Unit1 Final Assignment Berries

Hao Shen

2020/10/17

1 Summary

In this assignment, we do EAD for data set berries from USDA. In data cleaning process, we mainly separate mixed stings into several variables and delete variables with meaningless or replicated information. Then, according to data set attributions, we divided the whole date set into two parts:

- Bearing: a part contains information about farm chemical and fertilizer usage on berries.
- Market: a part with price, yield, area related information about berries.

For Bearing part, we use principal component analysis to explore variables' internal relationship and the results shows the PCA needs 4 principal components.

For Market part, since the relationships between variables are extremely simple, we summarize 13 equations to explain their connections.

Besides, we deploy a shiny app Berries-shiny for data display and a slide Berries-slide to present the results.

2 Data cleaning

After initially rough observation of the berries data set downloaded from USDA 1 , we can find it generally consisted by two parts:

- Bearing: one mainly about soil bearings on berries including farm chemical bearings like fungicide, herbicide, etc. and fertilizer bearings including nitrogen, potash, etc.
- Market: one contains market related data about berries including price, yield, acre, etc.

So, we believe the whole data sets should be separated into two part and analyzed respectively. Here we just use data of 'BLUEBERRIES' for example.

2.1 Select columns with useful information

First, we load data from local disk and filter those lines related to 'BLUBERRIES'. Then we drop some columns for the information they contained is either meaningless or replicated, and get a data frame of 7,597 observations with 7 variables summarized as follow:

```r
berries=read.csv('Berries-data/berries.csv')
b_berry=filter(berries,Commodity=='BLUEBERRIES')
b_berry=select(b_berry,
               -Program,
               -(Period:Geo.Level),
               -(State.ANSI:Watershed),
               -CV....)
summary(b_berry)
```

```
##       Year          State             Commodity           Data.Item
##  Min.   :2015   Length:7597        Length:7597        Length:7597
##  1st Qu.:2015   Class :character   Class :character   Class :character
##  Median :2017   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2017
##  3rd Qu.:2019
##  Max.   :2019
##     Domain          Domain.Category       Value
##  Length:7597        Length:7597        Length:7597
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

2.2 Separate 'b__berry' into two tables

Now, we start to separate the whole data set into two tables bearing and market as mentioned above by detecting whether 'BEARING' occurred in Data.Item.

Then in exploring remaining data or so called market related data, it is interesting to find 'BLUEBERRIES' from Maine State are labeled with WILD while others are all 'TAME'. This may caused by 'the wild blueberry, sometimes called the low-bush blueberry, is native to Maine and thrives in its glacier-churned soil and challenging seasons.' 2 However, for convenience, we will only label it in a new TAME column by 1 or 0 and ingore it in further analysis.

```
#write.csv(b_berry,'b_berry.csv',row.names=F)
#unique(b_berry$Data.Item)
b_berry_bearing=b_berry[str_detect(b_berry$Data.Item,'BEARING'),]
b_berry_market=b_berry[!str_detect(b_berry$Data.Item,'BEARING'),]
b_berry_market$TAME[str_detect(b_berry_market$Data.Item,'TAME')]=1
b_berry_market$TAME[str_detect(b_berry_market$Data.Item,'WILD')]=0
```

2.3 Create table1 'b__berry__bearing'

Here we come into the most complicated part to separate Data.Item, for each of them includes at least 3 parts of information with only 1 string like 'BLUEBERRIES, BEARING - APPLICATIONS, MEASURED IN LB'

- Commodity='BLUEBERRIES': a part contains the same information of Commodity column which can be deleted directly.
- Measurement='BEARING - APPLICATIONS': a part denote the measurement used to generate Values. It can be either 'TREATED' or 'APPLICATION'.
- Unit='MEASURED IN LB': a part denote the unit of Values. For observations with 'APPLICATION', units include LB, LB/ACRE/APPLICATION, etc. For observations with 'TREATED', unit only include percentage.

Since both comma and space contained in strings and varies from observation to observation, it is hard to use normal separator link ',' and ' '. And we believe it can be solved by plugging our own separator'#' into the right places, and just split strings by these new separators.

```
#unique(b_berry_bearing$Data.Item)
b_berry_bearing$Data.Item=str_replace_all(b_berry_bearing$Data.Item,
                                          c('BLUEBERRIES, BEARING - '='',
                                            ', MEASURED IN '='#'))
b_berry_bearing=separate(b_berry_bearing,Data.Item,c('Measurement','Unit'),'#')
```

After separation of Data.Item, we notice that observations with 'CHEMICAL' Domain included are different from those with 'FERTILIZER' in both Domain and Domain.Category columns. Therefore in next steps we need to separate the whole bearing table into two sub-tables and handle their data respectively.

```
#unique(b_berry_bearing$Domain)
b_berry_chemical=b_berry_bearing[
  str_detect(b_berry_bearing$Domain,'CHEMICAL'),]
b_berry_fertilizer=b_berry_bearing[
  str_detect(b_berry_bearing$Domain,'FERTILIZER'),]
```

2.3.1 Create sub-table 'b_berry_chemical'

For sub-table with 'CHEMICAL' Domain, it has a Domain.Category like 'CHEMICAL, INSECTICIDE: (PHOSMET = 59201)', which contains two parts:

- Domain='CHEMICAL, INSECTICIDE': more specific information than Domain about chemical types.
- Category='(PHOSMET = 59201)": a part with information of chemical components.

So, we decide to just delete original Domain column and separate the Domain.Category into two part, Domain and Category columns to gain contain more information with just 2 columns.

```
b_berry_chemical=select(b_berry_chemical,-Domain)
#unique(b_berry_chemical$Domain.Category)
b_berry_chemical$Domain.Category=str_replace_all(b_berry_chemical$Domain.Category,
                                                 c('CHEMICAL, '='',
                                                   ': '='#'))
b_berry_chemical=separate(b_berry_chemical,Domain.Category,c('Domain','Category'),'#')
```

2.3.2 Create sub-table 'b_berry_fertilizer'

For sub-table with 'FERTILIZER' Domain, its Domain.Category is like 'FERTILIZER: (SULFUR)' which also has two parts:

- Domain='FERTILIZER': a part contains the same information as Domain column.
- Category='(SULFUR)": a part with information of chemical components.

So, we can also delete original Domain column and separate the Domain.Category into two parts.

```
b_berry_fertilizer=select(b_berry_fertilizer,-Domain)
#unique(b_berry_fertilizer$Domain.Category)
b_berry_fertilizer=separate(b_berry_fertilizer,Domain.Category,c('Domain','Category'),': ')
```

2.3.3 Form new table1 'b_berry_bearing'

Now, we can combine these two sub-tables to form the new bearing table. It is displayed and summarized as follow:

```
b_berry_bearing=rbind(b_berry_chemical,b_berry_fertilizer)
head(b_berry_bearing)
```

```
##    Year   State   Commodity  Measurement Unit    Domain
## 50 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
## 51 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
## 52 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
## 53 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
## 54 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
## 55 2019 GEORGIA BLUEBERRIES APPLICATIONS   LB FUNGICIDE
##                                           Category  Value
## 50                           (AZOXYSTROBIN = 128810)  1,800
## 51 (BACILLUS AMYLOLIQUEFACIENS STRAIN D747 = 16482)   (NA)
## 52                          (BACILLUS SUBTILIS = 6479)   (NA)
## 53                        (BORAX DECAHYDRATE = 11102)    (D)
## 54                                 (BOSCALID = 128008)  2,300
## 55                      (CALCIUM POLYSULFIDE = 76702) 20,200
```

```
summary(b_berry_bearing)
```

```
##       Year         State             Commodity          Measurement       
##  Min.   :2015   Length:6834        Length:6834        Length:6834       
##  1st Qu.:2015   Class :character   Class :character   Class :character  
##  Median :2017   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :2017                                                           
##  3rd Qu.:2019                                                           
##  Max.   :2019                                                           
##      Unit              Domain            Category            Value          
##  Length:6834        Length:6834        Length:6834        Length:6834       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
## 
## 
## 
```

2.4Create table2 'b__berry__market'

Basically, the creation of table2 'b__berry__market' is the same as the creation of table1, with just a little changes in handling Data.Item column.

Here the most rows of Data.Item column have the same format of table1 'b__berry__chemical', consisted by Commodity, Measurement and Unit. But for some rows with Data.Item equal contains 'ACRES HARVESTED', it only have two parts of information, Commodity and Measurement. But it is early to handle, since if we separate them as others, the new Unit column will be filled by NA, which can be easily replaced by its real unit 'ACRE'.

After cleaning, it is displayed and summarized as follow:

```
b_berry_market=select(b_berry_market,-Domain,-Domain.Category)
#unique(b_berry_market$Data.Item)
b_berry_market$Data.Item=str_replace_all(b_berry_market$Data.Item,
                                       c('BLUEBERRIES, TAME'='#',
                                         'BLUEBERRIES, WILD'='#',
                                         ', MEASURED IN '='#'))
b_berry_market$Data.Item=str_replace_all(b_berry_market$Data.Item,
                                       c('# - '='',
                                         '#, '=''))
b_berry_market=separate(b_berry_market,Data.Item,c('Measurement','Unit'),'#')
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 54 rows [30, 40,
## 50, 60, 70, 80, 90, 100, 114, 153, 163, 173, 183, 193, 203, 213, 223, 237, 285,
## 295, ...].
```

```
#b_berry_market[is.na(b_berry_market$Unit),]
b_berry_market$Unit[is.na(b_berry_market$Unit)]='ACRE'
head(b_berry_market)
```

```
##   Year      State   Commodity                    Measurement   Unit Value TAME
## 1 2019 CALIFORNIA BLUEBERRIES                 PRICE RECEIVED $ / LB  2.85    1
## 2 2019 CALIFORNIA BLUEBERRIES FRESH MARKET - PRICE RECEIVED $ / LB  3.56    1
## 3 2019 CALIFORNIA BLUEBERRIES   PROCESSING - PRICE RECEIVED $ / LB  0.29    1
## 4 2019    FLORIDA BLUEBERRIES                 PRICE RECEIVED $ / LB  2.64    1
## 5 2019    FLORIDA BLUEBERRIES FRESH MARKET - PRICE RECEIVED $ / LB   (D)    1
## 6 2019    FLORIDA BLUEBERRIES   PROCESSING - PRICE RECEIVED $ / LB   (D)    1
```

```r
summary(b_berry_market)
```

```
##       Year         State             Commodity          Measurement       
##  Min.   :2015   Length:763         Length:763         Length:763        
##  1st Qu.:2015   Class :character   Class :character   Class :character  
##  Median :2017   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :2017                                                           
##  3rd Qu.:2018                                                           
##  Max.   :2019                                                           
##      Unit              Value                TAME       
##  Length:763         Length:763         Min.   :0.0000  
##  Class :character   Class :character   1st Qu.:1.0000  
##  Mode  :character   Mode  :character   Median :1.0000  
##                                        Mean   :0.9122  
##                                        3rd Qu.:1.0000  
##                                        Max.   :1.0000  
```

2.5 Remove useless b__berries and save data

We finally get two data sets of bearings and market after all above processes and we need to save them at two place: one in Berries-data folder for further EDA and Slide making, one in Berries-shiny for Shiny app to deploy.

```r
rm(b_berry_fertilizer,b_berry_chemical)
#for EDA & Slide
write.csv(b_berry_bearing,"Berries-data/b_berry_bearing.csv",row.names=F)
write.csv(b_berry_market,"Berries-data/b_berry_market.csv",row.names=F)
#for Shiny
write.csv(b_berry_bearing,"Berries-shiny/b_berry_bearing.csv",row.names=F)
write.csv(b_berry_market,"Berries-shiny/b_berry_market.csv",row.names=F)
```

3 EDA for ‘bearing’

EDA for ‘bearing’, especially for the chemical parts of bearing is extremely hard for a normal person since we do not have so much understanding of various farm chemicals. However, we can still analyze parts of them like total FUNGICIDE, HERBICIDE, INSECTICIDE, Others and total NITROGEN, PHOSPHATE, POTASH, SULFUR. Here we just use Principal Component Analysis (PCA) to analyze their relationships. Since the FERRTILIZER data only collected in 2015 and 2019, we will choose these two years only. Besides, for missing value, we use mean value to replace.

```r
# load data
bearing=read.csv('Berries-data/b_berry_bearing.csv')
bearing_value=filter(bearing,
                     Year%in%c('2019','2015'),
                     Unit=='LB',
                     Category%in%c('(NITROGEN)','(PHOSPHATE)','(POTASH)',
                                   '(SULFUR)','(TOTAL)')
                     )
# arrange data
bearing_value$DC=paste(bearing_value$Domain,bearing_value$Category)
```

```
bearing_value=select(bearing_value,DC,Value)
bearing_value=arrange(bearing_value,bearing_value[,1])
# handle missing value
for(i in unique(bearing_value$DC)){
  m=bearing_value$Value[bearing_value$Value!=' (D)'
                        &bearing_value$DC==i]%>%
    str_replace_all(c(','=''))%>%
    as.numeric()%>%
    mean()
  bearing_value$Value[bearing_value$Value==' (D)'
                    &bearing_value$DC==i]=m

}
# transform to numeric
bearing_value$Value=as.numeric(str_replace_all(bearing_value$Value,c(','='')))
# transform for PCA
j=1
for(i in unique(bearing_value$DC)){
  if(j==1)bearing_pca=bearing_value$Value[bearing_value$DC==i]
  else bearing_pca=cbind(bearing_pca,bearing_value$Value[bearing_value$DC==i])
  j=0
}
colnames(bearing_pca)=unique(bearing_value$DC)
head(bearing_pca)
```

```
##      FERTILIZER (NITROGEN) FERTILIZER (PHOSPHATE) FERTILIZER (POTASH)
## [1,]                667000                 567000              612000
## [2,]                973000                 238000              894000
## [3,]                540000                 423000              557000
## [4,]                437000                 587000              331000
## [5,]                677000                 566000              392000
## [6,]                902000                1179000             1090000
##      FERTILIZER (SULFUR) FUNGICIDE (TOTAL) HERBICIDE (TOTAL)
## [1,]              125000             91400             29800
## [2,]              426000            121100             50900
## [3,]              297625             36300             32700
## [4,]              297625             23000              6500
## [5,]              117000             48900             24900
## [6,]              794000            126500             39700
##      INSECTICIDE (TOTAL) OTHER (TOTAL)
## [1,]               19300          3700
## [2,]               60600         29600
## [3,]               19800           200
## [4,]               14900          9100
## [5,]               16400         24100
## [6,]               43000         21000
```

After getting the data, we use prcomp function for PCA and summary its components, cumulative proportion and plot 3 as follow:

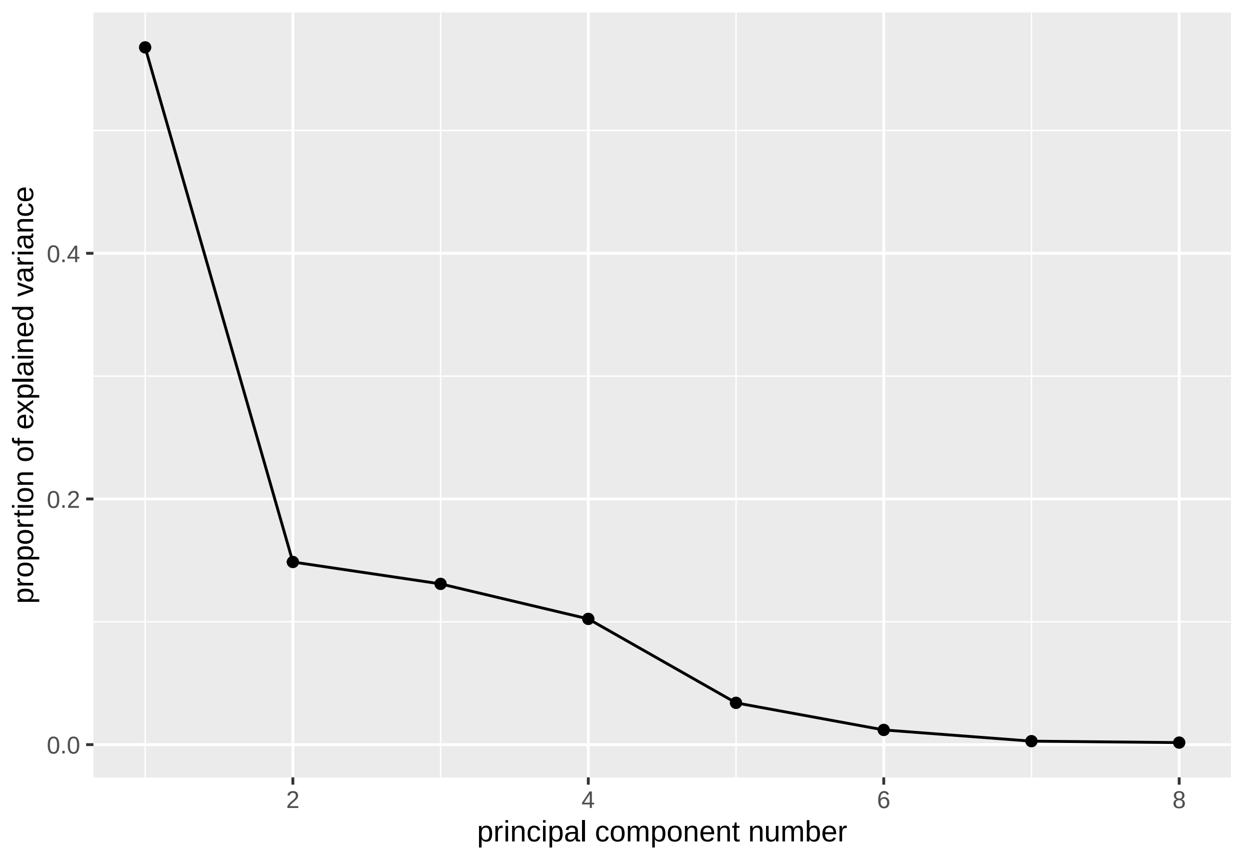
```
pca=prcomp(bearing_pca,center = T,scale. = T)
summary(pca)
```

```
## Importance of components:
##                            PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.1310 1.0906 1.0232 0.9049 0.52163 0.30949 0.15048
## Proportion of Variance 0.5676 0.1487 0.1309 0.1023 0.03401 0.01197 0.00283
## Cumulative Proportion  0.5676 0.7163 0.8472 0.9495 0.98354 0.99551 0.99834
##                            PC8
## Standard deviation     0.11512
## Proportion of Variance 0.00166
## Cumulative Proportion  1.00000
```

```r
print(pca)
```

```
## Standard deviations (1, .., p=8):
## [1] 2.1309559 1.0906399 1.0231929 0.9048890 0.5216349 0.3094911 0.1504806
## [8] 0.1151163
##
## Rotation (n x k) = (8 x 8):
##                                PC1          PC2         PC3         PC4
## FERTILIZER (NITROGEN)   0.37836880  0.383948253 -0.07055534  0.40262617
## FERTILIZER (PHOSPHATE)  0.24438396  0.186219530 -0.80521652 -0.02031252
## FERTILIZER (POTASH)     0.46085434  0.024250696 -0.04361279 -0.11009307
## FERTILIZER (SULFUR)     0.27461067 -0.521406442 -0.20111755 -0.56637995
## FUNGICIDE (TOTAL)       0.44851096 -0.007854105  0.01784719  0.15308273
## HERBICIDE (TOTAL)       0.38162932  0.030128476  0.39151573  0.14573906
## INSECTICIDE (TOTAL)     0.40304914 -0.179836407  0.34735022 -0.14907534
## OTHER (TOTAL)          -0.00800999 -0.715641465 -0.17332829  0.66157024
##                                PC5         PC6         PC7         PC8
## FERTILIZER (NITROGEN)  -0.03460227 -0.57655941  0.42141251  0.17643221
## FERTILIZER (PHOSPHATE) -0.10128969  0.02808025 -0.49361547  0.04591514
## FERTILIZER (POTASH)     0.20510582  0.13142642  0.20207714 -0.82025981
## FERTILIZER (SULFUR)    -0.28462344 -0.19478604  0.35761443  0.21130452
## FUNGICIDE (TOTAL)       0.27271447  0.66958652  0.20832213  0.45706234
## HERBICIDE (TOTAL)      -0.75688092  0.14106271 -0.28682238 -0.06238920
## INSECTICIDE (TOTAL)     0.46682467 -0.37326877 -0.53199387  0.14853381
## OTHER (TOTAL)           0.01723375 -0.06326030 -0.03673353 -0.12011261
```

```r
ggbiplot::ggscreeplot(pca)
```



As we can see, the first 4 principal component is important.

4 EDA for 'market'

The EDA for 'market' table is much easier. The relationships between different variables are extremly clear. We can use an example of California in 2019 to illustrate these:

```r
market=read.csv('Berries-data/b_berry_market.csv')
market_value=filter(market,
                    Year=='2019',
                    State=='CALIFORNIA')
market_value=select(market_value,-Commodity,-TAME)
market_value=arrange(market_value,market_value[,3])
(market_value=data.frame(No.=1:nrow(market_value),market_value))
```

```
##    No. Year      State                       Measurement      Unit        Value
## 1    1 2019 CALIFORNIA                   ACRES HARVESTED      ACRE        7,300
## 2    2 2019 CALIFORNIA FRESH MARKET - PRICE RECEIVED    $ / LB         3.56
## 3    3 2019 CALIFORNIA     FRESH MARKET - PRODUCTION         $ 199,930,000
## 4    4 2019 CALIFORNIA     FRESH MARKET - PRODUCTION        LB  56,160,000
## 5    5 2019 CALIFORNIA         NOT SOLD - PRODUCTION        LB   1,920,000
## 6    6 2019 CALIFORNIA                PRICE RECEIVED    $ / LB         2.85
## 7    7 2019 CALIFORNIA   PROCESSING - PRICE RECEIVED    $ / LB         0.29
## 8    8 2019 CALIFORNIA       PROCESSING - PRODUCTION         $   4,530,000
## 9    9 2019 CALIFORNIA       PROCESSING - PRODUCTION        LB  15,620,000
## 10  10 2019 CALIFORNIA                    PRODUCTION        LB  73,700,000
## 11  11 2019 CALIFORNIA         UTILIZED - PRODUCTION         $ 204,460,000
## 12  12 2019 CALIFORNIA         UTILIZED - PRODUCTION        LB  71,780,000
## 13  13 2019 CALIFORNIA                        YIELD LB / ACRE      10,100
```

So, here we conclude 13 equations to explain there relationships:

- Row1 = Row10 / Row13
- Row2 = Row3 / Row4
- Row3 = Row2 * Row4
- Row4 = Row3 / Row2
- Row5 = Row10 - Row12
- Row6 = Row11 / Row12
- Row7 = Row8 / Row9
- Row8 = Row7 * Row9
- Row9 = Row8 / Row7
- Row10 = Row5 + Row12
- Row11 = Row6 * Row12
- Row12 = Row4 + Row9
- Row13 = Row10 / Row1

5 Create Shiny

Then, we create a shiny app to display the data we get. The app Berries-shiny has been deployed on shinyapps.io. And the code is listed in app.R.

6 Create Slide

Finally we create a slide Berries-slide to summary our whole EDA process. The code of slide is in Berries-slide.Rmd.

7 Reference

[1] National Agricultural Statistics Service

[2] Visit Maine

[3] Vince Vu