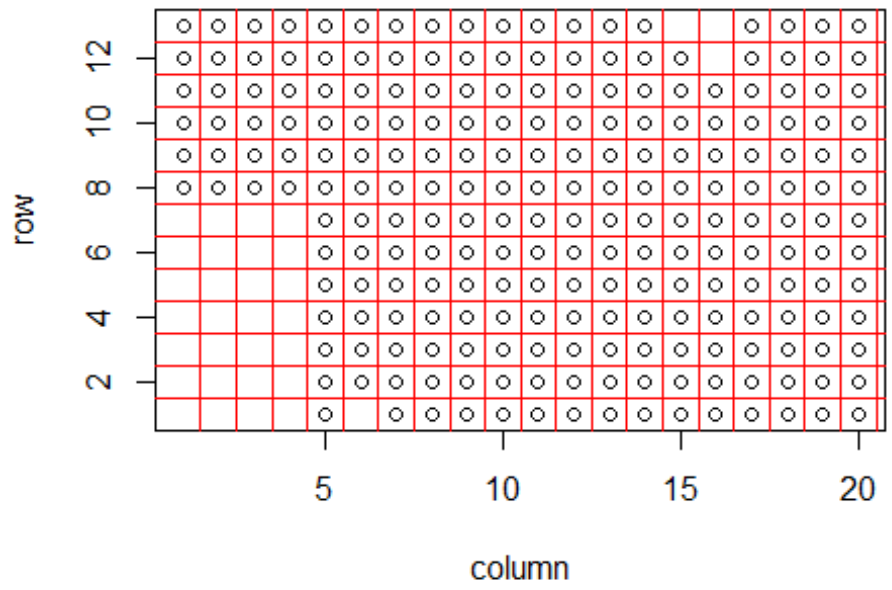
# Ben Loos Final Project STAT 600

The problem comes from data that is received from an agricultural field for crop performance. While there can be many factors that can contribute to how crops perform, we want to look at the seeding rate and if that has anything to do with the growth rate for certain crops. The reason we want to look at this variable is because it can be a great indicator for farmers' how if it can greatly impact on the productivity of the harvest. The goal of this analysis is to figure out if we can show how the data from recent years can help with the most recent data on past yields, and the current yields. The way we are going to do this is by aggregating the yield and seeding data in 50 m x 50m grid cells, which should help reduce noise and help create spatial framework. After we gather data on this, we will turn to normalizing the data to account for the different units across crops such as soybean yield being 60 bu/acre while corn yield is 180 bu/acre. This will allow for better comparison between the crops and will allow for better analysis on the question of whether the seeding rate can indicate how the yield will turn out. The variables that we will use from these data sets are the 'Northing' and 'Easting' variable to help organize and sort the data with these variables indciating the current location of the plots. Then we will use the 'yield' and 'AppliedRate' variables as our parameters for conducting our analysis and plotting.

##Grid - The grid will become the main focal point. We want to create the data in a grid to help show how the plots look somewhat like how they would be interpreted in the field. In addition, we will create a variable called "Cell" which will help distribute the data so that we can combine sections of the same field into the plot and make geospatial easier to interpret. We will use the 'Northing' and 'Easting' variables to help create the variable.

```
#read in the files
soyharvest17 <- read.csv("C:/Users/DSU/Downloads/A 2017 Soybeans
Harvest.csv")
 cornharvest18<- read.csv("C:/Users/DSU/Downloads/A 2018 Corn Harvest.csv")
cornseed18 <-read.csv("C:/Users/DSU/Downloads/A 2018 Corn Seeding.csv")
soyharvest19 <- read.csv("C:/Users/DSU/Downloads/A 2019 Soybeans
Harvest.csv")
cornharvest20 <- read.csv("C:/Users/DSU/Downloads/A 2020 Corn Harvest.csv")
cornseed20 <- read.csv("C:/Users/DSU/Downloads/A 2020 Corn Seeding.csv")

#create function for the spatial grid
gridfuction <- function(gridcells) {
  #the row variable should help define the northing variable into the grid
and help with creating cell variable
  gridcells$row <- ceiling(gridcells$Northing / 50)
  #Column variable goes with the cell variable but using the Easting variable
  gridcells$column <- ceiling(gridcells$Easting / 50)
  #Creating the cell variable to help identify spatial data with plots
  gridcells$cell <- gridcells$row * 1000 + gridcells$column
  return(gridcells)
}
```

```r
#convert the normal data frame to the grid and create variable row, column,
and the identifying variable cell
soyharvest17  <- gridfuction(soyharvest17)
cornharvest18 <- gridfuction(cornharvest18)
soyharvest19  <- gridfuction(soyharvest19)
cornharvest20 <- gridfuction(cornharvest20)
cornseed18 <- gridfuction(cornseed18)
cornseed20 <- gridfuction(cornseed20)

#Showcase of how the grid looks at a singular data set bases
plot(row ~ column,data=cornseed20)
abline(h=1:12+0.5,v=1:20+0.5,col='red')
```




## Aggregate data

```r
#aggregation of each data frame and produce a variable that fits the final
data format
soyharvest17agg <- aggregate(Yield ~ cell, data = soyharvest17, mean)
soyharvest17agg$Y17 <- soyharvest17agg$Yield
cornharvest18agg <- aggregate(Yield ~ cell, data = cornharvest18, mean)
cornharvest18agg$Y18 <- cornharvest18agg$Yield
cornseed18agg <- aggregate(AppliedRate ~ cell, data = cornseed18, mean)
cornseed18agg$AR18 <- cornseed18agg$AppliedRate
soyharvest19agg <- aggregate(Yield ~ cell, data = soyharvest19, mean)
soyharvest19agg$Y19 <- soyharvest19agg$Yield
cornharvest20agg <- aggregate(Yield ~ cell, data = cornharvest20, mean)
cornharvest20agg$Y20 <- cornharvest20agg$Yield
cornseed20agg <- aggregate(AppliedRate ~ cell, data = cornseed20 , mean)
cornseed20agg$AR20 <- cornseed20agg$AppliedRate
```

## Merge data

```
#Combining the individual aggregated data set into one central table with a
merged 'Cell' variable
combine <- merge(soyharvest17agg, cornharvest18agg, by = "cell")
combine <- merge(combine, cornseed18agg, by = "cell")
combine <- merge(combine, soyharvest19agg, by = "cell")
combine <- merge(combine, cornharvest20agg, by = "cell")

## Warning in merge.data.frame(combine, cornharvest20agg, by = "cell"):
column
## names 'Yield.x', 'Yield.y' are duplicated in the result

Combined <- merge(combine, cornseed20agg, by = "cell")

## Warning in merge.data.frame(combine, cornseed20agg, by = "cell"): column
names
## 'Yield.x', 'Yield.y' are duplicated in the result
```

To help with interpreting and aligning the data to make it easier for comparison we will normalize the data using the percent-of-mean. For this approach we will observe i in year j, the normalized value is defined as y_ij^*=100×y_ij/y ¯(·j) , *where* y ¯(·j ) swil be the mean of bu/acre that was given or the arithmetic mean of all observations for that variable in year j. This method will convert the values to percentages relative to the yearly average, allowing meaningful comparisons across crops with different units. This normalization within year will preserve spatial variation while removing scale differences across the dataset. By multiplying the values by 100 we can indicate whether there is above-average or below-average yield performance. While not the only way for normalization, it is best for making the results easier to interpret spatially and more suitable for causal analysis involving agricultural decisions where exact values are less important than if performance is exceeding or fewer than the projection.

```
#creating a function to make it easier to bring the percent to every
measurement that we want to look at.we want to
percentseeding <- function(x) {
    100 * x / mean(x, na.rm = TRUE)
}
percentsoybean <- function(x) {
    100 * x / 60
}
percentcorn <- function(x) {
    100 * x / 180
}
#Bringing the percent function into every variable that we intend to look at
soyharvest17$Yieldnorm <- percentsoybean(soyharvest17$Yield)
cornharvest18$Yieldnorm <- percentcorn(cornharvest18$Yield)
cornseed18$AppliedRatenorm <- percentseeding(cornseed18$AppliedRate)
soyharvest19$Yieldnorm <- percentsoybean(soyharvest19$Yield)
cornharvest20$Yieldnorm <- percentcorn(cornharvest20$Yield)
cornseed20$AppliedRatenorm <- percentseeding(cornseed20$AppliedRate)
```

```
# We will aggregate the new data to help with making the data easier to read
when we build the pairs plot
soyharvest17normagg <- aggregate(Yieldnorm ~ cell, data = soyharvest17, mean)
soyharvest17normagg$Y17 <- soyharvest17normagg$Yieldnorm
cornharvest18normagg <- aggregate(Yieldnorm ~ cell, data = cornharvest18,
mean)
cornharvest18normagg$Y18 <- cornharvest18normagg$Yieldnorm
cornseed18normagg <- aggregate(AppliedRatenorm ~ cell, data = cornseed18,
mean)
cornseed18normagg$AR18 <- cornseed18normagg$AppliedRatenorm
soyharvest19normagg <- aggregate(Yieldnorm ~ cell, data = soyharvest19, mean)
soyharvest19normagg$Y19 <- soyharvest19normagg$Yieldnorm
cornharvest20normagg <- aggregate(Yieldnorm ~ cell, data = cornharvest20,
mean)
cornharvest20normagg$Y20 <- cornharvest20normagg$Yieldnorm
cornseed20normagg <- aggregate(AppliedRatenorm ~ cell, data = cornseed20 ,
mean)
cornseed20normagg$AR20 <- cornseed20normagg$AppliedRatenorm
```

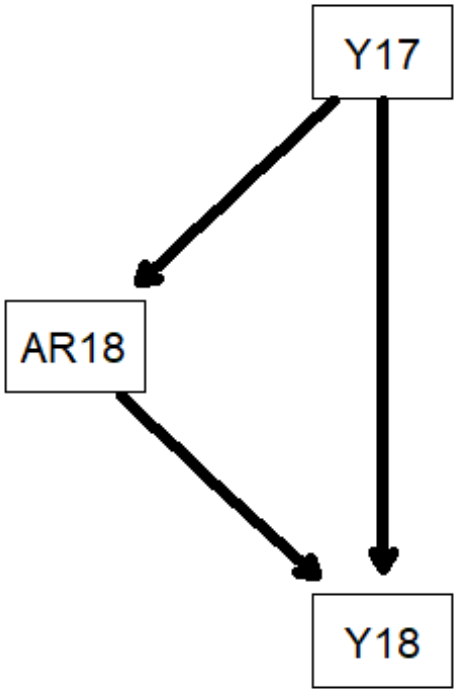
Combining the data sets together to help with comparison between the yields and seeding.

```
combinenorm <- merge(soyharvest17normagg, cornharvest18normagg, by = "cell")
combinenorm <- merge(combinenorm, cornseed18normagg, by = "cell")
combinenorm <- merge(combinenorm, soyharvest19normagg, by = "cell")
combinenorm <- merge(combinenorm, cornharvest20normagg, by = "cell")

## Warning in merge.data.frame(combinenorm, cornharvest20normagg, by =
"cell"):
## column names 'Yieldnorm.x', 'Yieldnorm.y' are duplicated in the result

CombinedNorm <- merge(combinenorm, cornseed20normagg, by = "cell")

## Warning in merge.data.frame(combinenorm, cornseed20normagg, by = "cell"):
## column names 'Yieldnorm.x', 'Yieldnorm.y' are duplicated in the result
```

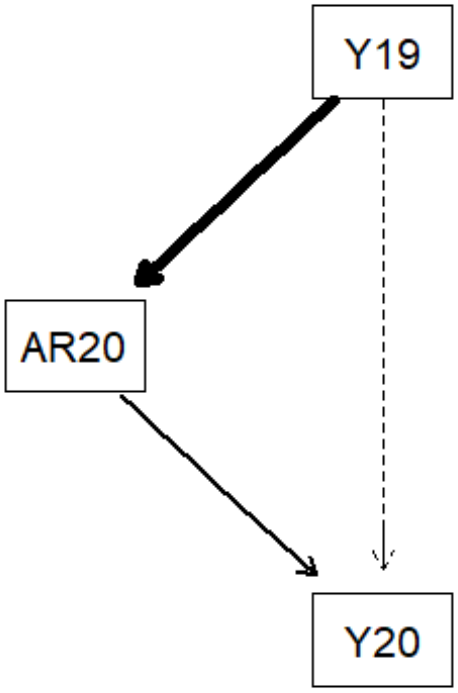
- Here we are displaying how the variables that we created in the grid will help define how we produce the results and findings.It's a great visual to show connection and relationship of the variables.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
install.packages("bnlearn")

BiocManager::install("Rgraphviz")

library(bnlearn)

## Warning: package 'bnlearn' was built under R version 4.5.2

modela.dag <- model2network("[Y17][AR18|Y17][Y18|AR18:Y17]")
fita = bn.fit(modela.dag, Combined[,c('Y17','AR18','Y18')])
```

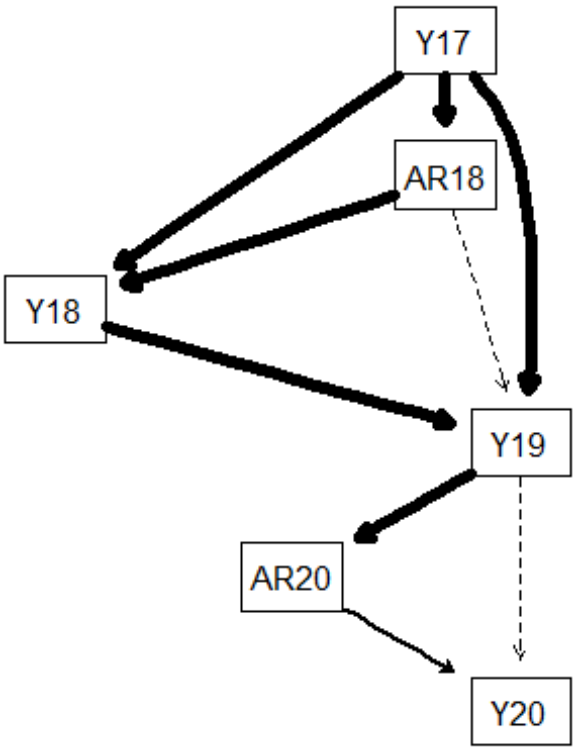
```
strengtha <- arc.strength(modela.dag, Combined[,c('Y17','AR18','Y18')])
strength.plot(modela.dag, strengtha)

## Loading required namespace: Rgraphviz
```




```
modelb.dag <- model2network("[Y19][AR20|Y19][Y20|AR20:Y19]")
fitv = bn.fit(modelb.dag, Combined[,c('Y19','AR20','Y20')])

strengthb <- arc.strength(modelb.dag, Combined[,c('Y19','AR20','Y20')])
strength.plot(modelb.dag, strengthb)
```



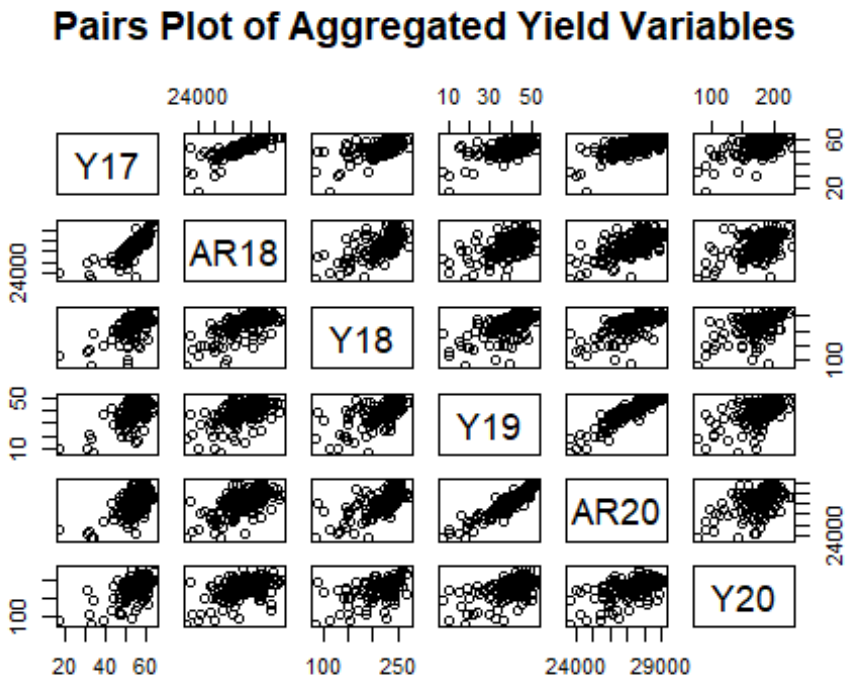

```
modelc.dag <-
model2network("[Y17][AR18|Y17][Y18|AR18:Y17][Y19|Y17:AR18:Y18][AR20|Y19][Y20|
AR20:Y19]")
fitc = bn.fit(modelc.dag,
Combined[,c('Y17','AR18','Y18','Y19','AR20','Y20')])

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




##Pair Plots

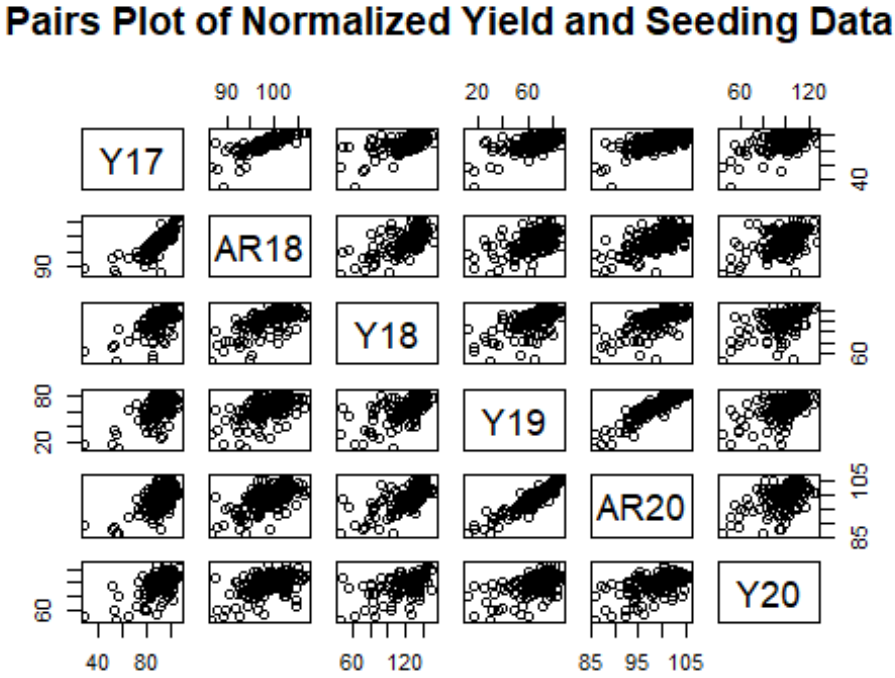
#Original using pair plot to showcase how the data columns represent the yield variables

```
pairs(Combined[, c('Y17', 'AR18', 'Y18', 'Y19', 'AR20', 'Y20')], main =
"Pairs Plot of Aggregated Yield Variables")
```




#Normalized Plot

```
pairs(CombinedNorm[, c('Y17', 'AR18', 'Y18', 'Y19','AR20','Y20')],
    main = "Pairs Plot of Normalized Yield and Seeding Data")
```



The output of the analysis shows what the use of normalization looks for the seeding plots. As we can see in the results there is not a huge difference what we see from the original data and the new normalized data set. This could indicate that there are not huge fluctuation within the variables that can change the data .By aggregating yield and seeding data into consistent spatial grid cells and normalizing values across years, the analysis separates management effects from inherent field productivity. Including historical yield allows us to distinguish whether low corn yields are primarily associated with reduced seeding rates or reflect persistently low-yielding areas of the field. This structure supports causal inference by accounting for prior productivity when evaluating the effect of seeding rate on yield.