Final Project STAT 600

Ashish Bhandari

08/04/2022

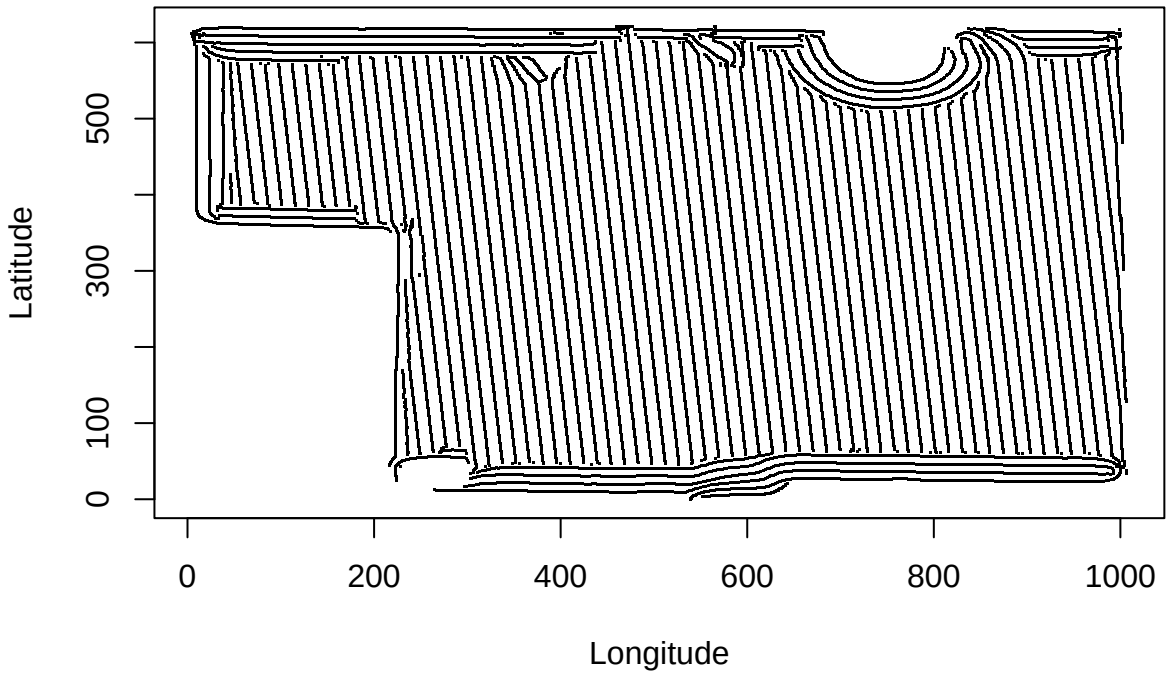
Introduction

We have yield data from a corn field for the years 2018, 2019 and 2020. When the corn seed was planted, it was planted at different seed densities in different parts of the field. The seeding density was targeted to be higher in the parts of the field that were expected to be more fertile. Fertility is based on the past years yield over different parts of the field. There are variations in the fields due to differences in the crop sowed in the previous years, their densities, crop type etc. Now, If a specific region of the corn field was low yielding in 2018, we need to understand the reason for the same.

Reading the data

The data provided to us contains six files, each related to some specific year and the yield or applied rate of a a specific crop in that year. Each row in each file contains a GPS position specified by the Longitude and the Latitude columns. Apart from the yield and the Applied rate, they also contain information on Moisture content, Distance etc.

```
library(readr)
library(readxl)

data1<- read.csv("~/Portfolio Data/A 2017 Soybeans Harvest.csv")
data2 <- read.csv("~/Portfolio Data/A 2018 Corn Harvest.csv")
data3 <- read.csv("~/Portfolio Data/A 2018 Corn Seeding.csv")
data4 <- read.csv("~/Portfolio Data/A 2019 Soybeans Harvest.csv")
data5 <- read.csv("~/Portfolio Data/A 2020 Corn Harvest.csv")
data6 <- read.csv("~/Portfolio Data/A 2020 Corn Seeding.csv")

plot(Latitude ~ Longitude,data = data1, pch = ".")
```



Algorithm

We followed a straight forward algorithm in this project. We defined a cell variable for each of the data frames, and then aggregated the data by grouping the yield or Applied Rates by Cell number in each data frame. Next, the pair plot was obtained between the merged data frames. The next step was to apply normalization techniques on the variables of interest. Then repeat aggregation and other procedures as done previously, to see how normalization affected our results.

Finding rows, columns and cells

```
row1 <- ceiling((data1$Latitude)/50)
col1 <- ceiling((data1$Longitude)/50)
data1$cell <- row1 * 1000 + col1

row2 <- ceiling((data2$Latitude)/50)
col2 <- ceiling((data2$Longitude)/50)
data2$cell <- row2 * 1000 + col2

row3 <- ceiling((data3$Latitude)/50)
col3 <- ceiling((data3$Longitude)/50)
data3$cell <- row3 * 1000 + col3
```

```
row4 <- ceiling((data4$Latitude)/50)
col4 <- ceiling((data4$Longitude)/50)
data4$cell <- row4 * 1000 + col4

row5 <- ceiling((data5$Latitude)/50)
col5 <- ceiling((data5$Longitude)/50)
data5$cell <- row5 * 1000 + col5

row6 <- ceiling((data6$Latitude)/50)
col6 <- ceiling((data6$Longitude)/50)
data6$cell <- row6 * 1000 + col6
```

Aggregating the data

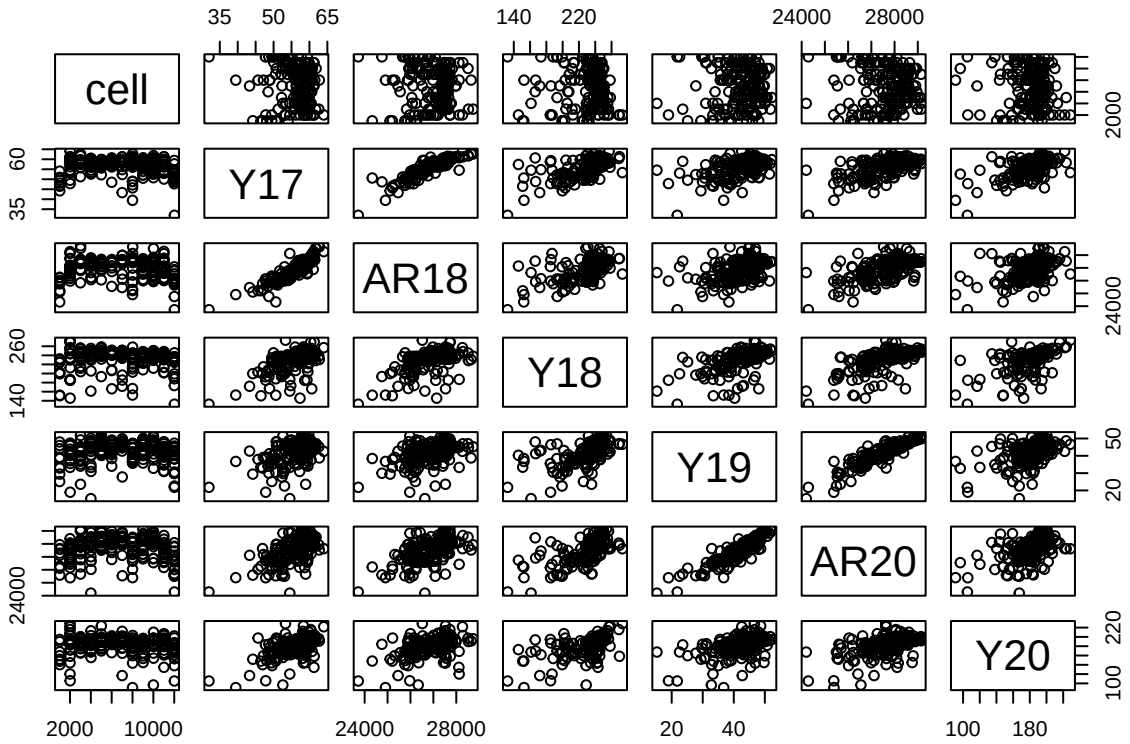
```
library(dplyr)
agg_data1 <- data1 %>% group_by(cell)  %>%
summarise(Y17 = mean(Yield), Count = length(Yield), .groups = 'drop')
agg_data1 <- agg_data1[agg_data1$Count >= 30,c(1,2)]

agg_data2 <- data3 %>% group_by(cell)  %>%
summarise(AR18 = mean(AppliedRate),Count = length(AppliedRate), .groups ='drop')

agg_data2 <- agg_data2[agg_data2$Count >= 30,c(1,2)]

agg_data3 <- data2 %>% group_by(cell)  %>%
summarise(Y18 = mean(Yield), Count = length(Yield), .groups = 'drop')

agg_data3 <- agg_data3[agg_data3$Count >= 30,c(1,2)]

agg_data4 = data4 %>% group_by(cell)  %>%
summarise(Y19 = mean(Yield), Count = length(Yield), .groups = 'drop')

agg_data4 <- agg_data4[agg_data4$Count >= 30,c(1,2)]

agg_data5 <- data6 %>% group_by(cell)  %>%
summarise(AR20 = mean(AppliedRate), Count = length(AppliedRate),
            .groups = 'drop')

agg_data5 <- agg_data5[agg_data5$Count >= 30,c(1,2)]

agg_data6 <- data5 %>% group_by(cell)  %>%
summarise(Y20 = mean(Yield), Count = length(Yield), .groups = 'drop')

agg_data6 <- agg_data6[agg_data6$Count >= 30,c(1,2)]
```

Merging the data

```
temp1 <- merge(agg_data1,agg_data2, by="cell")
temp2 <- merge(temp1,agg_data3,by="cell")
temp3 <- merge(temp2,agg_data4, by="cell")
temp4 <- merge(temp3,agg_data5, by="cell")
Combined.dat <- merge(temp4,agg_data6, by="cell")
```

Obtaining pairplot

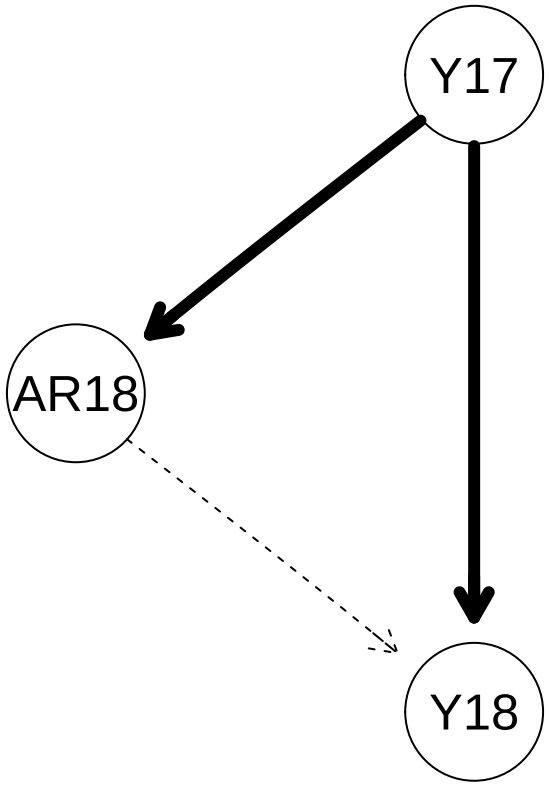
```
pairs(Combined.dat)
```



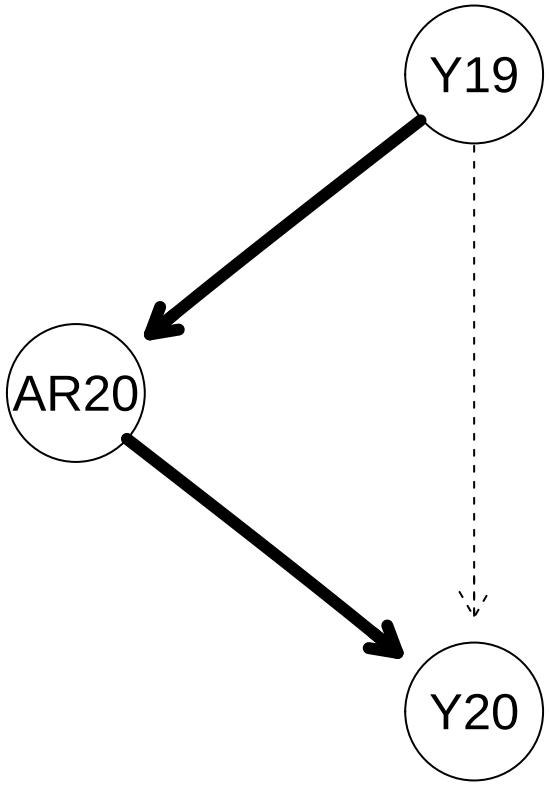
Obtaining strength plots

```
#install.packages("BiocManager")
#BiocManager::install("Rgraphviz")

library(bnlearn)
```

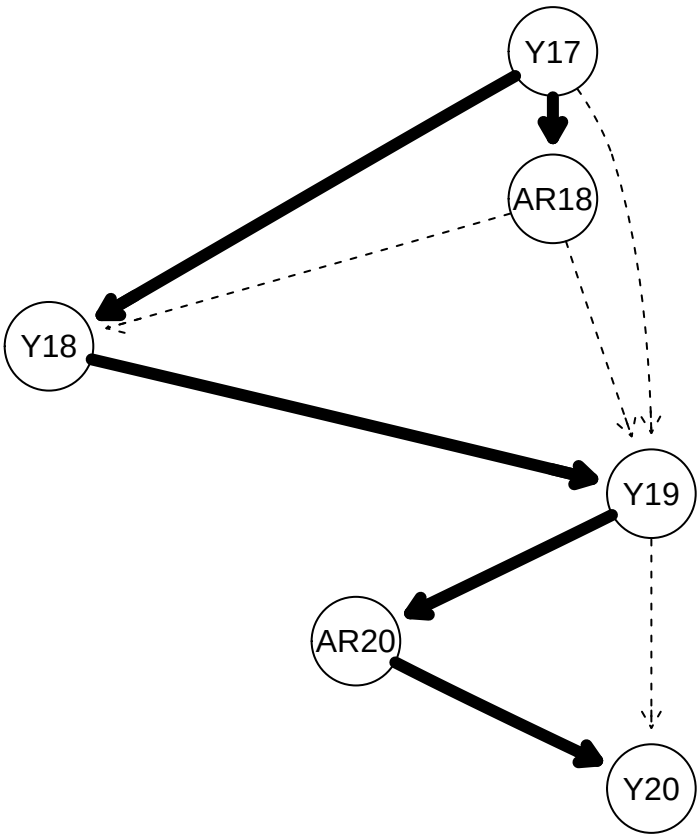
```
modela.dag <- model2network("[Y17][AR18|Y17][Y18|AR18:Y17]")
fit1 = bn.fit(modela.dag, Combined.dat[,c('Y17','AR18','Y18')])
#fit1
strengtha <- arc.strength(modela.dag, Combined.dat[,c('Y17','AR18','Y18')])
strength.plot(modela.dag, strengtha)
```




```
modelb.dag <- model2network("[Y19][AR20|Y19][Y20|AR20:Y19]")
fit2 = bn.fit(modelb.dag, Combined.dat[,c('Y19','AR20','Y20')])
#fit2
strengthb <- arc.strength(modelb.dag, Combined.dat[,c('Y19','AR20','Y20')])
strength.plot(modelb.dag, strengthb)
```




```
model1.dag <- model2network("[Y17][AR18|Y17][Y18|AR18:Y17][Y19|Y17:AR18:Y18][AR20|Y19][Y20|AR20:Y19]")
fit3 = bn.fit(model1.dag,
              Combined.dat[,c('Y17','AR18','Y18','Y19','AR20','Y20')])
#fit3
strength1 <- arc.strength(model1.dag,
                      Combined.dat[,c('Y17','AR18','Y18','Y19','AR20','Y20')])
strength.plot(model1.dag, strength1)
```



Performing normalization using Option 1

We have chosen rank normalization as our normalization technique. Rank normalization is an alternative to quantile normalization. It replaces each observation by its fractional rank (the rank divided by the total number of genes) within array [22, 23]. This normalization procedure achieves robustness to non-additive noise at the expense of losing some parametric information of expressions.

```
data1$Rank <- rank(data1$Yield)
data2$Rank <- rank(data2$Yield)
data3$Rank_AppliedRate <- rank(data3$AppliedRate)

data4$Rank <- rank(data4$Yield)
data5$Rank <- rank(data5$Yield)
data6$Rank_AppliedRate <- rank(data6$AppliedRate)
```

Aggregating the data

```
agg_data1_n <- data1 %>% group_by(cell)  %>%
summarise(Y17 = mean(Rank),Count = length(Rank), .groups = 'drop')

agg_data1_n <- agg_data1_n[agg_data1_n$Count >= 30,c(1,2)]
```

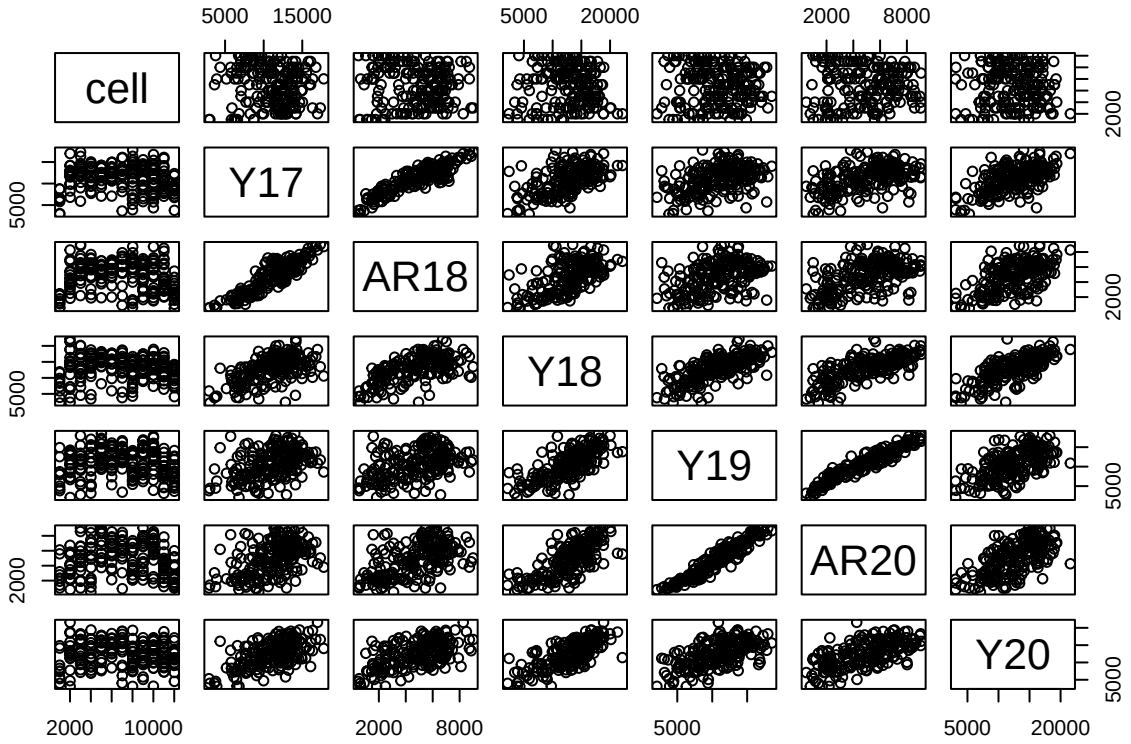
```r
agg_data2_n <- data3 %>% group_by(cell)  %>%
summarise(AR18 = mean(Rank_AppliedRate),Count = length(Rank_AppliedRate),
          .groups = 'drop')

agg_data2_n <- agg_data2_n[agg_data2_n$Count >= 30,c(1,2)]

agg_data3_n <- data2 %>% group_by(cell)  %>%
summarise(Y18 = mean(Rank), Count = length(Rank), .groups = 'drop')

agg_data3_n <- agg_data3_n[agg_data3_n$Count >= 30,c(1,2)]

agg_data4_n <- data4 %>% group_by(cell)  %>%
summarise(Y19 = mean(Rank), Count = length(Rank), .groups = 'drop')

agg_data4_n <- agg_data4_n[agg_data4_n$Count >= 30,c(1,2)]

agg_data5_n <- data6 %>% group_by(cell)  %>%
summarise(AR20 = mean(Rank_AppliedRate), Count = length(Rank_AppliedRate),
          .groups = 'drop')

agg_data5_n <- agg_data5_n[agg_data5_n$Count >= 30,c(1,2)]

agg_data6_n <- data5 %>% group_by(cell)  %>%
summarise(Y20 = mean(Rank),Count = length(Rank), .groups = 'drop')

agg_data6_n <- agg_data6_n[agg_data6_n$Count >= 30,c(1,2)]
```

Merging the data

```r
temp1 = merge(agg_data1_n,agg_data2_n, by="cell")
temp2 = merge(temp1,agg_data3_n,by="cell")
temp3 = merge(temp2,agg_data4_n, by="cell")
temp4 = merge(temp3,agg_data5_n, by="cell")
Combined.dat.n = merge(temp4,agg_data6_n, by="cell")
```

Obtaining pair plot

```r
pairs(Combined.dat.n)
```



Conclusion

The study helped us realize the important variables which affect the Yield and Applied rate in 2020 crop harvest and seeding.A prominent variable having strong positive correlation with AR20 is Y19, i.e., yield of Soybean in 2019. Likewise, we have other variables too. Also, normalizing the data helped reproduce better outcomes, indicating stronger association between the variables.