SOUTH DAKOTA
STATE UNIVERSITY

STATISTICAL PROGRAMMING: FINAL PROJECT

GROUP MEMBERS: PRINCE AGYAPONG, ISAAC GBENE & MICHAEL KOJO ABALO

AUGUST 4, 2023

Contents

LIST OF FIGURES

INTRODUCTION

Causal Inference is the process where causes are inferred from data. Considering data obtained from seeding rate and yield for the years 2017 to 2020 we want to understand the relationship between seeding rate (the number of seeds that you plant per unit of ground) and how much yield you will get from the field. Inferring from the geospatial images for various years it was conspicuous that some parts of the field had high seeding rates but did not necessarily have high yields and some parts of the field had medium seeding rate but had high yields. In 2017 for instance, soybeans were planted in the field and in 2018 corn was planted on the same field. There might be some kind of rotation effect that if you plant corn following soybeans you might get a different yield than if you planted corn and something else. Investigating how much the yield in 2017 affect the yield in 2018 would be something very interesting to look at in this case. Also, we might want to investigate the seeding rate for these two years as well. Considering the fact that seeding rate and yield are not of the same units so we can’t count them the same, we have to normalize the data in order to make them comparable in terms of units.

DATA EXPLORATION

In this section we explore our data obtained through plots, description of the variables, data cleaning, checking for outliers using boxplots and computing skewness and kurtosis considering our variables of interest.

```
A.2017.Soybeans.Harvest <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT
600/Final Project/A 2017 Soybeans Harvest.csv", header=TRUE)

A.2018.Corn.Seeding <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT 600/Final
Project/A 2018 Corn Seeding.csv", header=TRUE)

A.2018.Corn.Harvest <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT 600/Final
Project/A 2018 Corn Harvest.csv", header=TRUE)
```

```r
A.2019.Soybeans.Harvest <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT
600/Final Project/A 2019 Soybeans Harvest.csv", header=TRUE)

A.2020.Corn.Harvest <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT 600/Final
Project/A 2020 Corn Harvest.csv", header=TRUE)

A.2020.Corn.Seeding <- read.csv("C:/Users/GINIT/Desktop/SUMMER/STAT 600/Final
Project/A 2020 Corn Seeding.csv", header=TRUE)
```

Check if the data Read in as expected

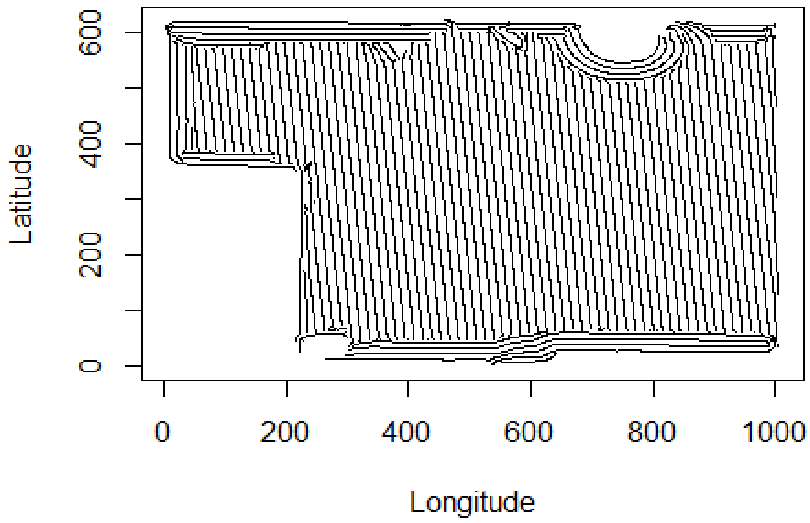

Figure 1: 2017 Soybeans Harvest

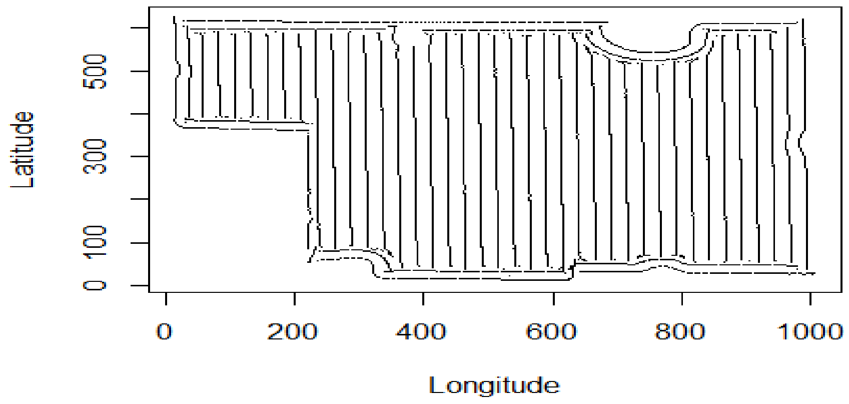

Figure 2: 2018 Corn Seeding

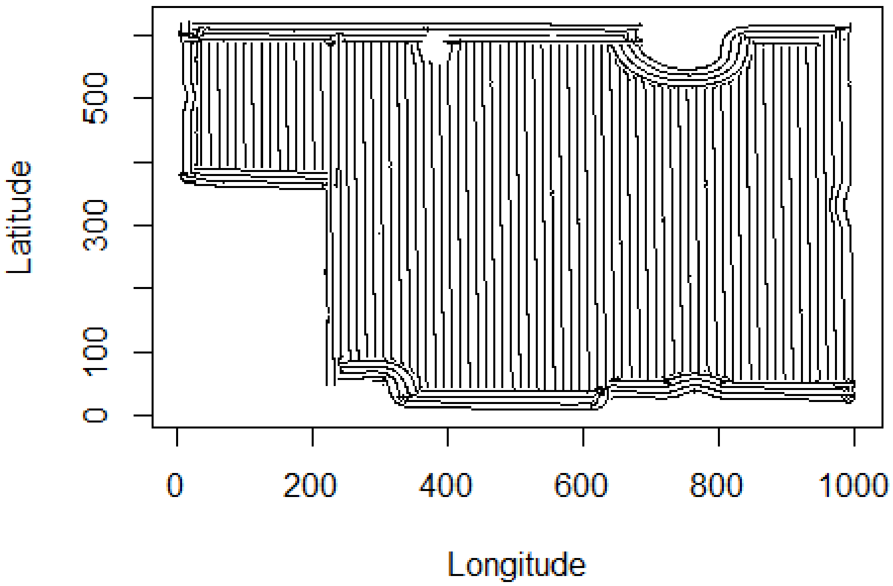

Figure 3: 2018 Corn Harvest

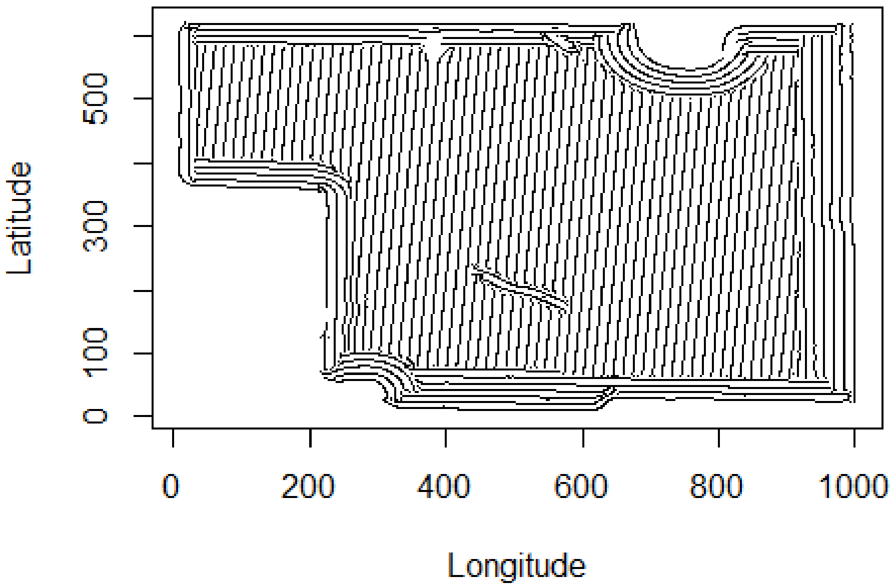

Figure 4: 2019 Soybeans Harvest

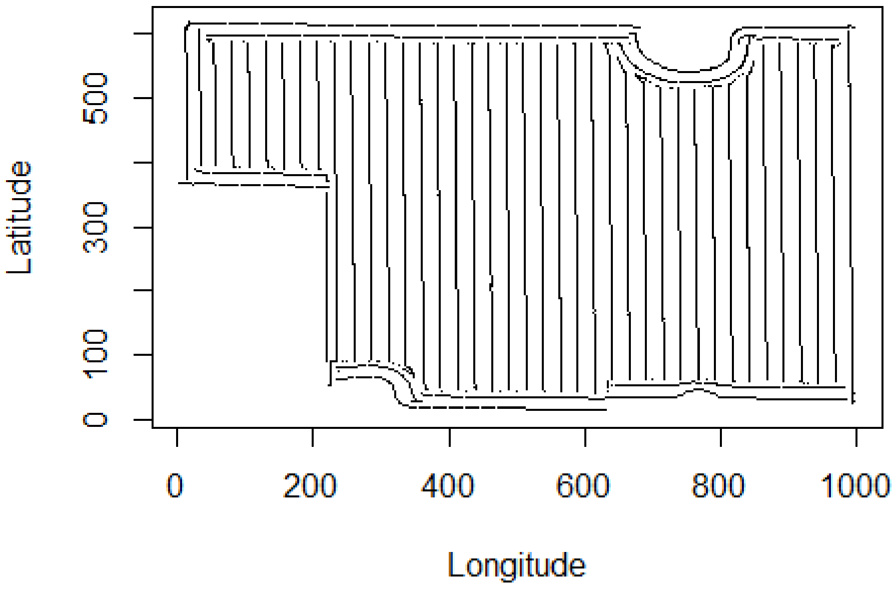

Figure 5: 2020 Corn Seeding

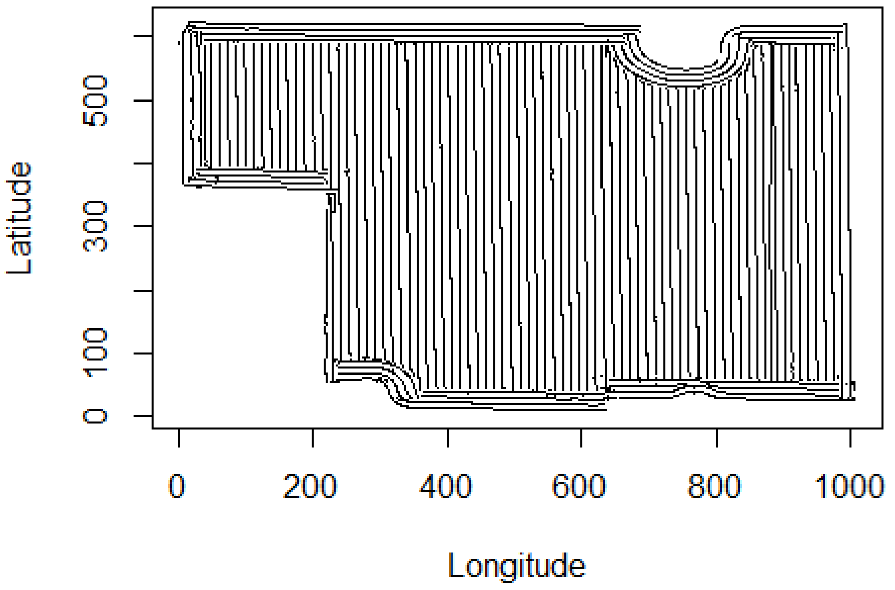

Figure 6: 2020 Corn Harvest

Data and description of variables

Corn Seeding dataset has 9 variables with the variables of interest being Longitude, Latitude and Applied Rate. The Harvest (Corn and Soybeans) datasets have 11 variables each, the variables of interest are Longitude, Latitude and Yield.

Latitude: imaginary lines that runs from the north to the south of the fields.

Longitude: imaginary lines that run east to west of fields.

Yield: the full amount of an agricultural or industrial product. Yield is measured in bushels per acre.

AppliedRate: The rate at which the seeds are planted. It is measured in seeds per acre.

Using the nrow() function in R, we found that 2017 Soybeans Harvest has 21,612 observations, 2018 Corn Seeding has 9221 observations, 2018 Corn Harvest has 25,146 observations, 2019 Soybeans Harvest has 20835 observations, 2020 Corn Seeding has 9498 observations, and 2020 Corn Harvest has 24623 observations.

DATA CLEANING

Exploring data types:

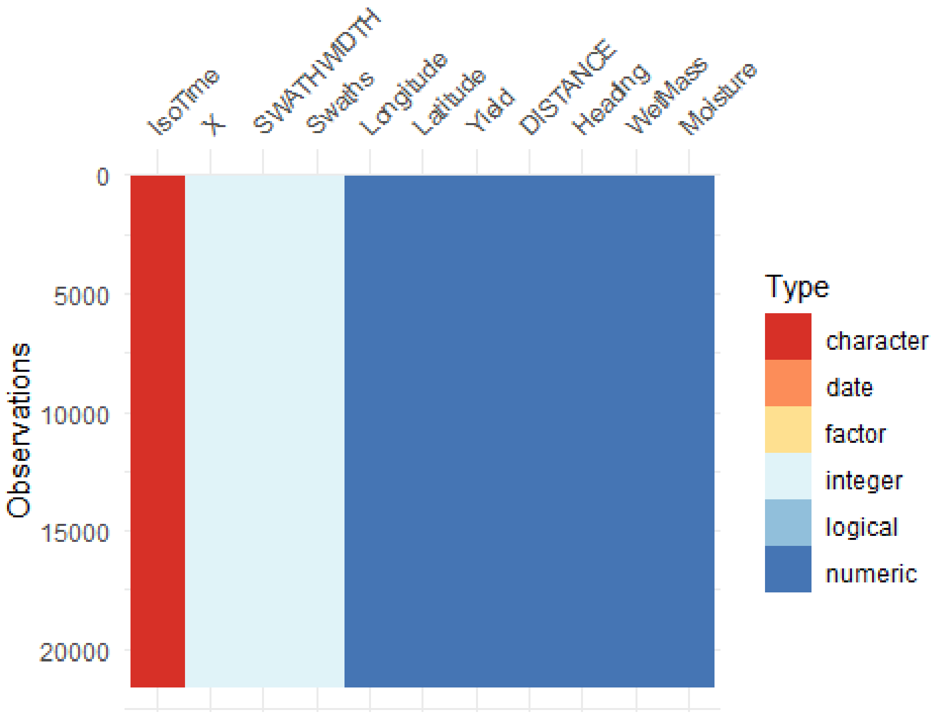

Figure 7: Data types of the variables in the harvest datasets

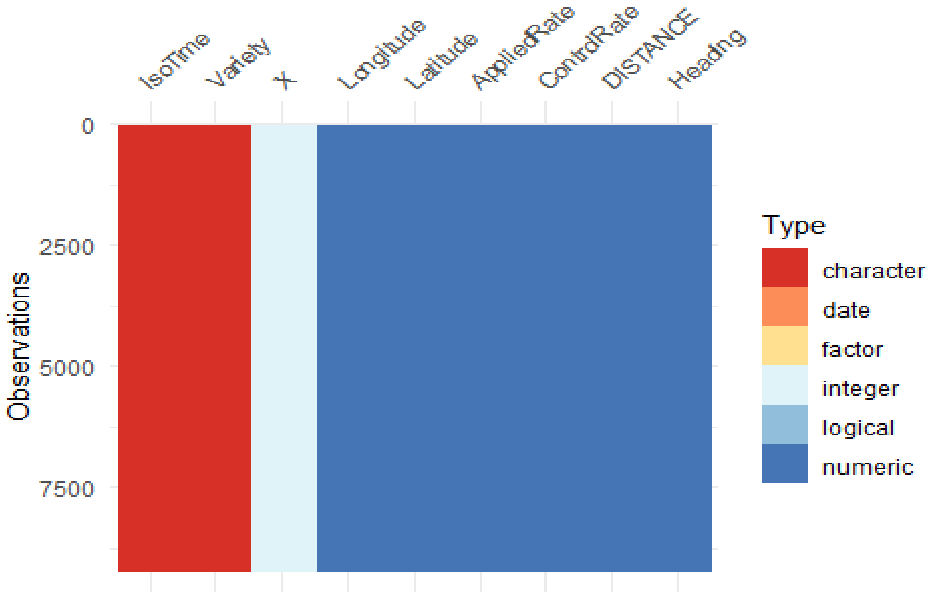

Figure 8: Data types of the variables in the seeding datasets

Checking for missing data

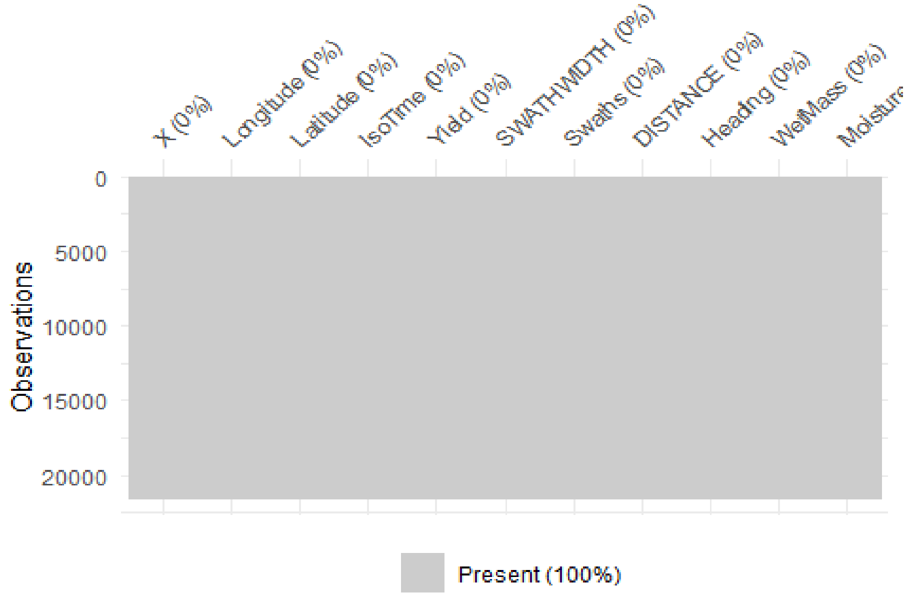

Figure 9: No Missing data in 2017 soybeans Harvest dataset

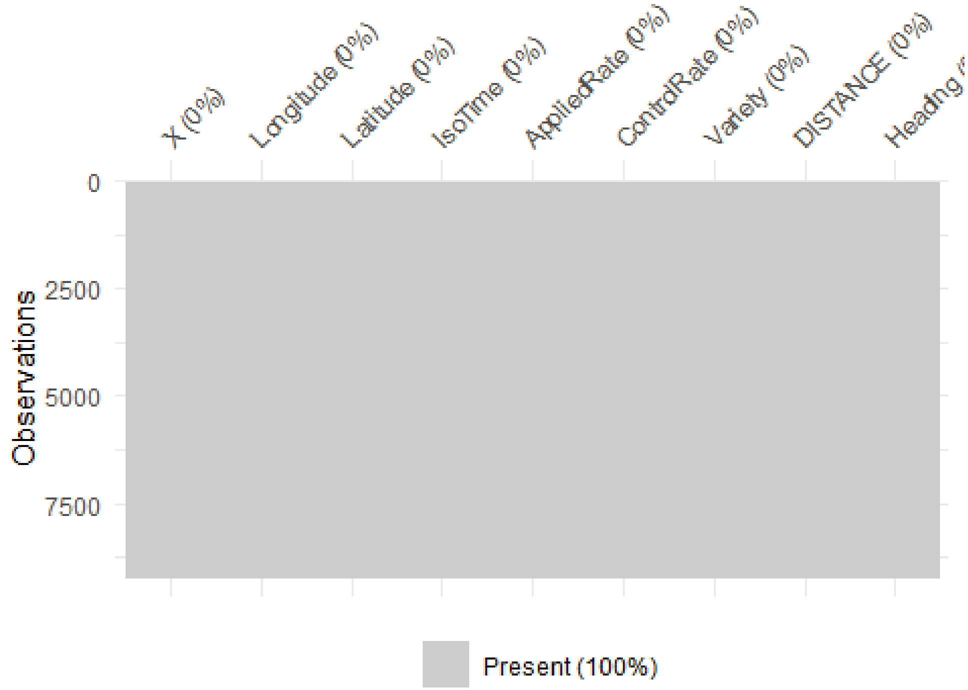

Figure 10: No Missing data in 2018 Corn seeding dataset

Similarly, there was no missing data in all the other four datasets as seen in the Appendix.

Checking For Outliers

```
par(mfrow=c(2,2))
boxplot(A.2017.Soybeans.Harvest$Yield, col = "red", horizontal = T, xlab=
"2017 Soybeans Yield")
boxplot(A.2018.Corn.Harvest$Yield, col = "red", horizontal = T, xlab= "2018
Corn Yield")
boxplot(A.2018.Corn.Seeding$AppliedRate, col = "red", horizontal = T, xlab=
"2018 Corn AppliedRate")
boxplot(A.2019.Soybeans.Harvest$Yield, col = "red", horizontal = T, xlab=
"2019 Soybeans Yield")
```

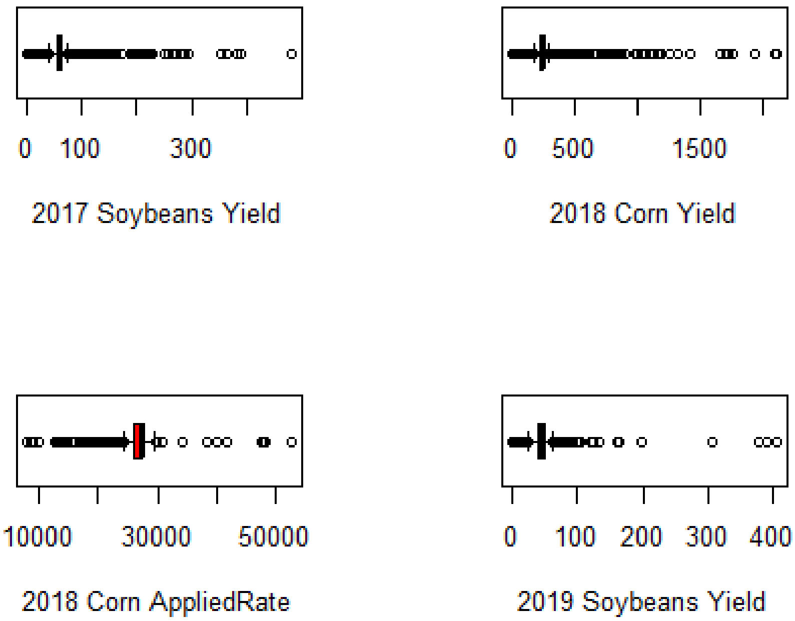

Figure 11: Boxplot 1

```
boxplot(A.2020.Corn.Harvest$Yield, col = "red", horizontal = T, xlab= "2020
Corn Yield")
boxplot(A.2020.Corn.Seeding$AppliedRate, col = "red", horizontal = T, xlab=
"2020 Corn AppliedRate")
```

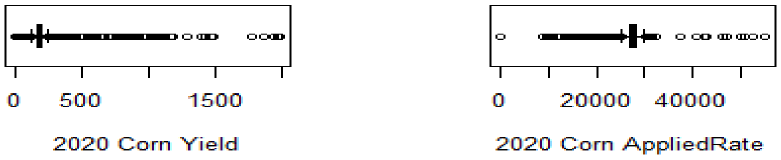

Figure 12: Boxplot 2

Based on the boxplot above for various years, it is apparent we have outliers associated with our variables of interest. How the outliers are going to be handled will be massively based on the normalization technique employed.

Computing Skewness and Kurtosis for Our Variables Of Interest

```
library(moments)
skewness(A.2017.Soybeans.Harvest$Yield)

## [1] 4.36817

kurtosis(A.2017.Soybeans.Harvest$Yield)

## [1] 113.3981

skewness(A.2018.Corn.Harvest$Yield)

## [1] 9.409054

kurtosis(A.2018.Corn.Harvest$Yield)

## [1] 221.4894

skewness(A.2018.Corn.Seeding$AppliedRate)

## [1] 0.6616141

kurtosis(A.2018.Corn.Seeding$AppliedRate)

## [1] 72.04693
```

```
skewness(A.2019.Soybeans.Harvest$Yield)

## [1] 2.461306

kurtosis(A.2019.Soybeans.Harvest$Yield)

## [1] 100.5045

skewness(A.2020.Corn.Harvest$Yield)

## [1] 9.968059

kurtosis(A.2020.Corn.Harvest$Yield)

## [1] 230.2404

skewness(A.2020.Corn.Seeding$AppliedRate)

## [1] -1.528581

kurtosis(A.2020.Corn.Seeding$AppliedRate)

## [1] 63.01505
```

Skewness and kurtosis were calculated to understand the distribution and shape of the data, precisely our variables of interest. For Soybeans Harvest Yield in 2017, the data displayed positive skewness (4.36817) with a leptokurtic distribution (kurtosis of 113.3981). Similarly, Corn Harvest Yield in 2018 exhibited highly positive skewness (9.409054) and a highly leptokurtic distribution (kurtosis of 221.4894). The Applied Rate for Corn Seeding in 2018 showed slight positive skewness (0.6616141) and a distribution with slight leptokurtic characteristics (kurtosis of 72.04693). In 2019, Soybeans Harvest Yield remained positively skewed (2.461306) with a slightly leptokurtic distribution (kurtosis of 100.5045). The Corn Harvest Yield in 2020 displayed highly positive skewness (9.968059) and a highly leptokurtic distribution (kurtosis of 230.2404). Lastly, the Applied Rate for Corn Seeding in 2020 exhibited negative skewness (-1.528581) and a distribution with slight leptokurtic characteristics (kurtosis of 63.01505). These statistics provide valuable insights into the data's deviation from a normal distribution.

ALGORITHM AND NORMALIZATION

The normalization method we opted for was the rank normalization method and we implemented this method to produce the next strength plots. Normalization may reduce the impact of the outliers on the analysis, as we've seen from our prelimenary analysis that our variables of interest were affected by outliers. The original data was still used since the rank method cleans the data compared to the other options and removes the noise by only preserving the ordering of observations. The normalization method was implemented before aggregating the data as instructed. We then merged the aggregated data on cells to obtain the combined.dat. Further explanations regarding the normalization method will be elaborated as we proceed.

Create The Grid

To create the grid, we need to create row and column variables of the Latitude and Longitude respectively by using the following formulae.

The grid has 50m by 50m dimensions.

$$Row = ceiling\left(\frac{Latitude}{50}\right)$$

$$Column = ceiling\left(\frac{Longitude}{50}\right)$$

$$Cell = 1000 * Row + Column$$

```
#Compute row and column variables
A.2017.Soybeans.Harvest$Row <- ceiling(A.2017.Soybeans.Harvest$Latitude/50)
A.2017.Soybeans.Harvest$Column <-
ceiling(A.2017.Soybeans.Harvest$Longitude/50)

A.2018.Corn.Harvest$Row <- ceiling(A.2018.Corn.Harvest$Latitude/50)
A.2018.Corn.Harvest$Column <- ceiling(A.2018.Corn.Harvest$Longitude/50)

A.2019.Soybeans.Harvest$Row <- ceiling(A.2019.Soybeans.Harvest$Latitude/50)
```

```
A.2019.Soybeans.Harvest$Column <-
ceiling(A.2019.Soybeans.Harvest$Longitude/50)

A.2020.Corn.Harvest$Row <-  ceiling(A.2020.Corn.Harvest$Latitude/50)
A.2020.Corn.Harvest$Column <- ceiling(A.2020.Corn.Harvest$Longitude/50)

A.2018.Corn.Seeding$Row <- ceiling(A.2018.Corn.Seeding$Latitude/50)
A.2018.Corn.Seeding$Column <- ceiling(A.2018.Corn.Seeding$Longitude/50)

A.2020.Corn.Seeding$Row <- ceiling(A.2020.Corn.Seeding$Latitude/50)
A.2020.Corn.Seeding$Column <- ceiling(A.2020.Corn.Seeding$Longitude/50)

#Create the cells
A.2018.Corn.Seeding$Cells <-
1000*A.2018.Corn.Seeding$Row+A.2018.Corn.Seeding$Column
A.2017.Soybeans.Harvest$Cells <-
1000*A.2017.Soybeans.Harvest$Row+A.2017.Soybeans.Harvest$Column
A.2018.Corn.Harvest$Cells <-
1000*A.2018.Corn.Harvest$Row+A.2018.Corn.Harvest$Column
A.2019.Soybeans.Harvest$Cells <-
1000*A.2019.Soybeans.Harvest$Row+A.2019.Soybeans.Harvest$Column
A.2020.Corn.Seeding$Cells <-
1000*A.2020.Corn.Seeding$Row+A.2020.Corn.Seeding$Column
A.2020.Corn.Harvest$Cells <-
1000*A.2020.Corn.Harvest$Row+A.2020.Corn.Harvest$Column

plot(Latitude ~ Longitude,data=A.2020.Corn.Harvest,pch = ".")
abline(h=1:12*50,v=1:20*50,col='red')
```

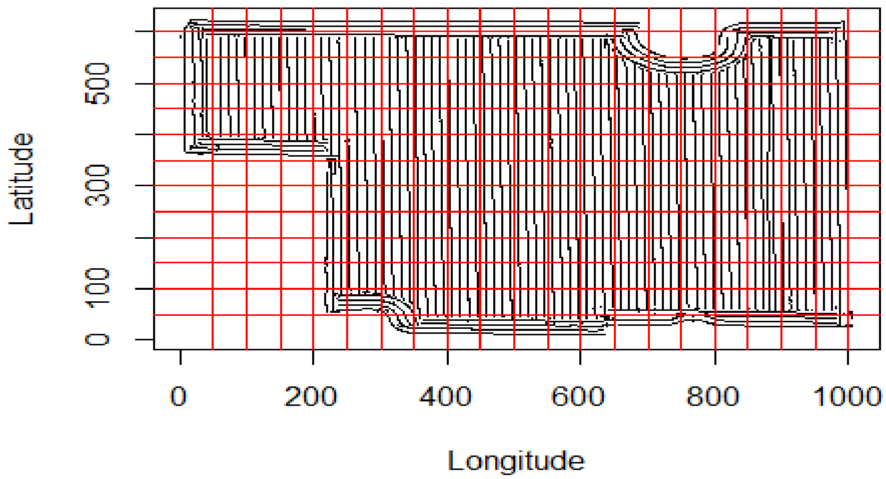

Figure 13: Harvest per grid for 2020 Corn Harvest Dataset

```
plot(Row ~ Column,data=A.2020.Corn.Harvest)
abline(h=1:12+0.5,v=1:20+0.5,col='red')
```

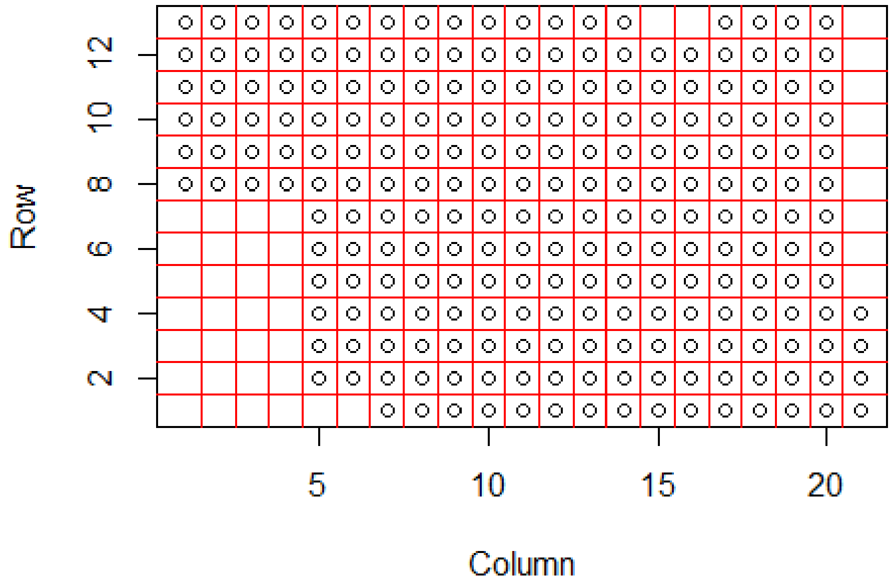

Figure 14: Grid with data points

Aggregate Data

In each grid cell, we compute the average and the length (number of observation) the yield and the AppliedRate for all the datasets. We select grid cells with at least 30 observations.

```
#Applied Rate
AR18 <- aggregate(A.2018.Corn.Seeding$AppliedRate,
by=list(A.2018.Corn.Seeding$Cells), FUN=mean)
Samp.AR18 <- aggregate(A.2018.Corn.Seeding$AppliedRate,
by=list(A.2018.Corn.Seeding$Cells), FUN=length)

AR20 <-  aggregate(A.2020.Corn.Seeding$AppliedRate,
by=list(A.2020.Corn.Seeding$Cells), FUN=mean)
Samp.AR20 <- aggregate(A.2020.Corn.Seeding$AppliedRate,
by=list(A.2020.Corn.Seeding$Cells), FUN=length)

#Yield
```

```r
Y17 <- aggregate(A.2017.Soybeans.Harvest$Yield,
by=list(A.2017.Soybeans.Harvest$Cells), FUN=mean)
Samp.Y17 <- aggregate(A.2017.Soybeans.Harvest$Yield,
by=list(A.2017.Soybeans.Harvest$Cells), FUN=length)

Y18 <- aggregate(A.2018.Corn.Harvest$Yield,
by=list(A.2018.Corn.Harvest$Cells), FUN=mean)
Samp.Y18 <- aggregate(A.2018.Corn.Harvest$Yield,
by=list(A.2018.Corn.Harvest$Cells), FUN=length)

Y19 <- aggregate(A.2019.Soybeans.Harvest$Yield,
by=list(A.2019.Soybeans.Harvest$Cells), FUN=mean)
Samp.Y19 <- aggregate(A.2019.Soybeans.Harvest$Yield,
by=list(A.2019.Soybeans.Harvest$Cells), FUN=length)

Y20 <- aggregate(A.2020.Corn.Harvest$Yield,
by=list(A.2020.Corn.Harvest$Cells), FUN=mean)
Samp.Y20 <- aggregate(A.2020.Corn.Harvest$Yield,
by=list(A.2020.Corn.Harvest$Cells), FUN=length)

#Subset the data to get those with atleast 30 observations
AR18 <- na.omit(subset(AR18,Samp.AR18>30)) #drop the nulls associated with
subseting the data
names(AR18)[1] <-'Cells'
names(AR18)[2] <- 'AR18'

AR20 <- na.omit(subset(AR20, Samp.AR20>30 ))
names(AR20)[1] <-'Cells'
names(AR20)[2] <- 'AR20'

Y17 <- na.omit(subset(Y17, Samp.Y17>30))
names(Y17)[1] <-'Cells'
names(Y17)[2] <- 'Y17'

Y18 <- na.omit(subset(Y18, Samp.Y18>30))
names(Y18)[1] <-'Cells'
names(Y18)[2] <- 'Y18'

Y19 <- na.omit(subset(Y19, Samp.Y19>30))
names(Y19)[1] <-'Cells'
names(Y19)[2] <- 'Y19'

Y20 <- na.omit(subset(Y20, Samp.Y20>30))
names(Y20)[1] <-'Cells'
names(Y20)[2] <- 'Y20'

merge1 <- merge(Y17, AR18, by="Cells")
merge2 <- merge(merge1,Y18, by="Cells")
merge3 <- merge(merge2, Y19, by="Cells")
```

```r
merge4 <- merge(merge3, AR20, by="Cells")
Combined.dat <- merge(merge4, Y20, by="Cells")
Combined.dat.ori <- Combined.dat

pairs(Combined.dat)
```

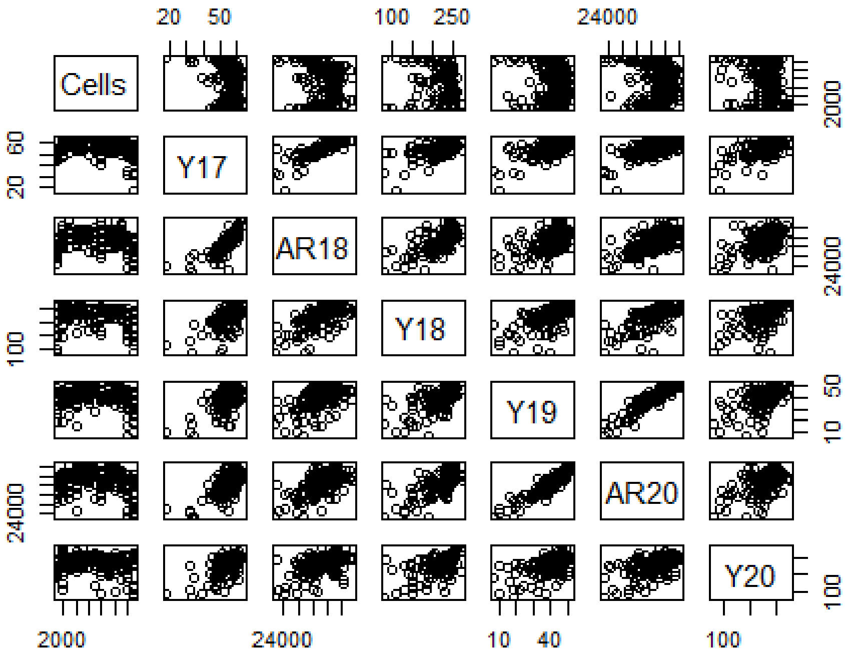

Figure 15: Pairs plot using original data

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


par(mfrow=c(1,3))
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')])

strength1 <- arc.strength(model1.dag,
Combined.dat[,c('Y17','AR18','Y18','Y19','AR20','Y20')])
strength.plot(model1.dag, strength1)
```

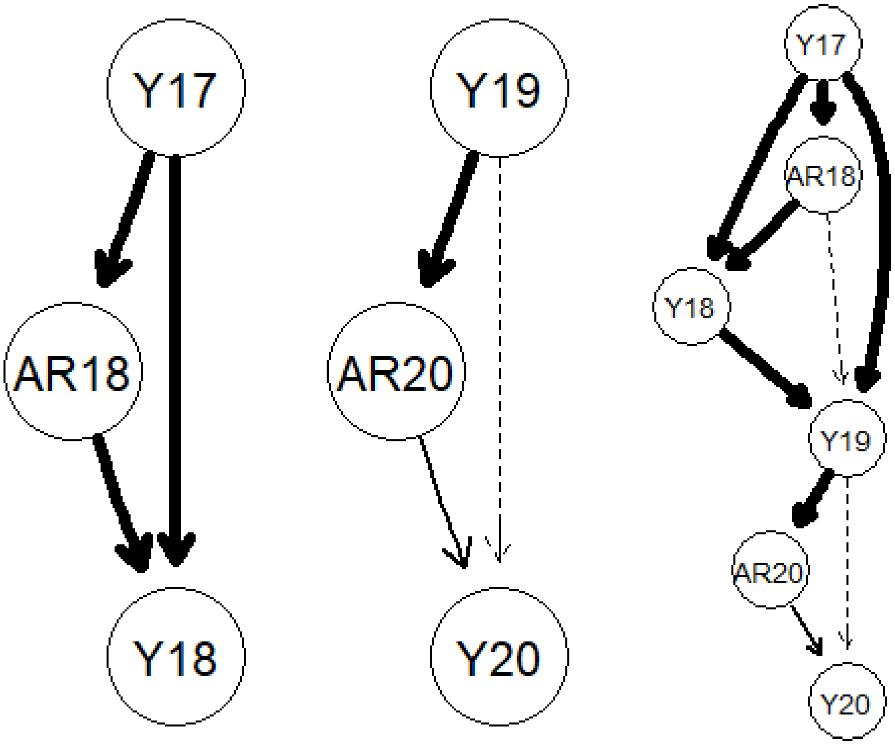

Figure 16: DAG plot using original data

Normalization method

Rank

Given several options to normalize data, we utilized the rank option. With rank normalization, we replace each entry in one array by its position (rank). Let j denote the year, y denote yield and i denote an observation. Then for j in {2017, 2018, 2019 and 2020} replace y_{ij} with $rank(y_{ij})$

```
#.RN denotes the ranked datasets
A.2017.Soybeans.Harvest.RN$RY17 <- rank(A.2017.Soybeans.Harvest.RN$Yield)
A.2018.Corn.Harvest.RN$RY18 <- rank(A.2018.Corn.Harvest.RN$Yield)
A.2019.Soybeans.Harvest.RN$RY19 <- rank(A.2019.Soybeans.Harvest.RN$Yield)
A.2020.Corn.Harvest.RN$RY20 <- rank(A.2020.Corn.Harvest.RN$Yield)

#Create row and column variables of the ranked datasets as done previously
A.2017.Soybeans.Harvest.RN$Row <-
ceiling(A.2017.Soybeans.Harvest.RN$Latitude/50)
A.2017.Soybeans.Harvest.RN$Column <-
ceiling(A.2017.Soybeans.Harvest.RN$Longitude/50)

A.2018.Corn.Harvest.RN$Row <- ceiling(A.2018.Corn.Harvest.RN$Latitude/50)
A.2018.Corn.Harvest.RN$Column <- ceiling(A.2018.Corn.Harvest.RN$Longitude/50)

A.2019.Soybeans.Harvest.RN$Row <-
ceiling(A.2019.Soybeans.Harvest.RN$Latitude/50)
A.2019.Soybeans.Harvest.RN$Column <-
ceiling(A.2019.Soybeans.Harvest.RN$Longitude/50)

A.2020.Corn.Harvest.RN$Row <-  ceiling(A.2020.Corn.Harvest.RN$Latitude/50)
A.2020.Corn.Harvest.RN$Column <- ceiling(A.2020.Corn.Harvest.RN$Longitude/50)

A.2018.Corn.Seeding.RN$Row <- ceiling(A.2018.Corn.Seeding.RN$Latitude/50)
A.2018.Corn.Seeding.RN$Column <- ceiling(A.2018.Corn.Seeding.RN$Longitude/50)

A.2020.Corn.Seeding.RN$Row <- ceiling(A.2020.Corn.Seeding.RN$Latitude/50)
A.2020.Corn.Seeding.RN$Column <- ceiling(A.2020.Corn.Seeding.RN$Longitude/50)

#Create the cells using the ranked datasets
A.2018.Corn.Seeding.RN$Cells <-
1000*A.2018.Corn.Seeding.RN$Row+A.2018.Corn.Seeding.RN$Column

A.2017.Soybeans.Harvest.RN$Cells <-
1000*A.2017.Soybeans.Harvest.RN$Row+A.2017.Soybeans.Harvest.RN$Column

A.2018.Corn.Harvest.RN$Cells <-
1000*A.2018.Corn.Harvest.RN$Row+A.2018.Corn.Harvest.RN$Column

A.2019.Soybeans.Harvest.RN$Cells <-
1000*A.2019.Soybeans.Harvest.RN$Row+A.2019.Soybeans.Harvest.RN$Column

A.2020.Corn.Seeding.RN$Cells <-
1000*A.2020.Corn.Seeding.RN$Row+A.2020.Corn.Seeding.RN$Column

A.2020.Corn.Harvest.RN$Cells <-
1000*A.2020.Corn.Harvest.RN$Row+A.2020.Corn.Harvest.RN$Column
```

```
plot(Latitude ~ Longitude,data=A.2020.Corn.Harvest.RN,pch = ".")
abline(h=1:12*50,v=1:20*50,col='red')
```

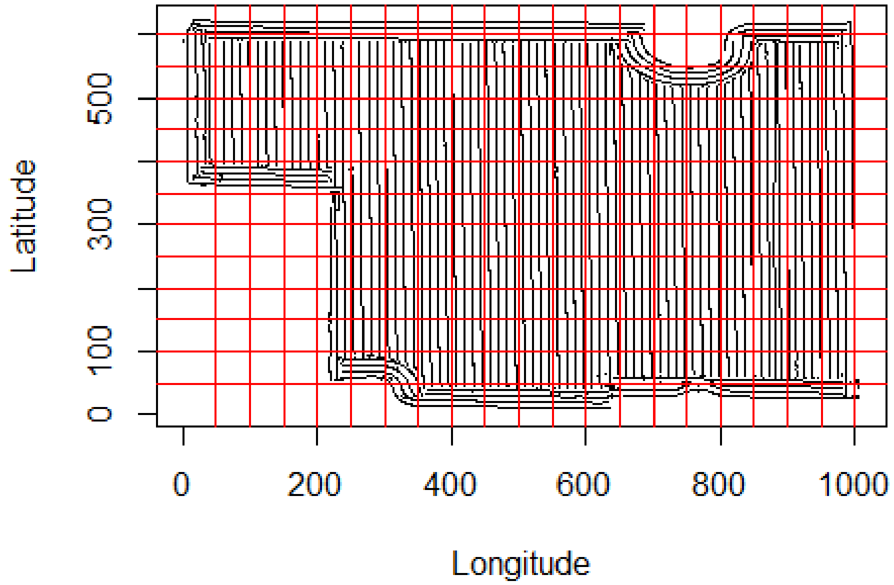

Figure 17: Harvest per grid of Normalized Data

```
plot(Row ~ Column,data=A.2020.Corn.Harvest.RN)
abline(h=1:12+0.5,v=1:20+0.5,col='red')
```

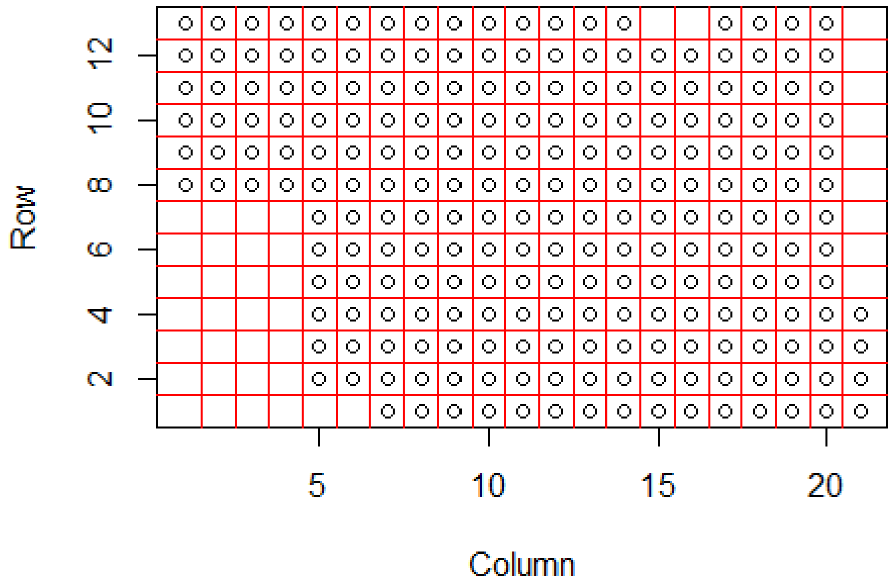

Figure 18: Grid with datapoints after normalization

Aggregate the ranked data

```
#Aggregate the data
#Applied Rate
AR18 <- aggregate(A.2018.Corn.Seeding.RN$AppliedRate,
by=list(A.2018.Corn.Seeding.RN$Cells), FUN=mean)
Samp.AR18 <- aggregate(A.2018.Corn.Seeding.RN$AppliedRate,
by=list(A.2018.Corn.Seeding.RN$Cells), FUN=length)

AR20 <-  aggregate(A.2020.Corn.Seeding.RN$AppliedRate,
by=list(A.2020.Corn.Seeding.RN$Cells), FUN=mean)
Samp.AR20 <- aggregate(A.2020.Corn.Seeding.RN$AppliedRate,
by=list(A.2020.Corn.Seeding.RN$Cells), FUN=length)

#Yield
Y17 <- aggregate(A.2017.Soybeans.Harvest.RN$RY17,
by=list(A.2017.Soybeans.Harvest.RN$Cells), FUN=mean)
Samp.Y17 <- aggregate(A.2017.Soybeans.Harvest.RN$RY17,
by=list(A.2017.Soybeans.Harvest.RN$Cells), FUN=length)

Y18 <- aggregate(A.2018.Corn.Harvest.RN$RY18,
by=list(A.2018.Corn.Harvest.RN$RY18), FUN=mean)
Samp.Y18 <- aggregate(A.2018.Corn.Harvest.RN$RY18,
by=list(A.2018.Corn.Harvest.RN$Cells), FUN=length)
```

```r
Y19 <- aggregate(A.2019.Soybeans.Harvest.RN$RY19,
by=list(A.2019.Soybeans.Harvest.RN$Cells), FUN=mean)
Samp.Y19 <- aggregate(A.2019.Soybeans.Harvest.RN$RY19,
by=list(A.2019.Soybeans.Harvest.RN$Cells), FUN=length)

Y20 <- aggregate(A.2020.Corn.Harvest.RN$RY20,
by=list(A.2020.Corn.Harvest.RN$Cells), FUN=mean)
Samp.Y20 <- aggregate(A.2020.Corn.Harvest.RN$RY20,
by=list(A.2020.Corn.Harvest.RN$Cells), FUN=length)

#Subset the data, only select data with at least 30 observations
AR18 <- na.omit(subset(AR18,Samp.AR18>30)) #drop the nulls associated with
subsetting the data
names(AR18)[1] <-'Cells'
names(AR18)[2] <- 'AR18'

AR20 <- na.omit(subset(AR20, Samp.AR20>30 ))
names(AR20)[1] <-'Cells'
names(AR20)[2] <- 'AR20'

Y17 <- na.omit(subset(Y17, Samp.Y17>30))
names(Y17)[1] <-'Cells'
names(Y17)[2] <- 'Y17'

Y18 <- na.omit(subset(Y18, Samp.Y18>30))
names(Y18)[1] <-'Cells'
names(Y18)[2] <- 'Y18'

Y19 <- na.omit(subset(Y19, Samp.Y19>30))
names(Y19)[1] <-'Cells'
names(Y19)[2] <- 'Y19'

Y20 <- na.omit(subset(Y20, Samp.Y20>30))
names(Y20)[1] <-'Cells'
names(Y20)[2] <- 'Y20'

merge1 <- merge(Y17, AR18, by="Cells")
merge2 <- merge(merge1,Y18, by="Cells")
merge3 <- merge(merge2, Y19, by="Cells")
merge4 <- merge(merge3, AR20, by="Cells")
Combined.dat <- merge(merge4, Y20, by="Cells")
Combined.dat.norm <- Combined.dat

pairs(Combined.dat.norm)
```

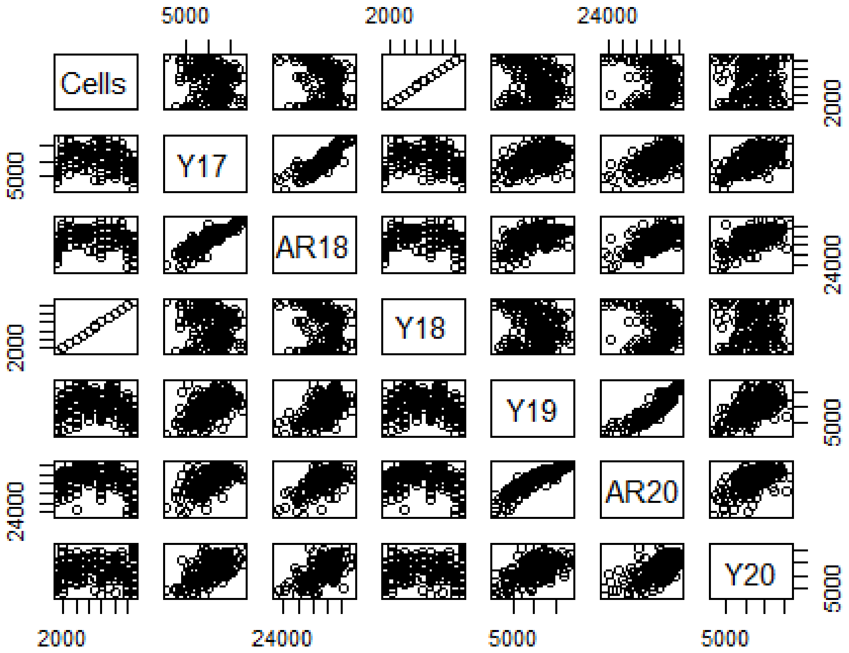

Figure 19: Pairs plot after normalization

```
par(mfrow=c(1,3))
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)
```

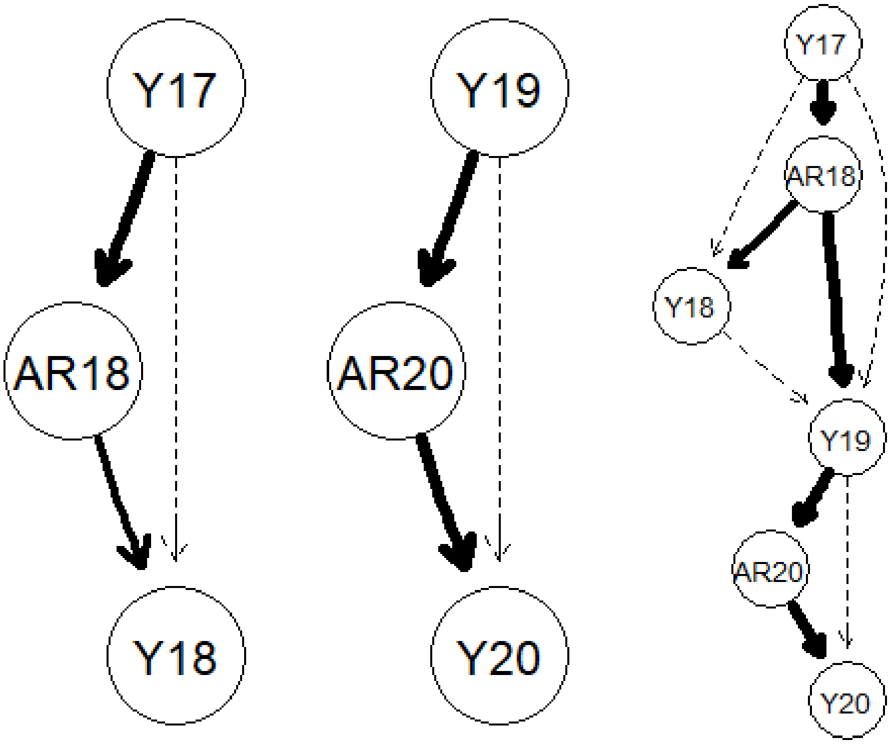

Figure 20: DAG plot after normalization

```
#reassign normalised data plot to be printed in final output
#without the reassigning, R will overwrite plot for the original dataset and
produce only results for the normalised data for both plots in my final
output
model1.dag.norm <- model1.dag
strength1.norm <- arc.strength(model1.dag.norm,
Combined.dat.norm[,c('Y17','AR18','Y18','Y19','AR20','Y20')])
```

```
par(mfrow=c(1,2))
strength.plot(model1.ori, strength.ori, main="Original Dataset")
strength.plot(model1.dag.norm, strength1.norm, main="Ranked Dataset")
```

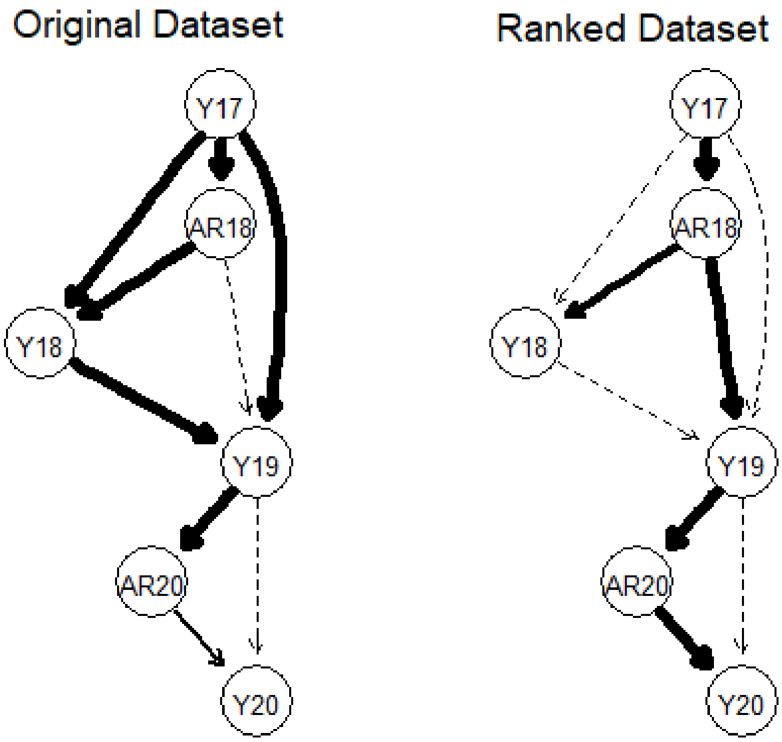

Figure 21:Final Output

CONCLUSION

The two plots show which variable has direct relationship with other variable. For example, if we look at the plot for the original dataset, we see that Y17(2017 yield) influences the AR18(2018 seeding AppliedRate).

For the plot associated with the normalized data, we see that there is a direct relationship from Y19 to AR20.

Another point to note is that the normalization(rank method) changed the results of our plot which intends influenced how we interpret how one variable is related to another.

CONTRIBUTIONS

Michael Kojo Abalo: Contributed to the data exploration, data cleaning, and normalization implementation. Also, responsible for creating visualizations associated with the normalized data.

Isaac Gbene: Contributed to the introduction, explained and defended why we should use the rank method, computing and interpretation of skewness and kurtosis. Additionally, responsible for scheduling meeting times and finding the perfect meeting place for each group member's convenience.

Prince Agyapong: Contributed to the data description, conclusion and was responsible for putting the work together in the expected order. Also, assisted in drawing conclusions based on the results.

All three members actively collaborated and contributed to the overall project; the experience was very interactive as we argued on how to implement our codes to get the desired results, but it was rewarding in the end.

APPENDIX

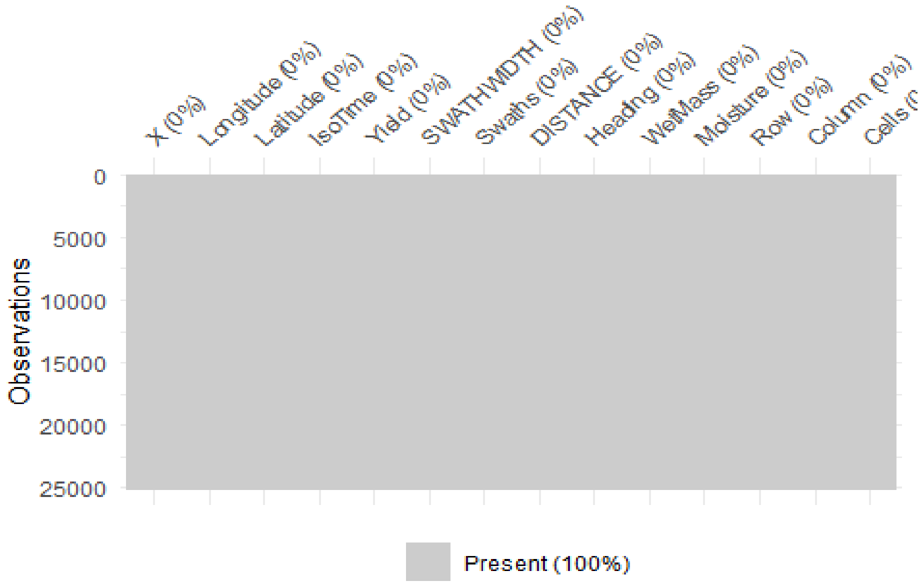

Figure 22:No Missing data in 2018 corn Harvest dataset

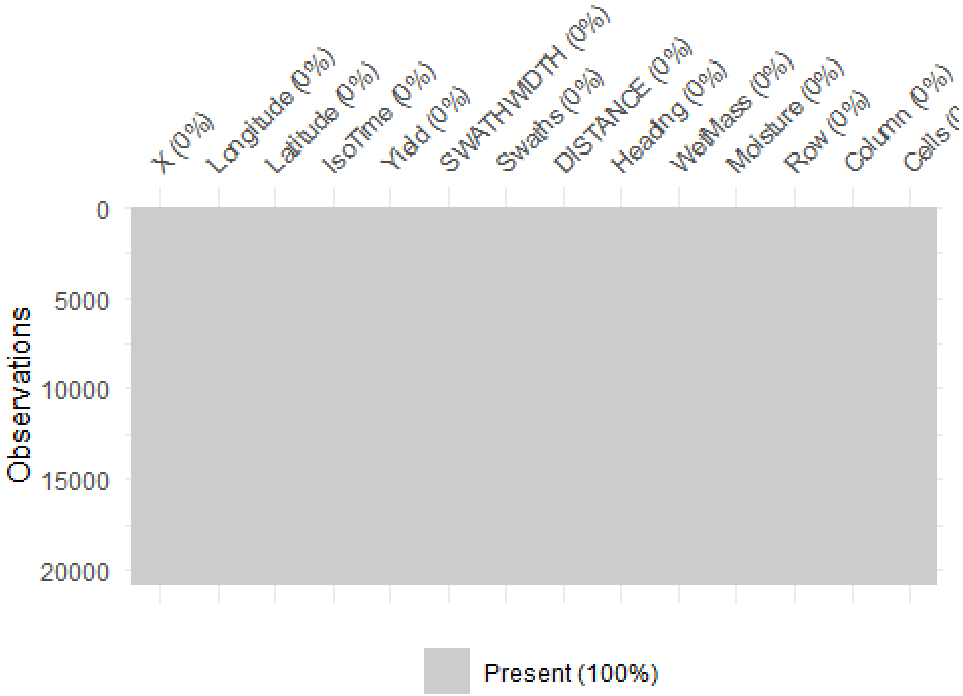

Figure 23:No Missing data in 2020 corn Harvest dataset

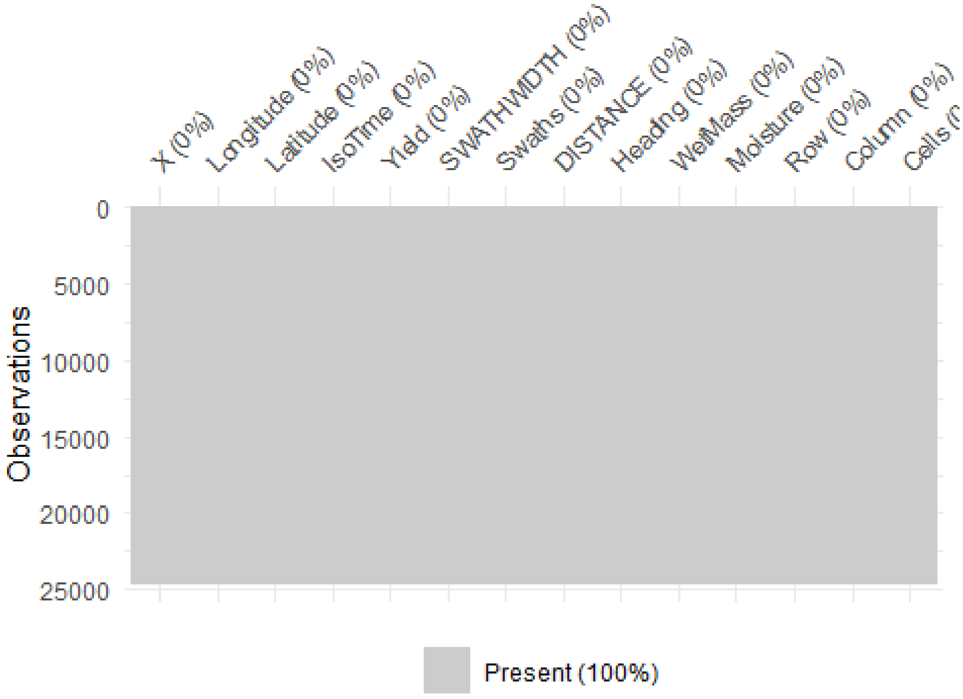

Figure 24: No Missing data in 2020 corn Harvest dataset

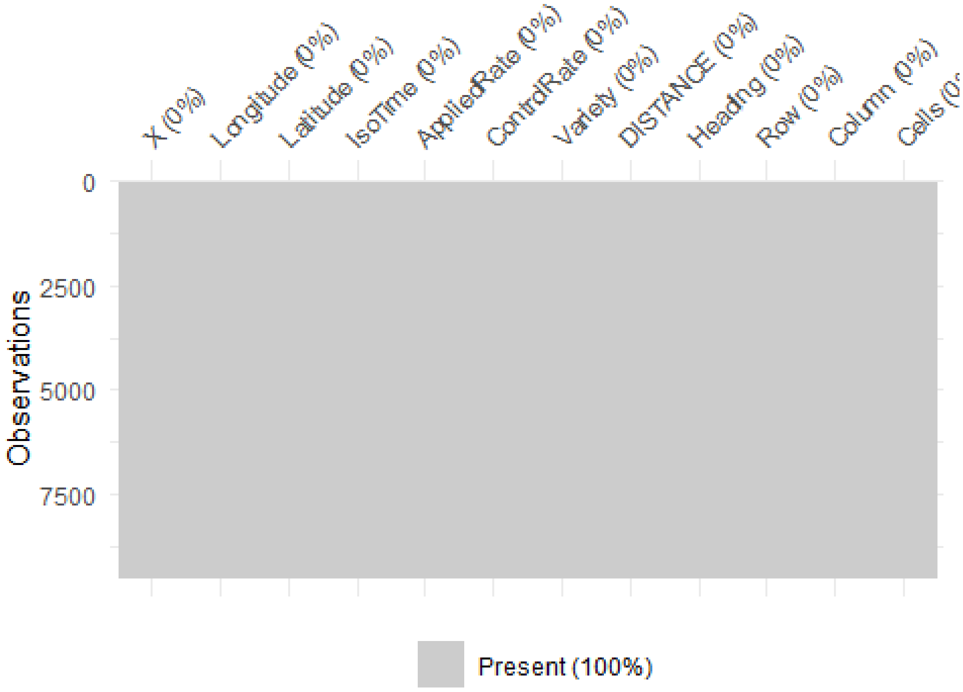

Figure 25: No Missing data in 2020 Corn seeding dataset