1920000  32410000
## [113]  93980000  54400000   1410000  52820000  54230000    973000    894000
## [120]    426000  84900000  44740000  40160000  84900000    540000    423000
## [127]    557000  47300000  38030000   1230000   8040000  46070000    437000
## [134]    587000    331000  36200000    430000  35770000    677000    566000
## [141]    392000 155500000  69040000   1400000  85060000 154100000  52940000
## [148]   6450000    902000   1179000   1090000    794000 163000000  50530000
## [155] 112300000 162830000
## 
## $group
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [112] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [149] 5 5 5 5 5 5 5 5
## 
## $names
## [1] "2015" "2016" "2017" "2018" "2019"
```

```
#State
p2=ggplot(bberry,aes(x=Year,y=Value,fill=State))+geom_bar(position="dodge",stat="identity")+xlab("year"
p2
```



```
#Production
p3=ggplot(bberry,aes(x=Year,y=Value,fill=production))+geom_bar(position="dodge",stat="identity")+xlab("
p3
```



###4.Reference

[1] National Agricultural Statistics Service

[2] Visit Maine

[3] Vince Vu