STAT 600 - Final Project

John Herbert

08/07/20

Project Statement

This project is to determine the classification of grid cells within a plot (400x600 meters) and determine if these grids have a low, average, or high yield rate. In addition, we will determine if these grid cells have stable, average, or unstable yields. This will be done by normalizing the estimates based on a ranking methodology.

The optimal grid cells have been predetermined to have 6 columns due to the harvester size, and 20 rows has been determined by optimation analysis in the Grid Cell Size section (120 grid cells total). We will then plot the results over each yeart's harvest formation for the merged data as well as each years data to see how each plot yielded and how stable those yield values are.

Document External Libraries

We have used *dplyr* package for data manipulation. We will use the *select()* and *filter()* functions to pick the variables based on their names and their values, respectively. We will also use *mutate* and *group_by* to determine ranking of estimates by year or data set.

The *ggplot2* package is used for plot creation and the *RColorBrewer* is used for creating color palettes for a more aesthetic graphs and to color the years for the plots. *gridExtra* is used to format graphs side by side.

```
# install.packages('dplyr')
suppressWarnings(suppressMessages(library(dplyr)))
# install.packages('ggplot2')
suppressWarnings(suppressMessages(library(ggplot2)))
# install.packages('RColorBrewer')
suppressWarnings(suppressMessages(library(RColorBrewer)))
# install.packages('gridExtra')
suppressWarnings(suppressMessages(library(gridExtra)))
```

Data Sets

Data sets are by year for 3 different types of crops on the same plot of land. The data frames are imported into a list to make iternation and calculations easier.

```
df <- list(home.2013=read.csv('home.2013.csv',header = T),
           home.2015=read.csv('home.2015.csv',header = T),
           home.2016=read.csv('home.2016.csv',header = T),
```

```
            home.2017=read.csv('home.2017.csv',header = T),
            home.2018=read.csv('home.2018.csv',header = T))
```

Grid Cell Size

Formula to Compute Required Replicates

$$n \geq 2 \times \left(\frac{CV}{\%Diff}\right)^2 \times \left(z_{\alpha/2} + z_{\beta}\right)^2$$

where $\%Diff = \frac{m_1 - m_2}{(m_1+m_2)/2}$ and $CV = \frac{sd_{pooled}}{(m_1+m_2)/2}$.

Formula used to help validate the grid cell size for classification and analysis by calculating the required replicates (RR).

```
rr.func <- function(g_mean,sd,dif,alpha = 0.05, beta = 0.20) {
  z_alpha <- qnorm(1-alpha/2)
  z_beta <- qnorm(1-beta)
  cv <- sd/g_mean
  return(ceiling(2*((cv/dif)^2)*((z_alpha + z_beta)^2)))
}
```

Validating Grid Cell Size

Function for Required Replicates to Optimize Rows

A function to graph a plot to determine the optimal number of rows based on 6 columns. The function iterates through each data set or year of data, then iterates through each iteration of rows: 1,2…40 to calculate the estimates and RR for each row split. It then creates a data frame that is used to plot RR vs Estimates and CV vs Estimates.

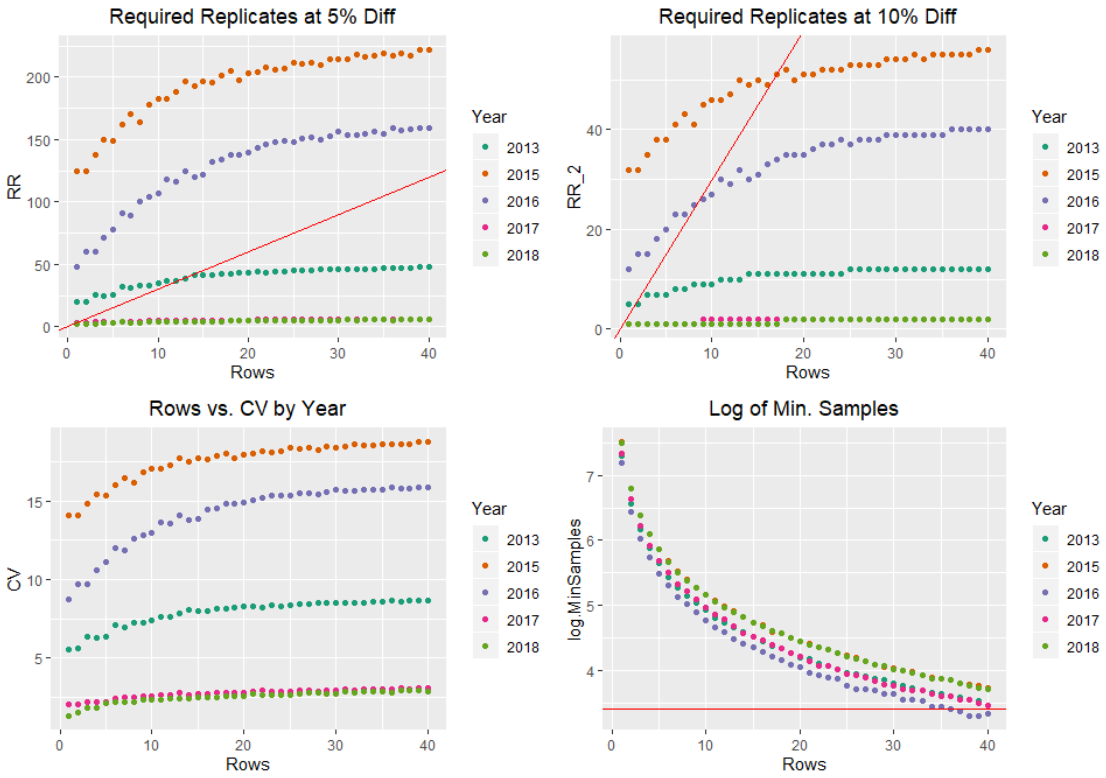
```
optrr.func <-
function(years,max_long,max_lat,lat_seq,long_length,dif_1,dif_2){
  #begin function
  # Determining length of the data frame to store results from the function
  optrr.length <- length(years)*tail(lat_seq,n=1)
  # Empty data frame used to store Year,CV, Rows, and RR
  optrr.dat <- data.frame(Year=numeric(length=optrr.length),
                          CV=numeric(length=optrr.length),
                          Rows=numeric(length=optrr.length),
                          RR=numeric(length=optrr.length),
                          RR_2=numeric(length=optrr.length),
                          log.MinSamples=numeric(length=optrr.length))
  k=1
  for(j in 1:length(years)){ # begin 1st for loop
    g_mean <- mean(df[[j]]$Yield)
    for(i in lat_seq){ # begin 2nd for loop
      n = 1
      #Temporary vector to store observations per grid cell to calculate the
```

```
estimates
      temp.est <- vector(length=lat_seq[i]*long_length)
      # Temporary vector to store sample sizes to calculate the minimum
sample number
      n_sum.est <- vector(length=lat_seq[i]*long_length)
      #Loop to determine which observations go into which grid cell
      for(long in 1:long_length){ # begin 3nd for loop
        for(lat in 1:lat_seq[i]){ # begin 4th for loop
          x <- df[[j]] %>% select(Yield,Longitude,Latitude) %>%
               filter(Longitude <= max_long/long_length * long &
                      Latitude <= max_lat/lat_seq[i] * lat &
                      Longitude >= max_long/long_length * (long-1) &
                      Latitude >= max_lat/lat_seq[i]* (lat-1)
               )
          temp.est[n] <- mean(x$Yield)
          n_sum.est[n] <- length(x$Yield)
          n <- n + 1
        } # end 4th for loop
      } # end 3rd for loop
      optrr.dat$Year[k] <- year[j]

      optrr.dat$CV[k] <- sd(temp.est)/g_mean*100
      optrr.dat$Rows[k] <- i
      # Running the RR function to determine RR per iteration
      # Using grand mean for RR calculations (mean for all observations
within a year)
      # Using the sample standard deviation for RR calculations
      # (SD for all estimates within a year)
      optrr.dat$RR[k] <- rr.func(g_mean,sd(temp.est),dif)
      optrr.dat$RR_2[k] <- rr.func(g_mean,sd(temp.est),dif_2)
      # Calculating the minimum number of samples per iteration then
converting to log
      optrr.dat$log.MinSamples[k] <- log(min(n_sum.est))
      k <- k + 1
    } # end 2nd for loop
  } # end 1st for loop
  # Converting the years to character string for graphing purposes
  optrr.dat$Year <- as.character(optrr.dat$Year)
  return(optrr.dat)
} # end function
```

Plots for RR of Grid Cells

```
year <- c(2013,2015,2016,2017,2018)
max_long <- 600
max_lat <- 400
lat_seq <- 1:40
long_length <- 6
dif = 0.05
dif_2 = 0.1
plot <- list()
```

```r
# Running optimal RR function to prepare data for graphing at diff = 5% & 10%
optrr.dat <- optrr.func(year,max_long,max_lat,lat_seq,long_length,dif,dif_2)
# Plotting data using ggplot and scale+_color_brewer to year color
# palette
plot[[1]]<-(ggplot(optrr.dat,aes(Rows,RR,colour = Year)) + geom_point() +
 scale_color_brewer(palette = 'Dark2') +
 ggtitle('Required Replicates at 5% Diff') +
 # Creates line at slope = 3 to show RR boundries
 geom_abline(intercept = 0, slope = 3, col = 'red') +
 theme(plot.title = element_text(hjust = 0.5)))

plot[[2]]<-(ggplot(optrr.dat,aes(Rows,RR_2,colour = Year)) + geom_point() +
 scale_color_brewer(palette = 'Dark2') +
 ggtitle('Required Replicates at 10% Diff') +
 geom_abline(intercept = 0, slope = 3, col = 'red') +
 theme(plot.title = element_text(hjust = 0.5)))

plot[[3]]<-(ggplot(optrr.dat,aes(Rows,CV,colour = Year)) + geom_point() +
 scale_color_brewer(palette = 'Dark2') + ggtitle('Rows vs. CV by Year') +
 theme(plot.title = element_text(hjust = 0.5)))

plot[[4]]<-(ggplot(optrr.dat,aes(Rows,log.MinSamples,colour = Year)) +
geom_point()
  + scale_color_brewer(palette = 'Dark2') + ggtitle('Log of Min. Samples') +
    geom_abline(intercept = log(30), slope = 0, col = 'red') +
    theme(plot.title = element_text(hjust = 0.5)))

grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],ncol=2)
```



The graphs are showing rows vs. RR and plotting each estimate by year. Each year is given a different color. In addition for reference, a rows vs. CV graph is given. Each of the rows. vs. RR graphs calculate a difference percent at 5% and 10%. The line shows a slope = 3 and intercept = 0. This is because there are 6 rows available, which gives 3 reps of 2 treatments available. So at rows = 20, there would be 60 grids available. Finally, the log of minimum samples shows the minimum sample number per estimate per row iteration to confirm that our sample size is more than the minimum threshold of 30 samples.

Based on this analysis, 20 rows whould be the optimal number to determine grid size. At rows = 20 (60 grids available for treatments), the RR at difference = 5% covers 3 of the 5 years, while at 10% it covers all years. In addition, the Log of Min. Samples graph shows that at 20 rows, we meet the minimum sample thresold of 30 observations per estimate (log of 30 samples ~ 3.40).

Data Screening

We are evaluating the harvest times for each of the yearly harvests to confirm that the entire harvest happened within a week's time.

The below analysis iterates through each year and calculates the difference between the minimum observation TimeStamp and each TimeStamp observation in the data set. If the difference is greater than 7, it gets a value of 1, if it is equal to 7 or less, it gets a value of 0. The values are then summed up per year. If any of the values are greater than 0, it violates the harvest timing threshold.

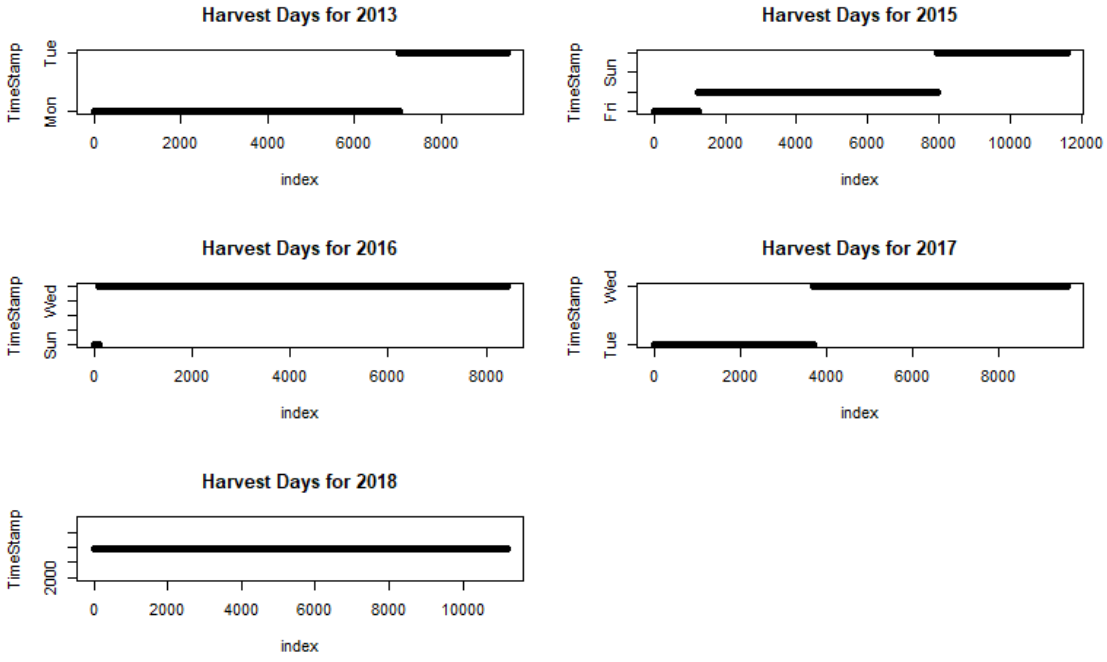
```
# Empty vector that will hold the summation of the screening analysis
ds <- vector(length=length(df))

for(j in 1:length(df)){ # begin 1st for loop
  # Transforms the TimeStamp data to a date format
  # I rounded the date to exclude hour and minute as it is not necessary
  # for this analysis
  df[[j]]$TimeStamp <- as.Date(as.character(df[[j]]$TimeStamp),'%Y-%m-%d')
  for(i in 1:length(df[[j]]$TimeStamp)){ # begin 2nd for loop
    # If statement that determines if the difference is less than or equal to
    # 7 days
    if(df[[j]]$TimeStamp[i]-(min(df[[j]]$TimeStamp))<=7){ # begin if
statement-True
      df[[j]]$Harvest[i] <- 0} # end if statement - True
    else df[[j]]$Harvest[i] <- 1
  } # end 2nd for loop
} # end 1st for loop
# Loop to sum Harvest column per year
for(i in 1:length(df)){ # begin for loop
  ds[i] <- sum(df[[i]]$Harvest)
} # end for loop

ds

## [1] 0 0 0 0 0
```

According to this analysis, each year was harvested within one week.

Graphical Confirmation of Data Screening

The loop below iterates through each year of data to plot the harvest to confirm that the time between harvest days is within 7 days. The graph plots the time stamp days of harvest over each observation per year.

```
par(mfrow=c(3,2))

for(j in 1:length(year)){
  index <- 1:length(df[[j]][,1])
  plot(df[[j]]$TimeStamp~index,main = paste0('Harvest Days for ',year[j]),
       ylab='TimeStamp')
}
```



The plots confirm our initial analysis that for each year the harvest days are within one week of each other. Harvest days for 2013 where Monday and Tuesday in the same week, 2015 was Friday, Saturday, and Monday in the same week, 2016 was Sunday and Thursday within the same week, 2017 was Tuesday and Wednesday within the same week, and 2018 was all done on the same day.

Checking Latitude and Longitude Boundries

Since 2 new data sets have been added to the analysis, we ran the the min and max of the longitude and latitude to confirm the plot size is the same across the data sets.

```
# Data frame used to store min/max data for longitude and latitude
long_lat_check <- data.frame(Year=year, lat_min=numeric(length = length(df)),
                             lat_max=numeric(length = length(df)),
                             long_min=numeric(length = length(df)),
                             long_max=numeric(length = length(df)))
# Loop to add data to the data frame across the years.
for(j in 1:length(df)){
  long_lat_check[j,2] <- min(df[[j]][,2])
  long_lat_check[j,3] <- max(df[[j]][,2])
  long_lat_check[j,4] <- min(df[[j]][,3])
  long_lat_check[j,5] <- max(df[[j]][,3])
}
long_lat_check

##   Year    lat_min  lat_max    long_min long_max
## 1 2013 0.08080050 399.9753 3.204898069 599.9816
## 2 2015 0.05954039 399.9310 6.398265795 595.7949
## 3 2016 0.01256038 399.9681 0.001517539 599.9697
```

```
## 4 2017 0.02201339 399.9977 0.027875273 599.8765
## 5 2018 0.03994049 399.9872 0.053314896 597.0580
```

We can see that all the data sets or years fall with 0 to 400 latitude and 0 to 600 longitude.

Normalization

For the 5 years we have 3 different crops that produce different mean yield values, therefore we need to normalize the estimates in order to merge all 5 years.

Option 1: Rank

The normalization method we chose is ranking each grid cell by yield estimates using the following formula:

$\{r_{ij}=r_{1j},r_{2j},{...},r_{Ij}}$ where r_{ij} is the r is the ranking of the estimates per year and j is the number of years where $J = 5$ for years $\{2013,2015,\ldots,2018\}$. i is the number of estimates per year where $I = 120$ for the number of grid cells per year.

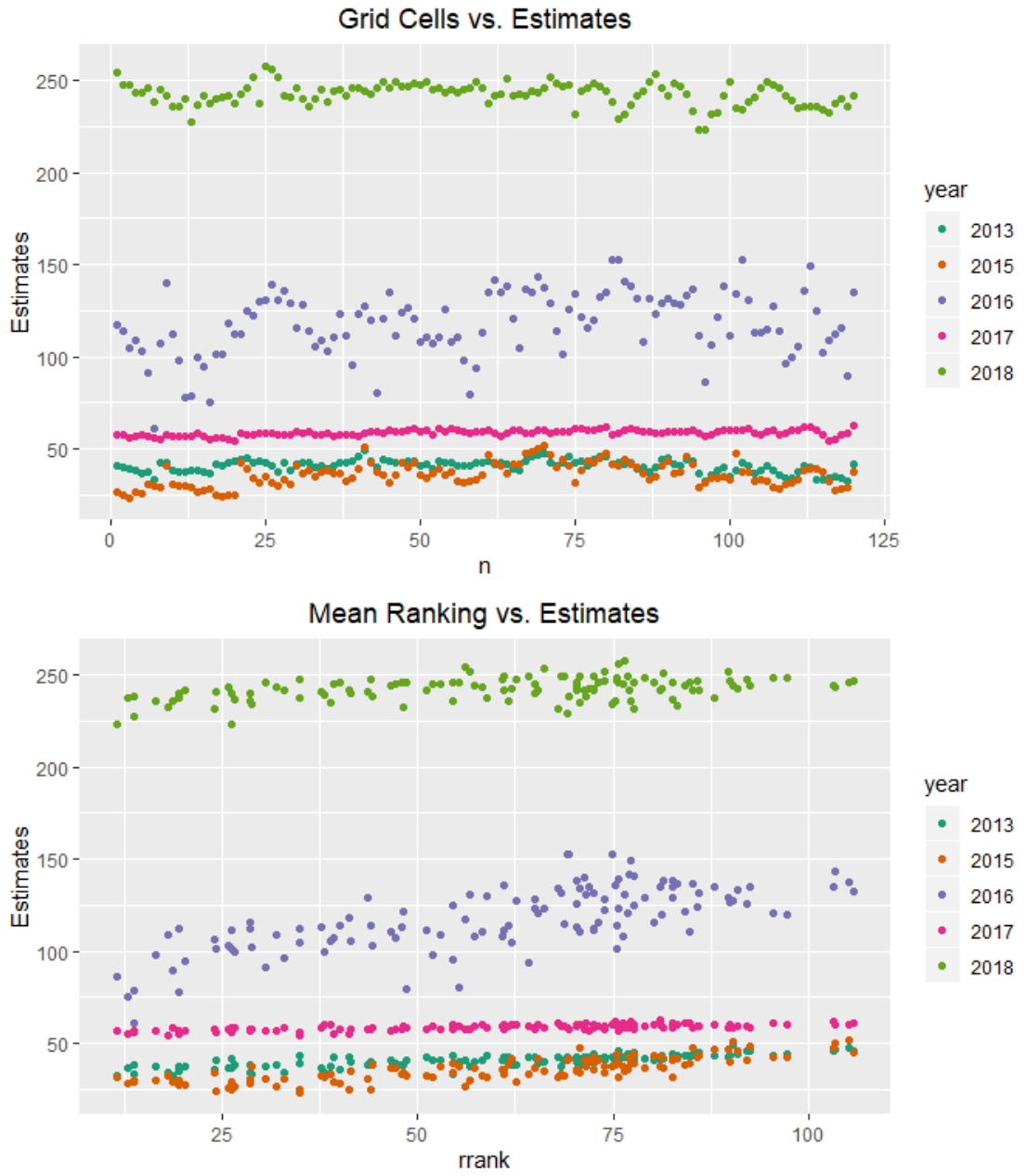
```
# Setting the number of rows and columns based on the grid size
lat_length <- 20
long_length <- 6
k=1
est.length <- lat_length*long_length*length(year)
est.dat <- data.frame(year=numeric(est.length),
Estimates=numeric(est.length),
                      n=numeric(est.length),

obs.sd=numeric(est.length),mu_class_y=numeric(est.length),
                      sd_class_y=numeric(est.length))

# Adding blank column for reference to add normalized mean and std. dev.
# to orginial data frames
for(j in 1:length(year)){
  df[[j]]$Ref_k <- numeric(length=dim(df[[j]])[1])
  df[[j]]$Ref_n <- numeric(length=dim(df[[j]])[1])
}

# Loop to calculate estimates for each grid cell
for(j in 1:length(year)){ # begin 1st for loop
  n=1
  for(long in 1:long_length){ # begin 2nd for loop
    for(lat in 1:lat_length){ # begin 3rd for loop
      x <- df[[j]] %>% select(Yield,Longitude,Latitude) %>%
           filter(Longitude <= max_long/long_length * long &
                  Latitude <= max_lat/lat_length * lat &
                  Longitude >= max_long/long_length * (long-1) &
                  Latitude >= max_lat/lat_length* (lat-1)
           )
      est.dat$year[k] <- year[j]
```

```
      est.dat$Estimates[k] <- mean(x$Yield)
      est.dat$n[k] <- n
      # Standard Deviation for each grid cell among the observations per year
      est.dat$obs.sd[k] <- sd(x$Yield)

      # Setting the grid cell number for merger of yearly data sets
      # Setting the reference number for merging and graphing purposes
      for(i in 1:dim(df[[j]])[1]){ # begin 4th for loop
        if(df[[j]]$Longitude[i] <= max_long/long_length * long &&
           df[[j]]$Latitude[i] <= max_lat/lat_length * lat &&
           df[[j]]$Longitude[i] >= max_long/long_length * (long-1) &&
           df[[j]]$Latitude[i] >= max_lat/lat_length* (lat-1)){
           df[[j]]$Ref_k[i] <- k
           df[[j]]$Ref_n[i] <- n
        }
      } # end 4th for loop
      k <- k + 1
      n <- n + 1
     } # end 3rd for loop
    } # end 2nd for loop
} # end 1st for loop
# Ranking each estimate by year for normalization
est.dat <- est.dat %>% group_by(year) %>% mutate(rank=rank(Estimates))
est.dat <- est.dat %>% group_by(n) %>% mutate(rrank=mean(rank))
# Converting the year column to category set for plotting purposes
est.dat$year <- as.character(est.dat$year)
```

Graphing the Rankings

First graph compares the yield estimates per grid cell over year. This plots the estimates by grid cell location.The second graph plots the estimates for each grid cell over the rankings (1 = smallest yield estimate, 120 = largest yield estimate) per year.

```
plot=list()
plot[[1]] <- (ggplot(est.dat,aes(n,Estimates,colour = year)) + geom_point()
  + scale_color_brewer(palette = 'Dark2') + ggtitle('Grid Cells vs.
Estimates')
  + theme(plot.title = element_text(hjust = 0.5)))

plot[[2]] <- (ggplot(est.dat,aes(rrank,Estimates,colour = year)) +
geom_point()
  + scale_color_brewer(palette = 'Dark2')+ ggtitle('Mean Ranking vs.
Estimates')
  + theme(plot.title = element_text(hjust = 0.5)))

grid.arrange(plot[[1]],plot[[2]],ncol=1)
```



The first graph confirms that there is a large discrepency between yield sizes per year depending on the crops harvested that year. The second graph confirms the ranking calculation is reasonably ordered. The x axis is pooled ranking per grid cell across the years and the the y axis is mean yield values for each year.

Merging Data

We have merged the mean rank,Latitude and Longitude of the 5 different years from 120 grids. Similarly, merged standard deviation of mean rank,Latitude and Longitude of the 5 different years from 120 grids.

```
est_merge.dat <- data.frame(n=1:120,
                  mu=numeric(length=120),
                  sd=numeric(length=120),
                  mu_class=numeric(length=120),
                  sd_class=numeric(length=120))
# Setting mean rank for each grid cell across each year
suppressMessages(mu <- est.dat %>% group_by(n) %>% summarize(mean(rank)))
# Setting standard deviation of normalized grid cell yield values across each
year
suppressMessages(sd <- est.dat %>% group_by(n) %>% summarize(sd(rank)))
est_merge.dat[2] <- mu[2]
est_merge.dat[3] <- sd[2]
```

Classification

Classified the 120 grids in the field based on the normalized rank following below rules:

1. If the normalized rank fall in the largest 25% of all the cells, classified the grid as “High Yielding”.
2. If the normalized rank fall in the smallest 25% of all the cells, classified the grid as “Low Yielding”.
3. All the other grids are classified as “Average Yield”.

Similarly, the field was classified as Stable Yielding, Unstable Yielding and Average Yield based on the Standard deviation of the mean estimates following below rules:

1. If the normalized Standard deviation fall in the largest 25% of all the cells classified the grid as “Unstable Yielding”.
2. If the normalized rank fall in the smallest 25% of all the cells classified the grid as “Stable Yielding”.
3. All the other grids are classified as “Average Yield”.

```
# for loop/ if statement to set mean and standard deviation classification
# for merged data.
for(i in 1:dim(est_merge.dat)[1]){
  # Using quantiles for 25% and 75% boundries of classification
  if(est_merge.dat$mu[i] >= quantile(est_merge.dat$mu)[4]){
    est_merge.dat$mu_class[i] <- 'High'}
  else if(est_merge.dat$mu[i] <= quantile(est_merge.dat$mu)[2]){
    est_merge.dat$mu_class[i] <- 'Low'}
  else est_merge.dat$mu_class[i] <- 'Average'

  if(est_merge.dat$sd[i] >= quantile(est_merge.dat$sd)[4]){
    est_merge.dat$sd_class[i] <- 'Unstable'}
  else if(est_merge.dat$sd[i] <= quantile(est_merge.dat$sd)[2]){
    est_merge.dat$sd_class[i] <- 'Stable'}
  else est_merge.dat$sd_class[i] <- 'Average'
}
# Creating arrays for mean and standard deviation quantiles for each of the
years
```

```r
q_m <- do.call('rbind',tapply(est.dat$Estimates,est.dat$year, quantile))
q_s <- do.call('rbind',tapply(est.dat$obs.sd,est.dat$year, quantile))
# for loop/if statement to add mean and standard deviation classifications
# for each year
for(j in 1:length(year)){
  for(i in 1:dim(est.dat)[1]){
    if(est.dat$Estimates[i] >= q_m[j,4] &&
       est.dat$year[i] == year[j]){
       est.dat$mu_class_y[i] <- 'High'}
    else if(est.dat$Estimates[i] <= q_m[j,2] &&
       est.dat$year[i] == year[j]){
       est.dat$mu_class_y[i] <- 'Low'}
    else if(est.dat$Estimates[i] < q_m[j,4] &&
       est.dat$Estimates[i] > q_m[j,2] &&
       est.dat$year[i] == year[j]){
       est.dat$mu_class_y[i] <- 'Average'}

    if(est.dat$obs.sd[i] >= q_s[j,4] &&
       est.dat$year[i] == year[j]){
       est.dat$sd_class_y[i] <- 'Unstable'}
    else if(est.dat$obs.sd[i] <= q_s[j,2] &&
       est.dat$year[i] == year[j]){
       est.dat$sd_class_y[i] <- 'Stable'}
    else if(est.dat$obs.sd[i] < q_s[j,4] &&
       est.dat$Estimates[i] > q_s[j,2] &&
       est.dat$year[i] == year[j]){
       est.dat$sd_class_y[i] <- 'Average'}
  }
}

# Creating columns for mean, standard deviation, and classification for
# each of the original data sets for graphing/analysis purposes
for(j in 1:length(year)){
  df[[j]]$mu <- numeric(length=dim(df[[j]])[1])
  df[[j]]$sd <- numeric(length=dim(df[[j]])[1])
  df[[j]]$mu_class <- numeric(length=dim(df[[j]])[1])
  df[[j]]$sd_class <- numeric(length=dim(df[[j]])[1])
  df[[j]]$mu_y <- numeric(length=dim(df[[j]])[1])
  df[[j]]$sd_y <- numeric(length=dim(df[[j]])[1])
  df[[j]]$mu_class_y <- numeric(length=dim(df[[j]])[1])
  df[[j]]$sd_class_y <- numeric(length=dim(df[[j]])[1])
}
# for loop to add mean, standard deviation, and classification
# to the original data set from the merged data set
for(i in 1:dim(est_merge.dat)[1]){ # begin 1st for loop
  for(j in 1:length(year)){ # begin 2nd for loop
    for(k in 1:dim(df[[j]])[1]){ # begin 3rd for loop
      if(est_merge.dat$n[i]==df[[j]]$Ref_n[k]){ # begin if statement
        df[[j]]$mu[k] <- est_merge.dat$mu[i]
        df[[j]]$sd[k] <- est_merge.dat$sd[i]
```

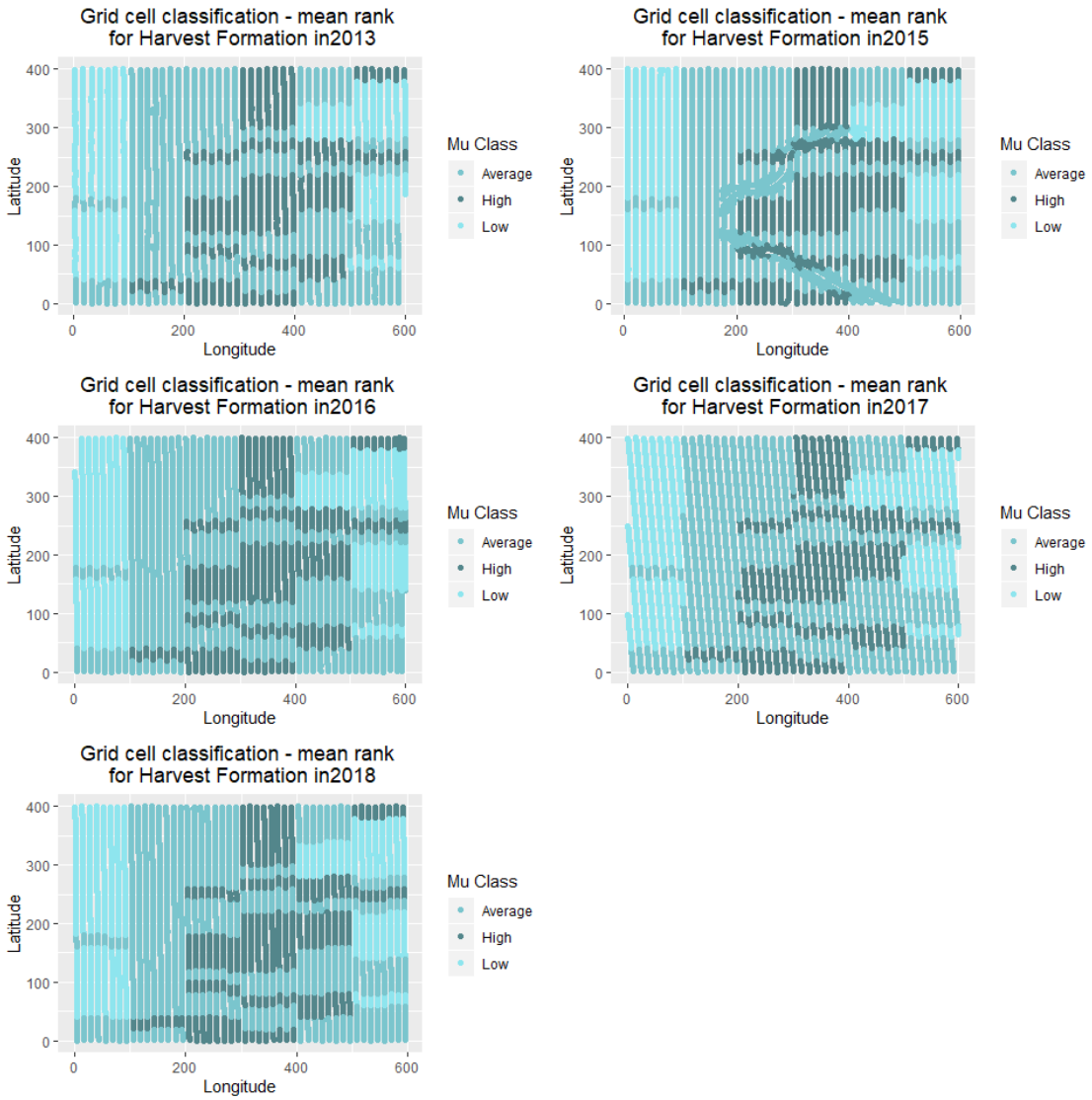
```
        df[[j]]$mu_class[k] <- est_merge.dat$mu_class[i]
        df[[j]]$sd_class[k] <- est_merge.dat$sd_class[i]
      } # end if statement
    } # end 3rd for loop
  } # end 1st for loop
} # end 1st for loop

# for loop to add mean, standard deviation, and classification
# to the original data set from the pre merged data set
for(i in 1:dim(est.dat)[1]){ # begin 1st for loop
  for(j in 1:length(year)){ # begin 2nd for loop
    for(k in 1:dim(df[[j]])[1]){ # begin 3rd for loop
       if(est.dat$n[i] == df[[j]]$Ref_n[k] &&
       est.dat$year[i] == year[j]){ # begin if statement
         df[[j]]$mu_y[k] <- est.dat$Estimates[i]
         df[[j]]$sd_y[k] <- est.dat$obs.sd[i]
         df[[j]]$mu_class_y[k] <- est.dat$mu_class_y[i]
         df[[j]]$sd_class_y[k] <- est.dat$sd_class_y[i]
       } # end if statement
    } # end 3rd for loop
  } # end 2nd for loop
} # end 1st for loop
```

Plotting Merged Data

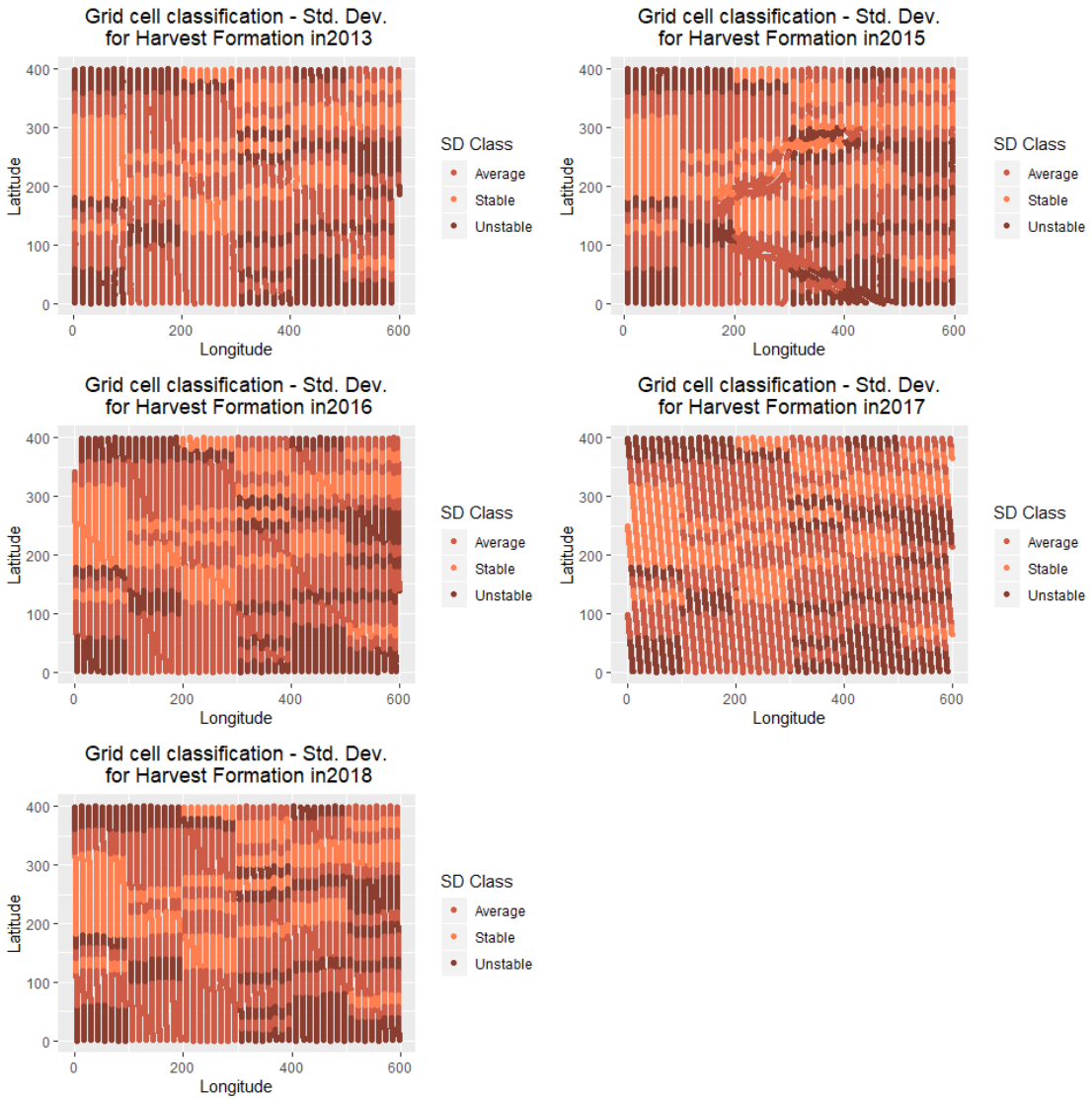
The Yield classes were visually represented by plotting the Longitude as independent variable and Latitude as dependent variable while distinguishing the grid cells with different colors based on their class.

Two similar plots were produced. One graph to illustrate the classification by normalized mean and one based on standard deviation of normalized mean for the merged data across harvest formations for all the years.

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color =
mu_class))+geom_point()
+ scale_color_manual(values = c("cadetblue3", "cadetblue4", "cadetblue2"))
+ ggtitle(paste0('Grid cell classification - mean rank \n for Harvest
Formation in',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='Mu Class'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



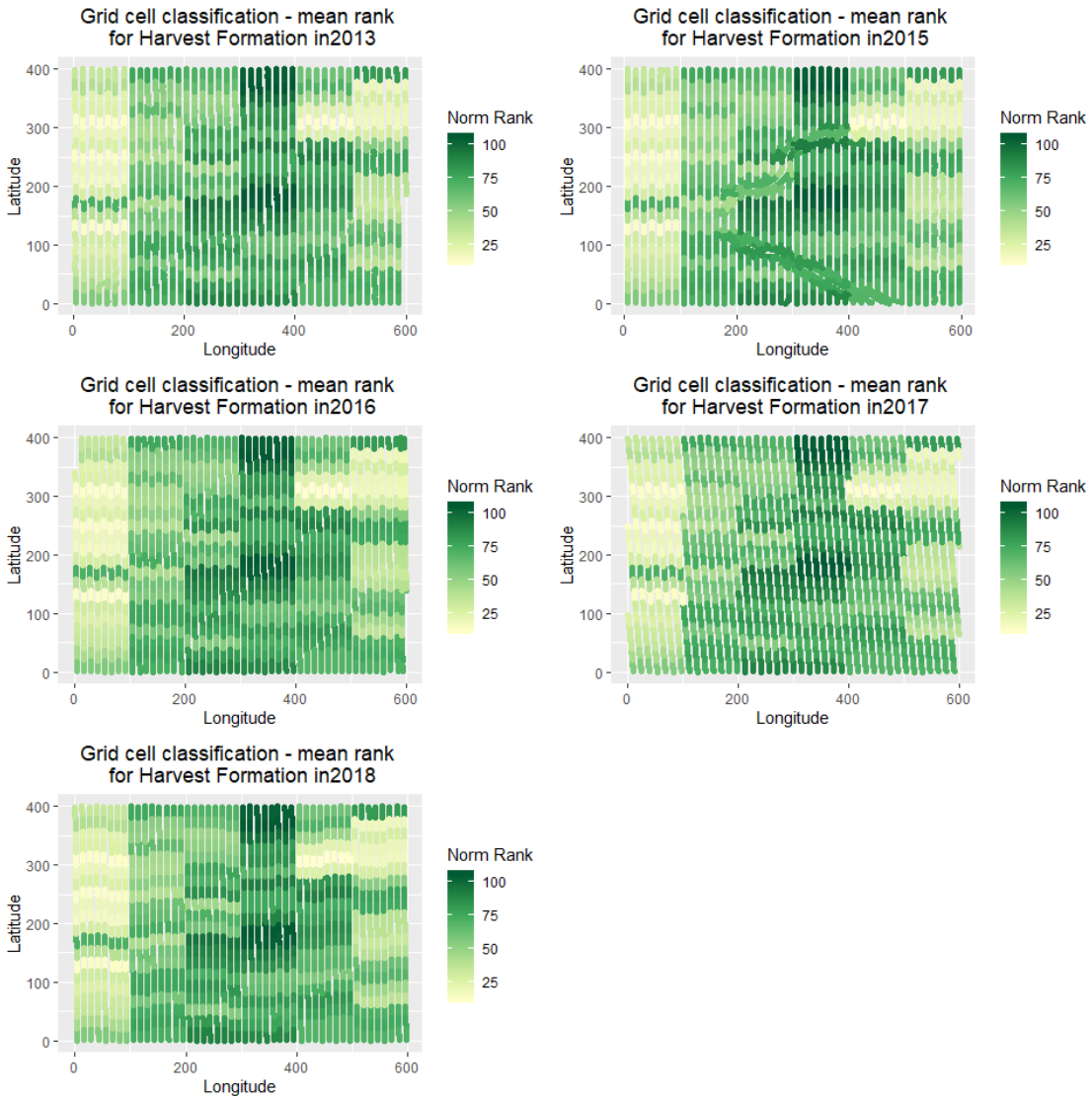

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color = sd_class)) +
geom_point()
+ scale_color_manual(values = c("coral3", "coral", "coral4"))
+ ggtitle(paste0('Grid cell classification - Std. Dev.\n for Harvest
Formation in',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='SD Class'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



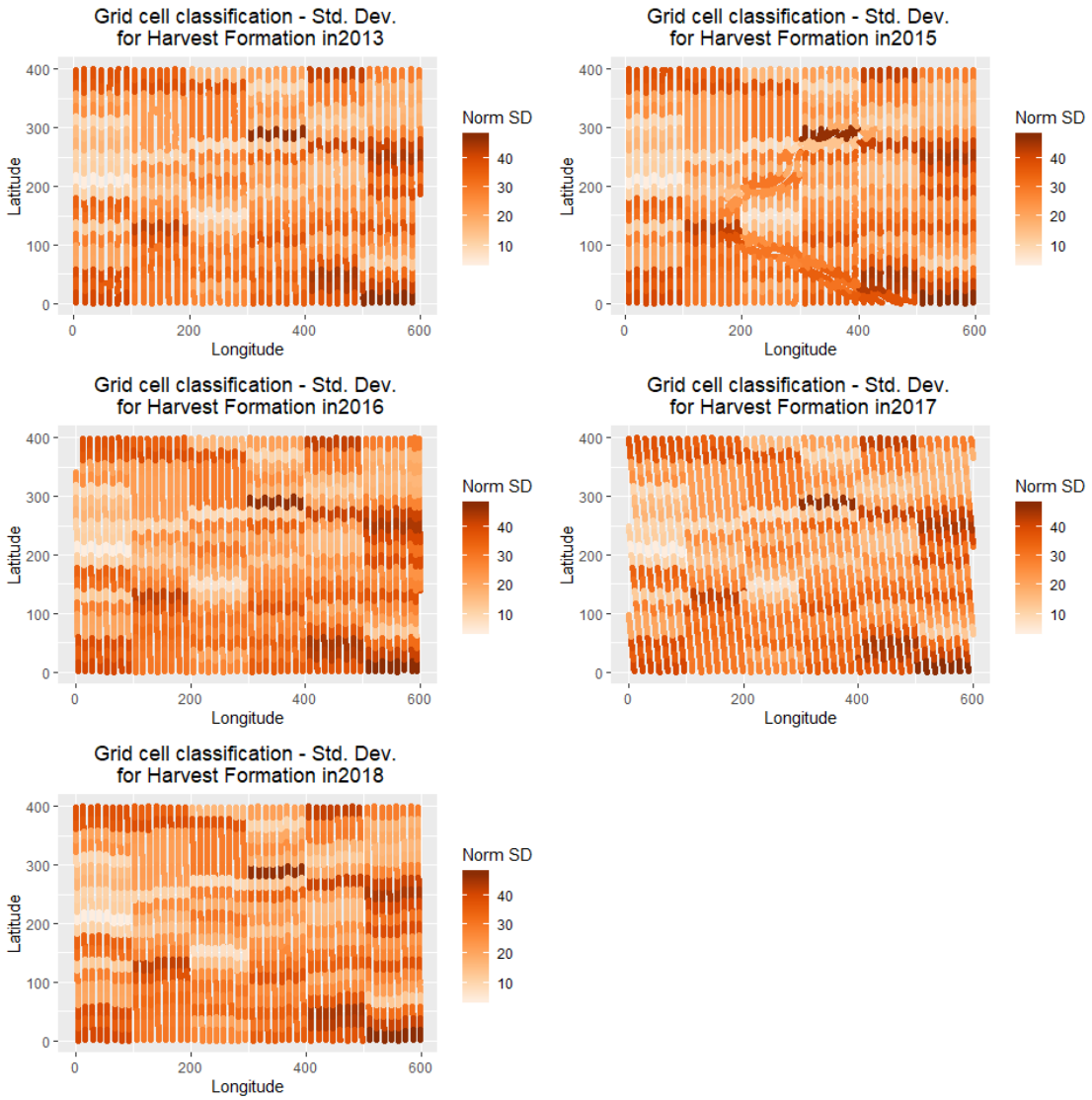
The plots clearly show that not all the areas that are high yielding give a stable yield.

Additionally, graphs were plotted for 120 grids for individual year showing the mean estimates per grid(mean rank) with Longitude as independent variable and Latitude as dependent Variable. Graphed the merged data across harvest formations for all the years.

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color = mu)) +
geom_point()
+ scale_color_distiller(type = 'seq',palette = 'YlGn',direction = 1)
+ ggtitle(paste0('Grid cell classification - mean rank \n for Harvest
Formation in',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='Norm Rank'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



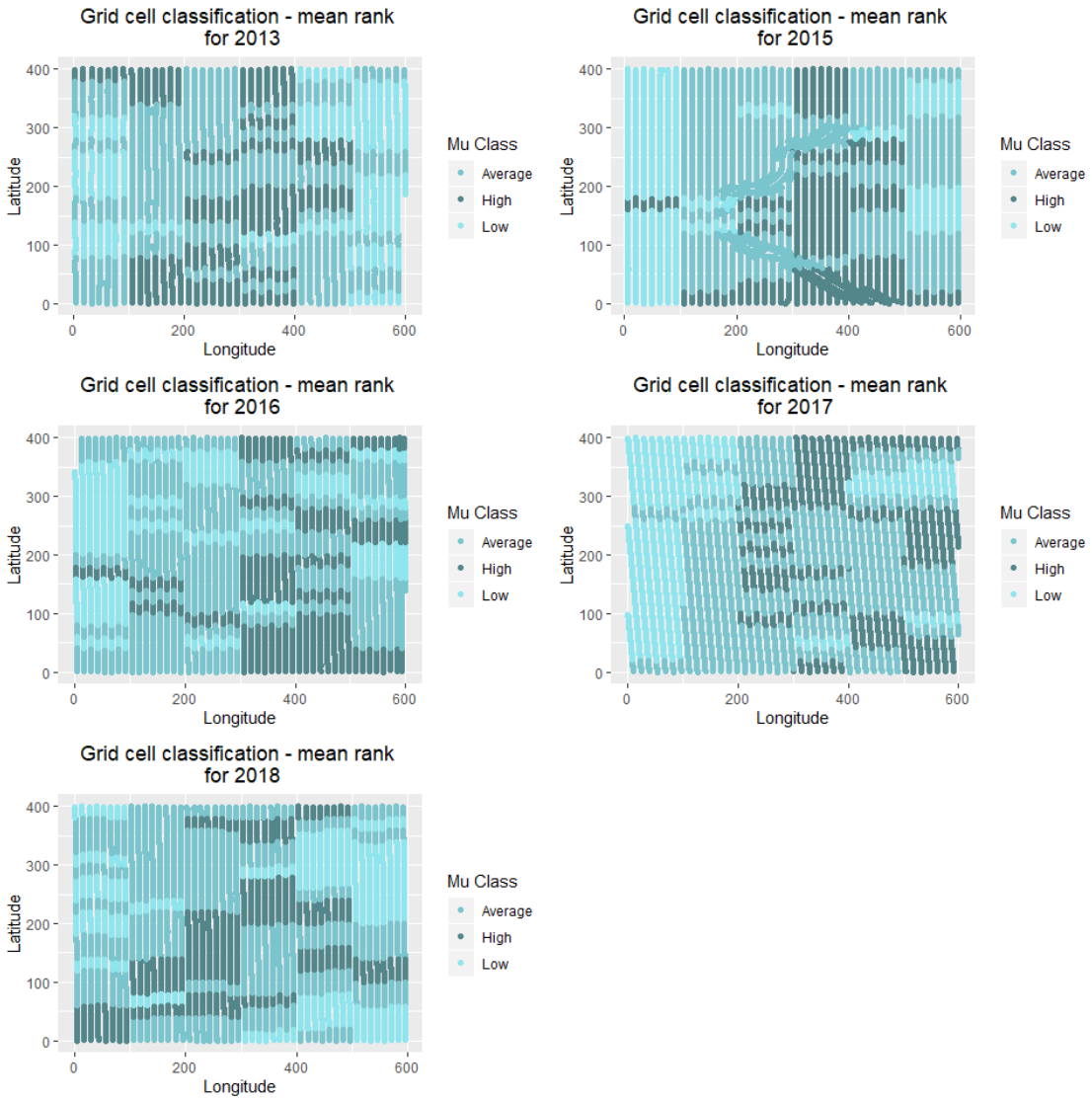

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color = sd)) +
geom_point()
+ scale_color_distiller(type = 'seq',palette = 'Oranges',direction = 1)
+ ggtitle(paste0('Grid cell classification - Std. Dev. \n for Harvest
Formation in',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='Norm SD'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



Plotting Yearly Data

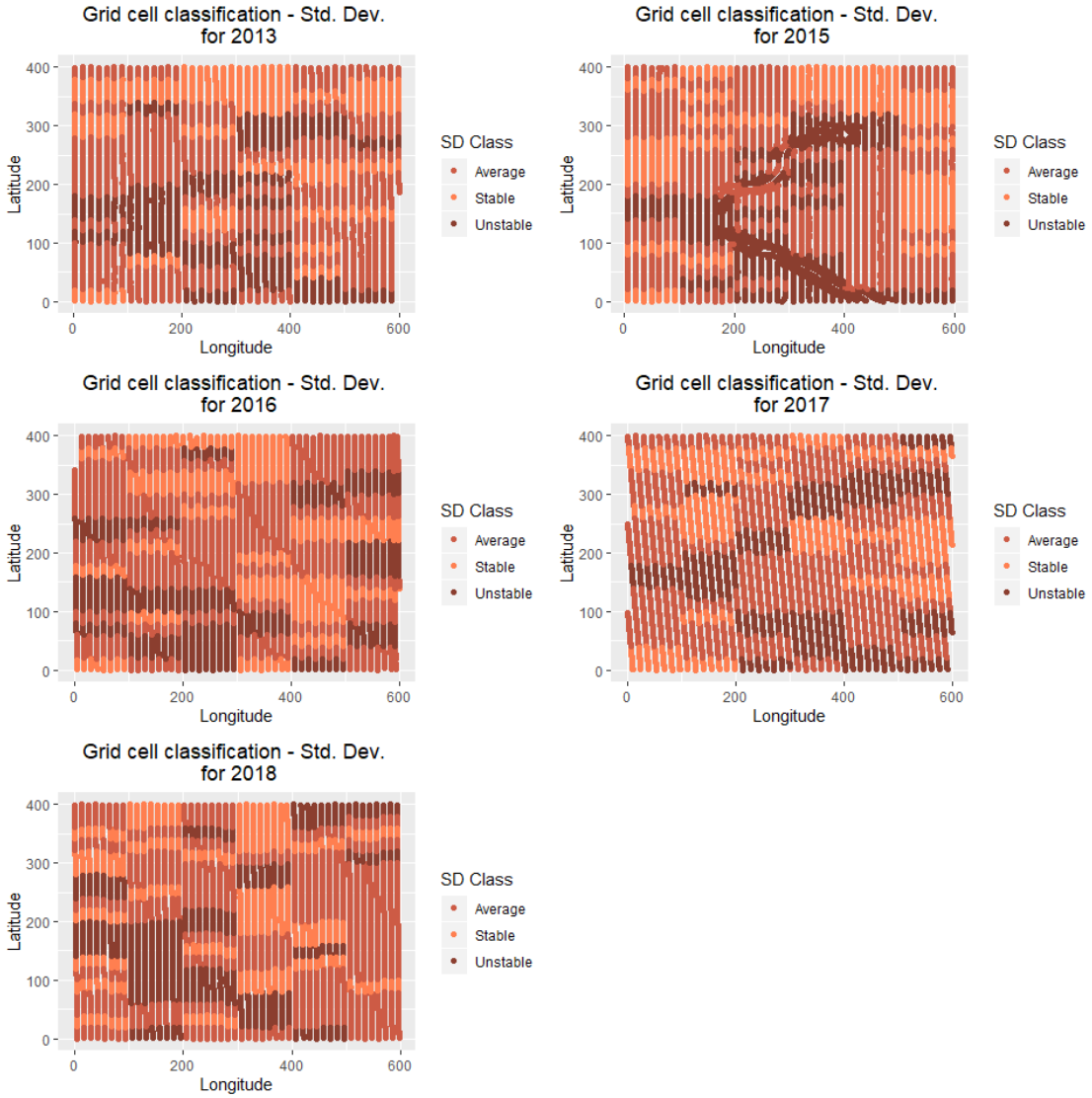
The below plot distinguishes the grids by their Yield classes (High Yielding, Low Yielding, Average Yield) for each year separately. Values are yield means per grid cell.

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color = mu_class_y)) +
geom_point()
+ scale_color_manual(values = c("cadetblue3", "cadetblue4", "cadetblue2"))
+ ggtitle(paste0('Grid cell classification - mean rank \n for ',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='Mu Class'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



The below plot distinguishes the grids by the standard deviation of the estimates per grid (Unstable Yielding, Stable Yielding, Average Yield) for each year separately. Values are standard deviations per grid cell for each of the observations.

```
plot <- list()
for(j in 1:5){
plot[[j]] <- (ggplot(df[[j]],aes(Longitude,Latitude,color = sd_class_y)) +
geom_point()
+ scale_color_manual(values = c("coral3", "coral", "coral4"))
+ ggtitle(paste0('Grid cell classification - Std. Dev. \n for ',year[j]))
+ theme(plot.title = element_text(hjust = 0.5))
+ labs(color='SD Class'))
}
grid.arrange(plot[[1]],plot[[2]],plot[[3]],plot[[4]],plot[[5]],ncol=2)
```



Conclusion

As suggested, the field was divided into 6 columns and 20 rows. We validated the selected grid size by calculating the number of required replicates and CV. The number of observations per grid were more than 30 bushels and the plot of log minimum samples clearly illustrates the same. We screened the data to check if all the crops were harvested within a week.And the plot indicates that all the crops were harvested within a week for all the years.Then we verified the boundaries of the field for all the years.The yield was then normalized by ranking them and mean of yield was calculated per grid.

Finally,mean rank and standard deviation were calculated for the ranks of all years to indicate the performance of the field per grid over 5 years.The field was classified as High,Low and Average Yielding based on the normalized rank and as Stable, Unstable and Average Yielding based on the Standard deviation of the normalized rank. Graphs were

plotted to show the same.Further, additional graphs were plotted for the mean rank for original data for individual years and their yield classes.

Team Contribution

John Herbert's collaboration included the problem statement, coding, comments, summaries, and formulas for Validating grid cell size, plots for RR & CV of grid cells, data screening, Graphical confirmation of Data screening, checking latitude and longitude boundaries, normalization by rank and their graphs.

Snigdha Peddi's collaboration included the coding, comments, summaries, and formulas for merging the data, classification, plotting merged data (mean rank, standard deviation from original data and merged data), wrote the conclusion and team collaboration section.

Both John and Snigdha collaborated equally on editing and finalization of this document.