Final Project

Abdulkadir Said & Mohamed Ahmed

8/9/2020

Project Overview

This is a continuation of the analysis started with the midterm project. We will use the three data sets from the midterm project and two additional data sets for the years 2013 and 2018. Overall, there are 5 data sets that will be used for this analysis. These data sets are from harvesting in the years 2013,2015-2018. We will divide each data set into grid cells, then compute a yield estimate for each cell. The wide format will be used to divide and combine the cells that have been divided into 120 grid cells. In the wide format the rows will show the different cells (120 grid cells). Then there will be a column for the unique identifier (i.e. ID), a column for Yield Estimate from 2013, a column for the Yield Estimates from 2015, a column for the Yield Estimates from 2016, a column for the Yield Estimates from 2017, and a column for the Yield Estimates from 2018. Then We will compute the rank for each year and an overall rank for each grid cell across the years.

I, Abdulkadir Said, wrote an overview of the project, a function to append (col, row, and cells), normalized the data, and plotted the after normalization distribution plots, and plotted classification plots for yield scores.

Mohamed Ahmed, wrote the conclusion of the project, screened the data, wrote a function to divide the grid cells, and plotted before normalization distribution plots, and plotted classification plots for standard deviation.

Data

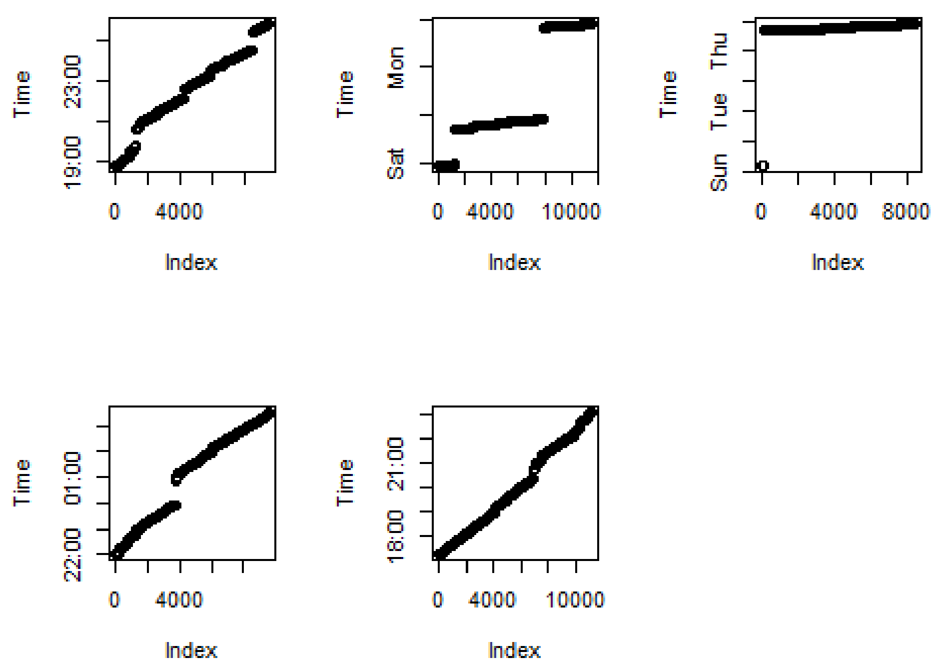
```
home2013 <- read.csv("C:/Users/abdul/OneDrive/Desktop/Kaggle/Stat600/Final Project/home.2013.csv", header=T, sep = ",")
home2013.dat <- data.frame(home2013)

home2015 <- read.csv("C:/Users/abdul/OneDrive/Desktop/Kaggle/Stat600/Final Project/home.2015.csv", header=T, sep = ",")
home2015.dat <- data.frame(home2015)

home2016 <- read.csv("C:/Users/abdul/OneDrive/Desktop/Kaggle/Stat600/Final Project/home.2016.csv", header=T, sep = ",")
home2016.dat <- data.frame(home2016)

home2017 <- read.csv("C:/Users/abdul/OneDrive/Desktop/Kaggle/Stat600/Final Project/home.2017.csv", header=T, sep = ",")
home2017.dat <- data.frame(home2017)
```

```r
home2018 <- read.csv("C:/Users/abdul/OneDrive/Desktop/Kaggle/Stat600/Final Project/home.2018.csv", header=T, sep = ",")
home2018.dat <- data.frame(home2018)
```

Data Screening: Is Harvest less than 7 Days?

one of the conditions we had was to use only data sets with seven days interval or less. We converted Timestamps column from character to Date and Time then we plotted each data set to check if it meets the requirement.

```r
# Converting character data to date and time
Time2013 <- as.POSIXct(home2013$TimeStamp)
Time2015 <- as.POSIXct(home2015$TimeStamp)
Time2016<- as.POSIXct(home2016$TimeStamp)
Time2017 <- as.POSIXct(home2017$TimeStamp)
Time2018 <- as.POSIXct(home2018$TimeStamp)

# plotting to check for 7 day interval in each data set
par(mfrow=c(2,3))
plot(Time2013, ylab = "Time")
plot(Time2015, ylab = "Time")
plot(Time2016, ylab = "Time")
plot(Time2017, ylab = "Time")
plot(Time2018, ylab = "Time")
```



Appending Cell

Identifiers to be used for classification Also, the data was normalized by computing the rank for each year's data. only heading are shown to save space. the view the whole data, use the RMD file.

```r
home.Function <- function(home,yield,latitude,longitude){
  # range of latitude
  minlat <- 0
  maxlat <- max(home$Latitude)
  rangelat <- maxlat-minlat
  # range of longitude
  minlong <- 0
  maxlong <- max(home$Longitude)
  rangelong <- maxlong - minlong

  home$Row <- ceiling(20*home$Latitude/rangelat)
  home$Col <- ceiling(6*home$Longitude/rangelong)
  home$Cell <- (home$Row*1000 + home$Col)
  home$rank <- rank(home$Yield)

  return(home)

}

# The first 6 rows of the appended data
home2013 <- home.Function(home=home2013, yield = home2013$Yield, latitude =
home2013$Latitude, longitude = home2013$Longitude)
home2015 <- home.Function(home=home2015, yield = home2015$Yield, latitude =
home2015$Latitude, longitude = home2015$Longitude)
home2016 <- home.Function(home=home2016, yield = home2016$Yield, latitude =
home2016$Latitude, longitude = home2016$Longitude)
home2017 <- home.Function(home=home2017, yield = home2017$Yield, latitude =
home2017$Latitude, longitude = home2017$Longitude)
home2018 <- home.Function(home=home2018, yield = home2018$Yield, latitude =
home2018$Latitude, longitude = home2018$Longitude)

head(home2013,5)
```

```
##      Yield Latitude Longitude           TimeStamp Row Col  Cell   rank
## 1 43.57736 399.4129  3.274650 2013-09-30 18:47:00  20   1 20001 6495.5
## 2 47.40136 397.4667  3.257958 2013-09-30 18:47:01  20   1 20001 8070.5
## 3 46.38477 395.5015  3.216063 2013-09-30 18:47:02  20   1 20001 7701.0
## 4 49.19995 393.5342  3.257626 2013-09-30 18:47:03  20   1 20001 8556.0
## 5 42.86166 391.6324  3.258953 2013-09-30 18:47:04  20   1 20001 6142.0
```

```r
head(home2015,5)
```

```
##      Yield  Latitude Longitude           TimeStamp Row Col Cell rank
## 1 40.85889 0.8848444  120.8712 2015-07-17 22:36:24   1   2 1002 8074
## 2 43.54383 2.1551468  120.8833 2015-07-17 22:36:25   1   2 1002 8955
## 3 42.33718 3.4424795  120.8776 2015-07-17 22:36:26   1   2 1002 8543
```

```
## 4 39.21862 4.7188642  120.8685 2015-07-17 22:36:27   1   2 1002 7493
## 5 38.24887 6.0019946  120.8820 2015-07-17 22:36:28   1   2 1002 7135
```

```r
head(home2016,5)
```

```
##       Yield Latitude Longitude           TimeStamp Row Col Cell rank
## 1  93.43079 139.8663  599.9697 2016-10-30 04:04:22   7   6 7006 1739
## 2  85.54778 143.7336  599.9374 2016-10-30 04:04:24   8   6 8006 1352
## 3  91.67040 146.6649  599.9179 2016-10-30 04:04:26   8   6 8006 1641
## 4 101.74151 149.5970  599.9059 2016-10-30 04:04:27   8   6 8006 2232
## 5 111.19036 152.5513  599.8653 2016-10-30 04:04:28   8   6 8006 2921
```

```r
head(home2017,5)
```

```
##      Yield Latitude Longitude           TimeStamp Row Col Cell rank
## 1 58.69077 1.391954  256.8762 2017-10-10 22:01:07   1   3 1003 4715
## 2 58.25433 3.152152  256.6823 2017-10-10 22:01:08   1   3 1003 4312
## 3 65.81719 4.796789  256.4935 2017-10-10 22:01:09   1   3 1003 9075
## 4 61.12727 6.444853  256.3046 2017-10-10 22:01:10   1   3 1003 6902
## 5 58.82723 8.111275  256.1368 2017-10-10 22:01:11   1   3 1003 4849
```

```r
head(home2018,5)
```

```
##      Yield Latitude Longitude           TimeStamp Row Col  Cell rank
## 1 249.6109 399.2460  264.9795 2018-11-02 17:15:40  20   3 20003 7642
## 2 257.6665 397.4478  265.0273 2018-11-02 17:15:41  20   3 20003 9515
## 3 259.5920 395.7073  265.0585 2018-11-02 17:15:42  20   3 20003 9815
## 4 253.4787 393.9365  265.0588 2018-11-02 17:15:43  20   3 20003 8658
## 5 243.9952 392.1895  265.0903 2018-11-02 17:15:44  20   3 20003 5959
```

Cell Divisions

A function was defined to divide the data into grid cells

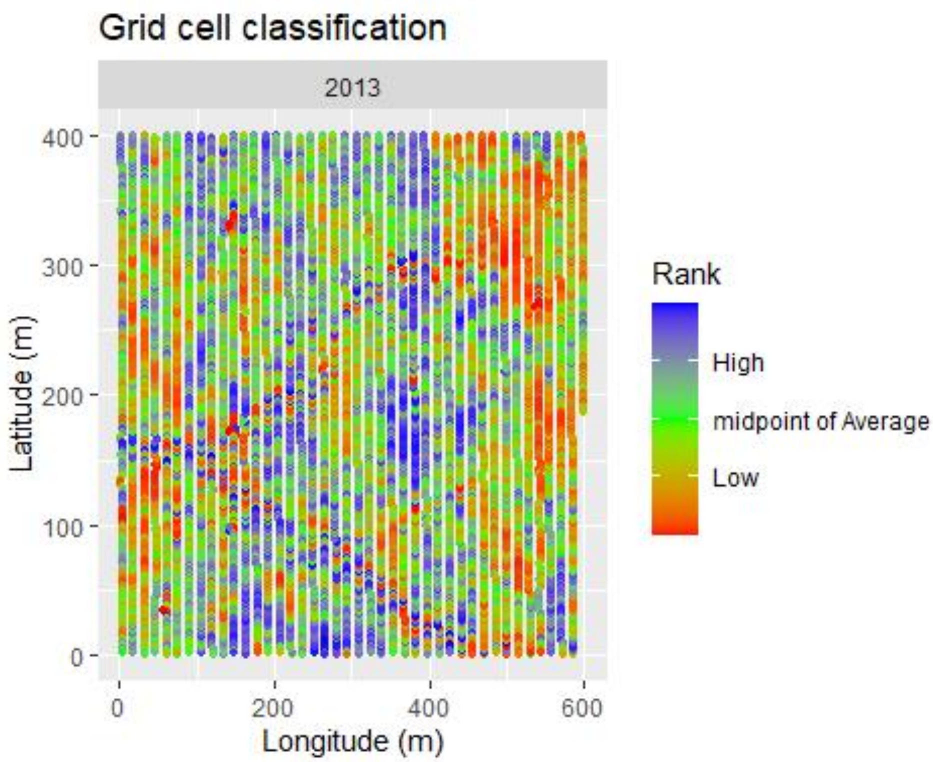
```r
Cell.divisions <- function(homeYear,yield, longitude, latitude){
  # range of latitude
  minlat <- 0
  maxlat <- max(latitude)
  rangelat <- maxlat-minlat
  #range of longitude
  minlong <- 0
  maxlong <- max(longitude)
  rangelong <- maxlong - minlong
  home.divisions <- data.frame(Divisions=1)
  home.divisions$MinYield=NA
  home.divisions$MaxYield=NA
  home.divisions$Gridcellnumber=NA
  home.divisions$mean=NA
  home.divisions$sd=NA

  for (i in 1:length(home.divisions$Divisions)){
```

```
    required.replicates <- function (cv, diff, alpha = 0.05, beta=0.2) {
    alpha <- 0.05
    beta <- 0.2
    z_alpha <- (qnorm(1-alpha/2))
    z_beta  <- (qnorm(1-beta))
    n <-  round(2*((cv/diff)^2)*((z_alpha + z_beta)^2),0)
    return(n)
  }
    div <- i
    homeYear$Row <- ceiling(20*div*latitude/rangelat)
    homeYear$Col <- ceiling(6*div*longitude/rangelong)
    homeYear$Cell <- homeYear$Row*1000 + homeYear$Col
    yield <- tapply(homeYear$Cell,homeYear$Cell,length)
    means <- tapply(homeYear$Yield,homeYear$Cell,mean)

    home.divisions$Gridcellnumber[i] <- length(means)
    home.divisions$MinYield[i] <- min(yield)
    home.divisions$MaxYield[i] <- max(yield)
    home.divisions$mean[i] <- mean(means)
    home.divisions$sd[i] <- sd(means)
    home.divisions$cv[i] <- (100*home.divisions$sd[i]/home.divisions$mean[i])
    home.divisions$RR2.5 <- (required.replicates(cv=home.divisions$cv, diff =
2.5))
    home.divisions$RR5 <- (required.replicates(cv=home.divisions$cv, diff =
5))
    home.divisions$RR10 <- (required.replicates(cv=home.divisions$cv, diff =
10))
  }
  return(home.divisions)
}
Cell.divisions(home2013,yield= home2013$Yield, longitude =
home2013$Longitude, latitude = home2013$Latitude)

##   Divisions MinYield MaxYield Gridcellnumber     mean       sd       cv
RR2.5
## 1         1       67       93            120 40.46977 3.340354 8.253949
171
##   RR5 RR10
## 1  43   11

Cell.divisions(home2015,yield= home2015$Yield, longitude =
home2015$Longitude, latitude = home2015$Latitude)

##   Divisions MinYield MaxYield Gridcellnumber     mean       sd       cv
RR2.5
## 1         1       87      131            120 35.78163 6.393858 17.86911
802
##   RR5 RR10
## 1 200   50
```

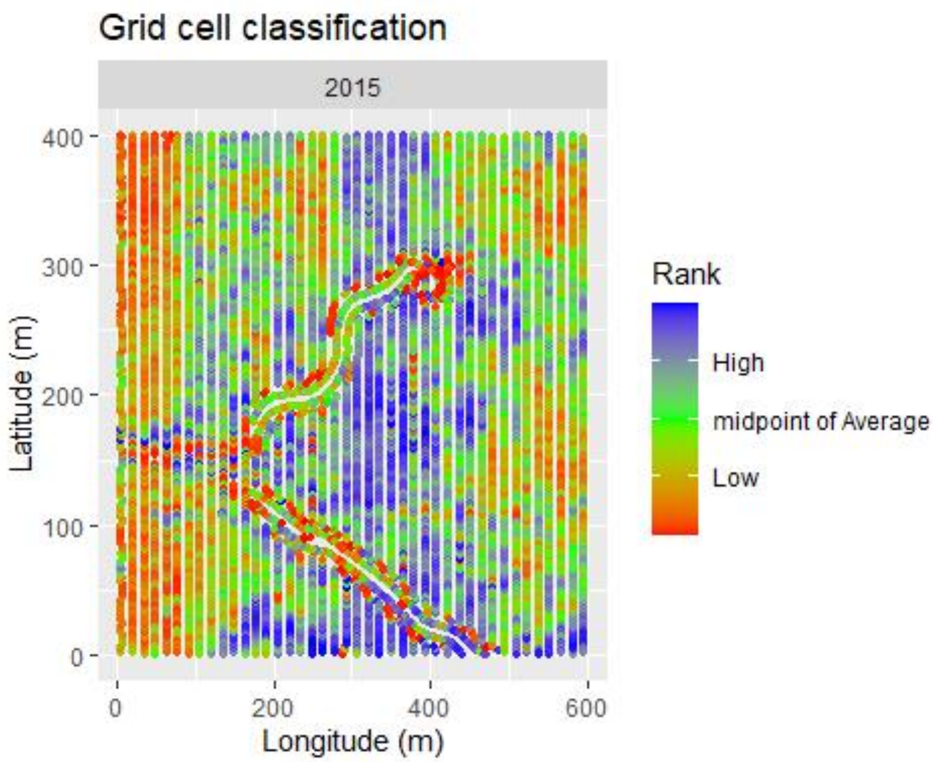
```
Cell.divisions(home2016,yield= home2016$Yield, longitude =
home2016$Longitude, latitude = home2016$Latitude)

##   Divisions MinYield MaxYield Gridcellnumber     mean       sd       cv
RR2.5
## 1         1       58       91            120 117.6105 17.55513 14.92649
560
##   RR5 RR10
## 1 140   35

Cell.divisions(home2017,yield= home2017$Yield, longitude =
home2017$Longitude, latitude = home2017$Latitude)

##   Divisions MinYield MaxYield Gridcellnumber    mean       sd       cv
RR2.5
## 1         1       68       96            120 58.4982 1.619086 2.767754
19
##   RR5 RR10
## 1   5    1

Cell.divisions(home2018,yield= home2018$Yield, longitude =
home2018$Longitude, latitude = home2018$Latitude)

##   Divisions MinYield MaxYield Gridcellnumber     mean       sd       cv
RR2.5
## 1         1       86      134            120 242.6754 6.214401 2.560787
16
##   RR5 RR10
## 1   4    1
```

Ranking the Means

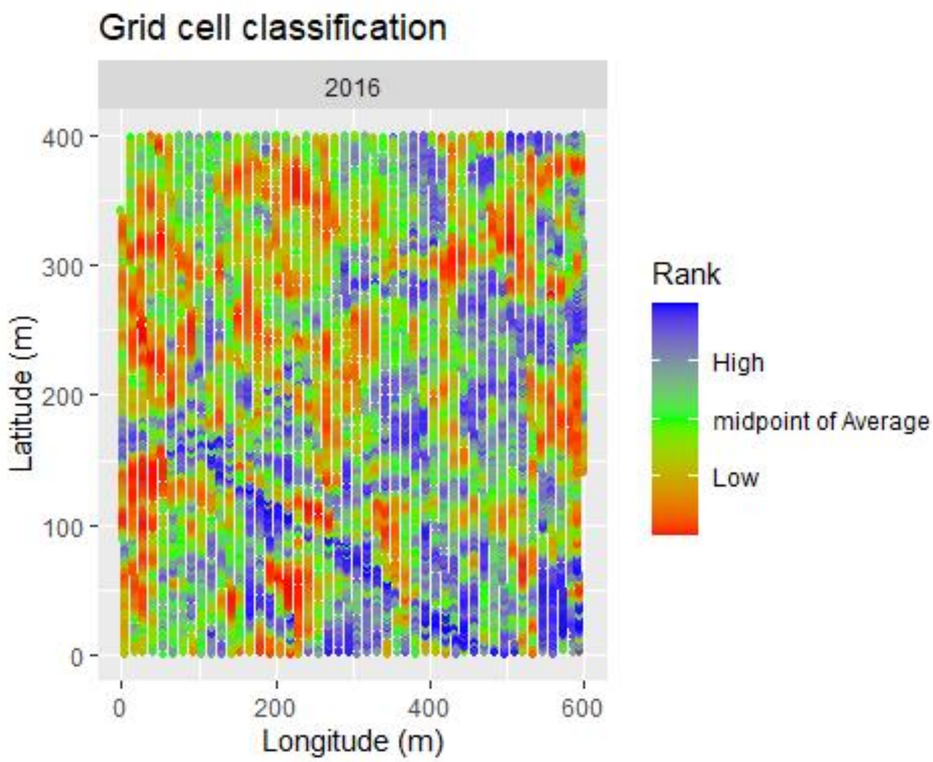
```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.2

ggplot(data = home2013, mapping = aes(x = Longitude, y = Latitude))+
geom_point(aes(color = rank), size = 0.9)+
scale_colour_gradientn(colours = rainbow(3), breaks = c(2376,4750,7124),
labels = c("Low", "midpoint of Average", "High"))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2013) + ggtitle("Grid cell classification")
```



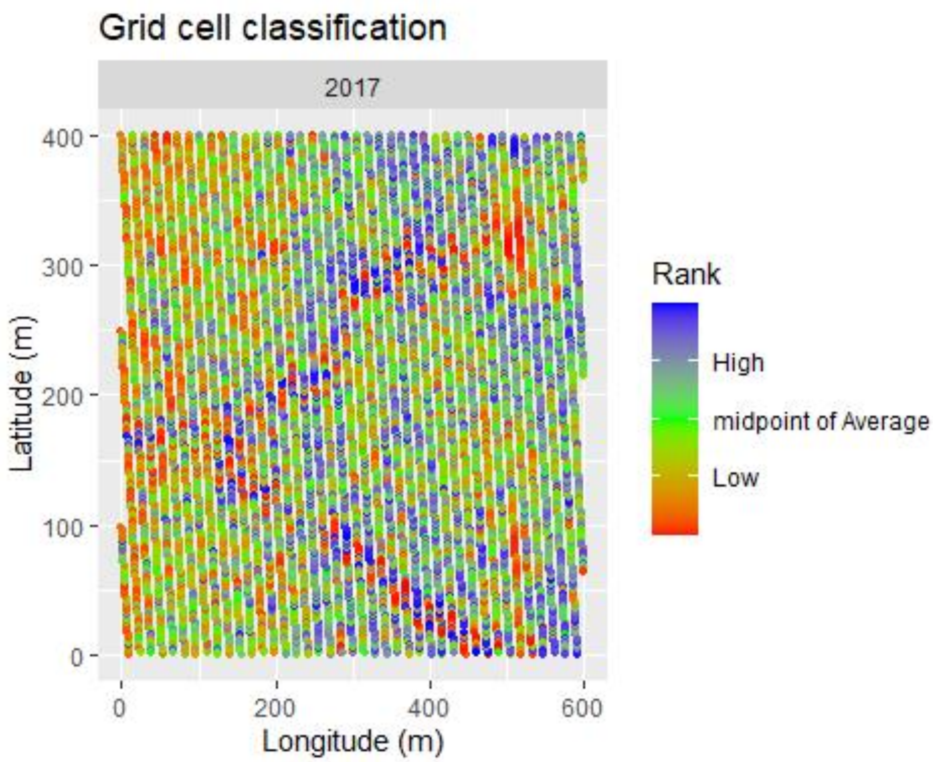

```r
ggplot(data = home2015, mapping = aes(x = Longitude, y = Latitude))+
geom_point(aes(color = rank), size = 0.9)+
scale_colour_gradientn(colours = rainbow(3), breaks = c(2898,5796,8694),
labels = c("Low", "midpoint of Average", "High"))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2015) + ggtitle("Grid cell classification")
```



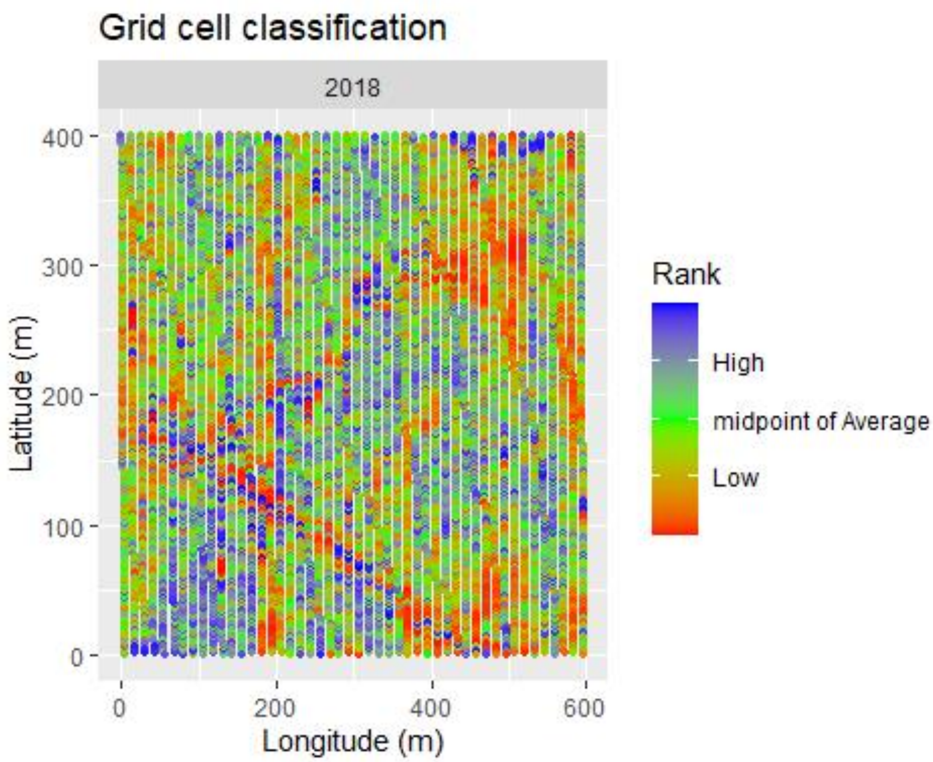

```
ggplot(data = home2016, mapping = aes(x = Longitude, y = Latitude))+
geom_point(aes(color = rank), size = 0.9)+
scale_colour_gradientn(colours = rainbow(3), breaks = c(2104,4207,6310),
labels = c("Low", "midpoint of Average", "High"))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2016) + ggtitle("Grid cell classification")
```




```
ggplot(data = home2017, mapping = aes(x = Longitude, y = Latitude))+
geom_point(aes(color = rank), size = 0.9)+
scale_colour_gradientn(colours = rainbow(3), breaks = c(2396 ,4789,7184),
labels = c("Low", "midpoint of Average", "High"))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2017) + ggtitle("Grid cell classification")
```




```
ggplot(data = home2018, mapping = aes(x = Longitude, y = Latitude))+
geom_point(aes(color = rank), size = 0.9)+
scale_colour_gradientn(colours = rainbow(3), breaks = c(2796,5592,8388),
labels = c("Low", "midpoint of Average", "High"))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2018) + ggtitle("Grid cell classification")
```



Aggregating Yield Estimates & Normalized Computations

Here we have 6 columns as shown in the below code. Here we are computing the means for each of 120 Grid Cells.

```
GrandMeanRows <- data.frame(
  CellNumber=1:120,
  YieldEstimate2013 = tapply(home2013$Yield,home2013$Cell,mean),
  YieldEstimate2015 = tapply(home2015$Yield,home2015$Cell,mean),
  YieldEstimate2016 = tapply(home2016$Yield,home2016$Cell,mean),
  YieldEstimate2017 = tapply(home2017$Yield,home2017$Cell,mean),
  YieldEstimate2018 = tapply(home2018$Yield,home2018$Cell,mean))
head(GrandMeanRows)

##      CellNumber YieldEstimate2013 YieldEstimate2015 YieldEstimate2016
## 1001          1          40.78269          26.21633          117.0053
## 1002          2          44.30403          42.19282          112.6399
## 1003          3          49.37674          51.13233          127.2392
## 1004          4          43.24956          46.71617          135.2822
## 1005          5          41.30299          41.58601          152.2427
## 1006          6          37.92880          47.29986          134.5051
##      YieldEstimate2017 YieldEstimate2018
## 1001          57.68473          254.9570
## 1002          58.18460          242.5142
## 1003          58.73003          244.7709
## 1004          60.05934          237.5016
```

```
## 1005          57.64959          238.7490
## 1006          60.16846          234.8132
```

After aggregating the Yield Estimates, below we are going to add a normalized Latitude (down 120 rows), a normalized Longitude, and a normalized standard deviation. The data below has total of 120 rows. For the purpose of saving space, we are going to only show the first 5 rows. If you want to view the whole data, please go to the RMD file and remove the head() function.

```
# All Five Data Sets Combined with the unique Identifiers
harvestAgg <- data.frame(
  Cell = 1:120,
  LatitudeAGG2013 = tapply(home2013$Latitude, home2013$Cell, mean),
  LongitudeAGG2013 = tapply(home2013$Longitude, home2013$Cell, mean),
  YieldEstimate2013 = tapply(home2013$Yield,home2013$Cell,mean),
  SDYield2013 = tapply(home2013$Yield,home2013$Cell,sd),
  LatitudeAGG2015 = tapply(home2015$Latitude, home2015$Cell, mean),
  LongitudeAGG2015 = tapply(home2015$Longitude, home2015$Cell, mean),
  YieldEstimate2015 = tapply(home2015$Yield,home2015$Cell,mean),
  SDYield2015 = tapply(home2015$Yield,home2015$Cell,sd),
  LatitudeAGG2016 = tapply(home2016$Latitude, home2016$Cell, mean),
  LongitudeAGG2016 = tapply(home2016$Longitude, home2016$Cell, mean),
  YieldEstimate2016 = tapply(home2016$Yield,home2016$Cell,mean),
  SDYield2016 = tapply(home2016$Yield,home2016$Cell,sd),
  LatitudeAGG2017 = tapply(home2017$Latitude, home2017$Cell, mean),
  LongitudeAGG2017 = tapply(home2017$Longitude, home2017$Cell, mean),
  YieldEstimate2017 = tapply(home2017$Yield,home2017$Cell,mean),
  SDYield2017 = tapply(home2017$Yield,home2017$Cell,sd),
  LatitudeAGG2018 = tapply(home2018$Latitude, home2018$Cell, mean),
  LongitudeAGG2018 = tapply(home2018$Longitude, home2018$Cell, mean),
  YieldEstimate2018 = tapply(home2018$Yield,home2018$Cell,mean),
  SDYield2018 = tapply(home2018$Yield,home2018$Cell,sd),
  RowYieldEstimateMean = rowMeans(GrandMeanRows[,-1])
)

head(harvestAgg,5)

##      Cell LatitudeAGG2013 LongitudeAGG2013 YieldEstimate2013 SDYield2013
## 1001    1        9.932726         47.01996          40.78269    3.493918
## 1002    2       10.286210        149.41001          44.30403    6.154753
## 1003    3        9.750030        251.94380          49.37674    8.346921
## 1004    4       10.026206        355.13635          43.24956    7.103325
## 1005    5        9.966944        448.33362          41.30299   11.859931
##      LatitudeAGG2015 LongitudeAGG2015 YieldEstimate2015 SDYield2015
## 1001        9.951777         48.90171          26.21633    4.567857
## 1002        9.668017        147.84048          42.19282    8.804372
## 1003       10.032981        251.07624          51.13233   14.848197
## 1004       10.283762        350.63241          46.71617   11.972626
## 1005       10.088728        449.58727          41.58601   13.797646
##      LatitudeAGG2016 LongitudeAGG2016 YieldEstimate2016 SDYield2016
```

```
## 1001        9.999348         46.67186          117.0053    13.16346
## 1002       10.026689        147.97435          112.6399    31.44776
## 1003        9.747628        242.70399          127.2392    37.81029
## 1004        9.936912        352.16938          135.2822    19.29458
## 1005       10.310580        449.67783          152.2427    34.80214
##      LatitudeAGG2017 LongitudeAGG2017 YieldEstimate2017 SDYield2017
## 1001        9.942865         52.65224          57.68473    3.150681
## 1002       10.035862        153.36579          58.18460    2.867638
## 1003       10.020572        253.72792          58.73003    5.537253
## 1004        9.825151        347.79622          60.05934    3.489122
## 1005        9.707292        446.19098          57.64959   12.162859
##      LatitudeAGG2018 LongitudeAGG2018 YieldEstimate2018 SDYield2018
## 1001        9.970083         48.56268          254.9570    15.84413
## 1002        9.976924        149.67395          242.5142    20.63998
## 1003        9.933193        249.54600          244.7709    14.60349
## 1004       10.043223        351.67328          237.5016    17.12370
## 1005       10.135355        451.94759          238.7490    38.05492
##      RowYieldEstimateMean
## 1001             99.32921
## 1002             99.96710
## 1003            106.24983
## 1004            104.56177
## 1005            106.30607
```

By aggregating the data, we are able to compute the rank of the Yield Estimates for each year as well as across the five years. That allows us to see the evolution of the Yield Estimates across the five year's data. also we are producing a ColumnHarvest.dat data frame which shows the overall yield estimate (i.e. Grand Mean for that year) for the years 2013 and 2015-2018. We are also producing a yearly standard deviation as well.

```
library("matrixStats")

## Warning: package 'matrixStats' was built under R version 4.0.2

RowHarvest.dat <- data.frame(
  CellNumber=harvestAgg[,1],
  LatitudeAGG2013 = tapply(home2013$Latitude, home2013$Cell, mean),
  LongitudeAGG2013 = tapply(home2013$Longitude, home2013$Cell, mean),
  YieldEstimate2013 = harvestAgg$YieldEstimate2013,
  rank2013 = rank(GrandMeanRows$YieldEstimate2013),
  LatitudeAGG2015 = tapply(home2015$Latitude, home2015$Cell, mean),
  LongitudeAGG2015 = tapply(home2015$Longitude, home2015$Cell, mean),
  YieldEstimate2015 = harvestAgg$YieldEstimate2015,
  rank2015 = rank(GrandMeanRows$YieldEstimate2015),
  LatitudeAGG2016 = tapply(home2016$Latitude, home2016$Cell, mean),
  LongitudeAGG2016 = tapply(home2016$Longitude, home2016$Cell, mean),
  YieldEstimate2016 = harvestAgg$YieldEstimate2016,
  rank2016 = rank(GrandMeanRows$YieldEstimate2016),
  LatitudeAGG2017 = tapply(home2017$Latitude, home2017$Cell, mean),
  LongitudeAGG2017 = tapply(home2017$Longitude, home2017$Cell, mean),
```

```
  YieldEstimate2017 = harvestAgg$YieldEstimate2017,
  rank2017 = rank(GrandMeanRows$YieldEstimate2017),
  LatitudeAGG2018 = tapply(home2018$Latitude, home2018$Cell, mean),
  LongitudeAGG2018 = tapply(home2018$Longitude, home2018$Cell, mean),
  YieldEstimate2018 = harvestAgg$YieldEstimate2018,
  rank2018 = rank(GrandMeanRows$YieldEstimate2018),
  RowYieldEstimateMean = harvestAgg$RowYieldEstimateMean,
  RowYieldSD = rowSds(GrandMeanRows[,-1], center =
harvestAgg$RowYieldEstimateMean),
  RowRank = rank(harvestAgg$RowYieldEstimateMean)

  )
head(RowHarvest.dat,5)

##      CellNumber LatitudeAGG2013 LongitudeAGG2013 YieldEstimate2013
rank2013
## 1001          1        9.932726         47.01996          40.78269
55
## 1002          2       10.286210        149.41001          44.30403
109
## 1003          3        9.750030        251.94380          49.37674
120
## 1004          4       10.026206        355.13635          43.24956
100
## 1005          5        9.966944        448.33362          41.30299
64
##      LatitudeAGG2015 LongitudeAGG2015 YieldEstimate2015 rank2015
## 1001        9.951777         48.90171          26.21633        8
## 1002        9.668017        147.84048          42.19282      102
## 1003       10.032981        251.07624          51.13233      119
## 1004       10.283762        350.63241          46.71617      113
## 1005       10.088728        449.58727          41.58601       96
##      LatitudeAGG2016 LongitudeAGG2016 YieldEstimate2016 rank2016
## 1001        9.999348         46.67186          117.0053       60
## 1002       10.026689        147.97435          112.6399       48
## 1003        9.747628        242.70399          127.2392       79
## 1004        9.936912        352.16938          135.2822      103
## 1005       10.310580        449.67783          152.2427      118
##      LatitudeAGG2017 LongitudeAGG2017 YieldEstimate2017 rank2017
## 1001        9.942865         52.65224          57.68473       39
## 1002       10.035862        153.36579          58.18460       48
## 1003       10.020572        253.72792          58.73003       64
## 1004        9.825151        347.79622          60.05934       98
## 1005        9.707292        446.19098          57.64959       36
##      LatitudeAGG2018 LongitudeAGG2018 YieldEstimate2018 rank2018
## 1001        9.970083         48.56268          254.9570      118
## 1002        9.976924        149.67395          242.5142       57
## 1003        9.933193        249.54600          244.7709       70
## 1004       10.043223        351.67328          237.5016       25
## 1005       10.135355        451.94759          238.7490       32
```

```
##      RowYieldEstimateMean RowYieldSD RowRank
## 1001             99.32921   93.59474      57
## 1002             99.96710   84.64970      60
## 1003            106.24983   83.90358     114
## 1004            104.56177   83.22036     104
## 1005            106.30607   87.22506     116
```

```
# Column means and Ranks
ColumnHarvest.dat <- data.frame(
  YieldEstimate2013 = mean(harvestAgg$YieldEstimate2013),
  SD2013 = sd(harvestAgg$YieldEstimate2013),
  YieldEstimate2015 = mean(harvestAgg$YieldEstimate2015),
  SD2015 = sd(harvestAgg$YieldEstimate2015),
  YieldEstimate2016 = mean(harvestAgg$YieldEstimate2016),
  SD2016 = sd(harvestAgg$YieldEstimate2016),
  YieldEstimate2017 = mean(harvestAgg$YieldEstimate2017),
  SD2017 = sd(harvestAgg$YieldEstimate2017),
  YieldEstimate2018 = mean(harvestAgg$YieldEstimate2018),
  SD2018 = sd(harvestAgg$YieldEstimate2018)
)
ColumnHarvest.dat
```

```
##   YieldEstimate2013   SD2013 YieldEstimate2015   SD2015 YieldEstimate2016
## 1          40.46977 3.340354          35.78163 6.393858          117.6105
##     SD2016 YieldEstimate2017   SD2017 YieldEstimate2018   SD2018
## 1 17.55513           58.4982 1.619086          242.6754 6.214401
```

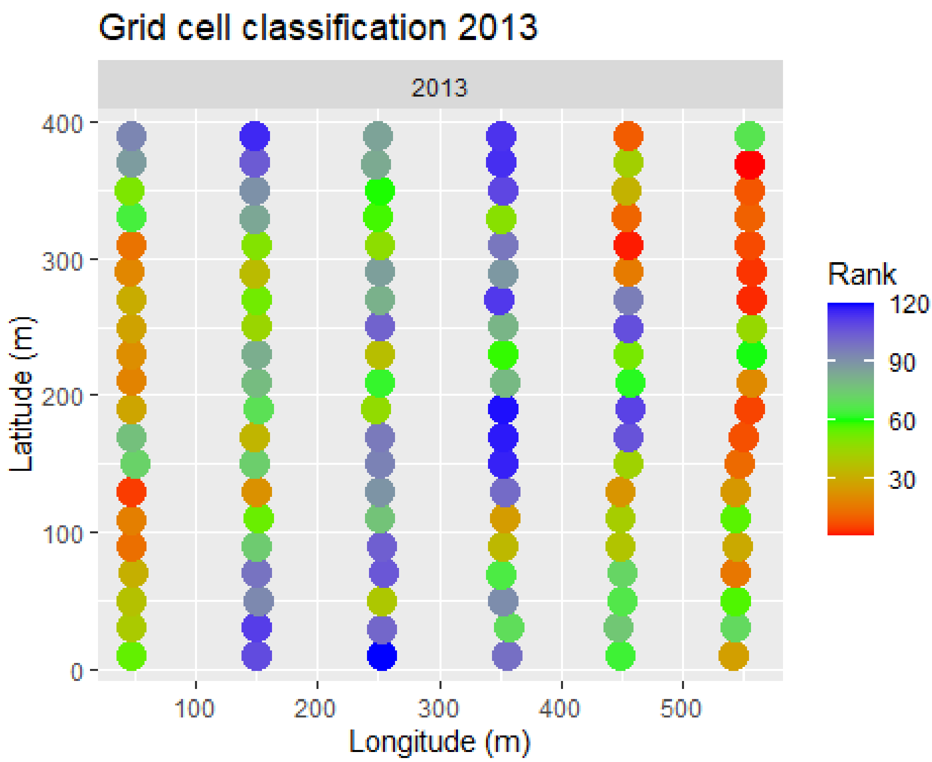
Classification According to Normalized Ranks (Estimated Yield Scores)

Below we are producing a classification plot of the normalized ranks. we have chosen the normalized longitude to be the independent variable axis and the normalized latitude to be the dependent variable axis. Then we are going to plot the ranks and classify in terms of color as high, low, or medium according to their rank. The highest possible rank is going to be 120. with 90-120 being the 3rd Quantile or the highest 25% of the ranks. hence, any rank above 90 is going to be considered as ‘high rank’ or the darker blue of the scale. The lowest possible rank is going to be 1 and any rank below 30 is going to be classified as ‘low rank’ which basically corresponds to be 1st Quantile. Any rank between 30 and 90 is going to be classified as average or medium.

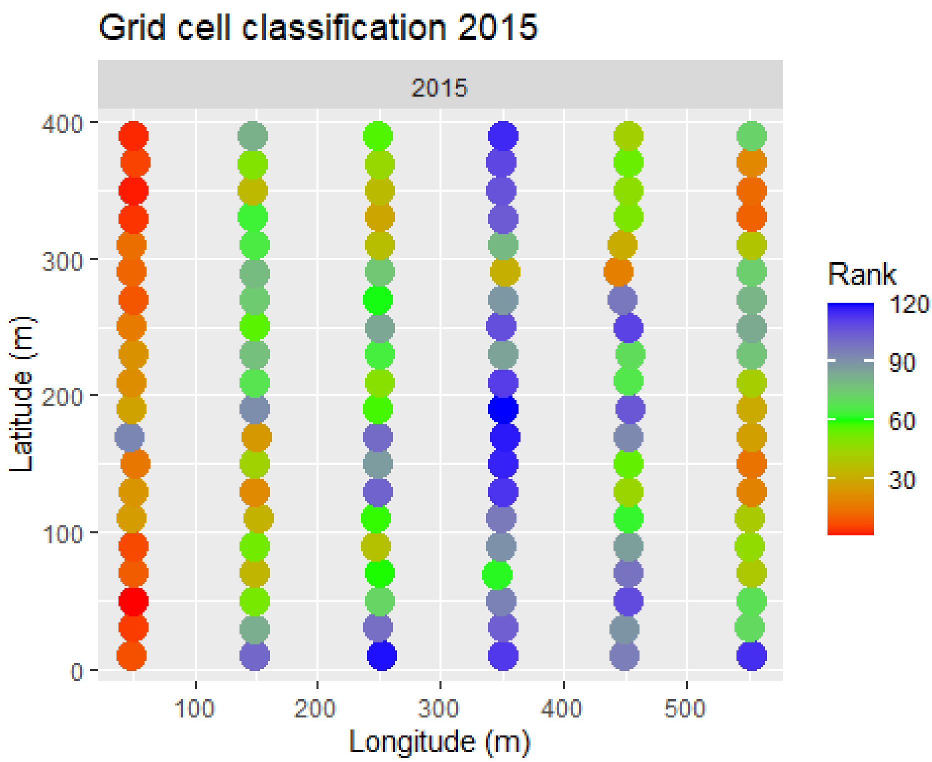
```
library(ggplot2)
library(wesanderson)
```

```
## Warning: package 'wesanderson' was built under R version 4.0.2
```

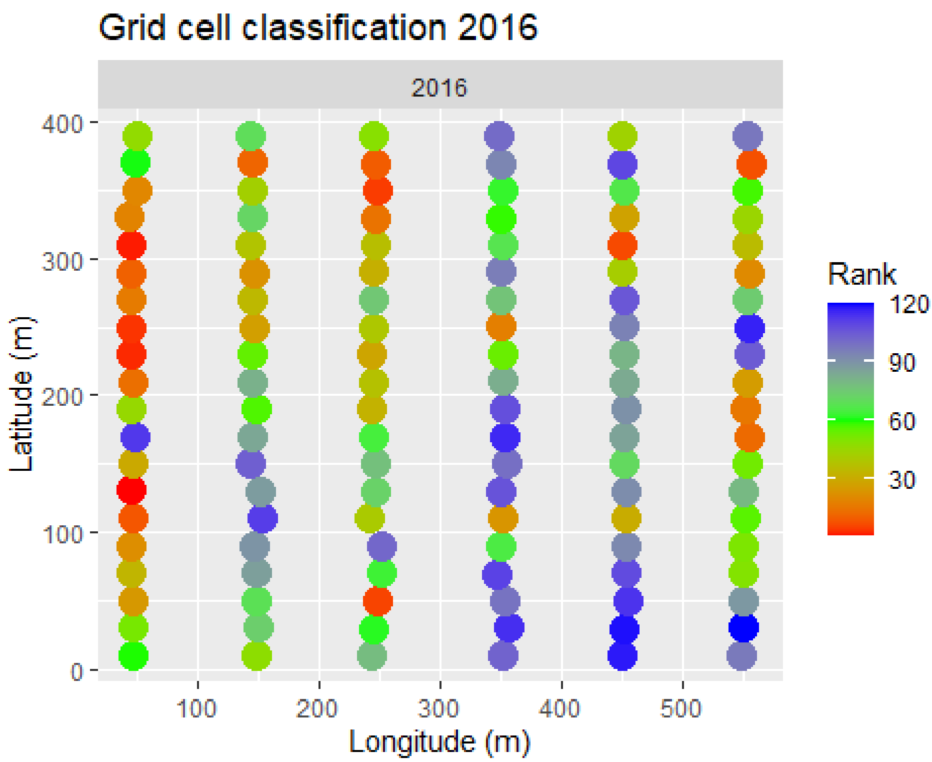
```
ggplot(data = RowHarvest.dat, mapping = aes(x = LongitudeAGG2013, y =
LatitudeAGG2013))+
geom_point(aes(color = rank2013), size = 5)+
scale_colour_gradientn(colours = rainbow(3))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2013) + ggtitle("Grid cell classification 2013")
```



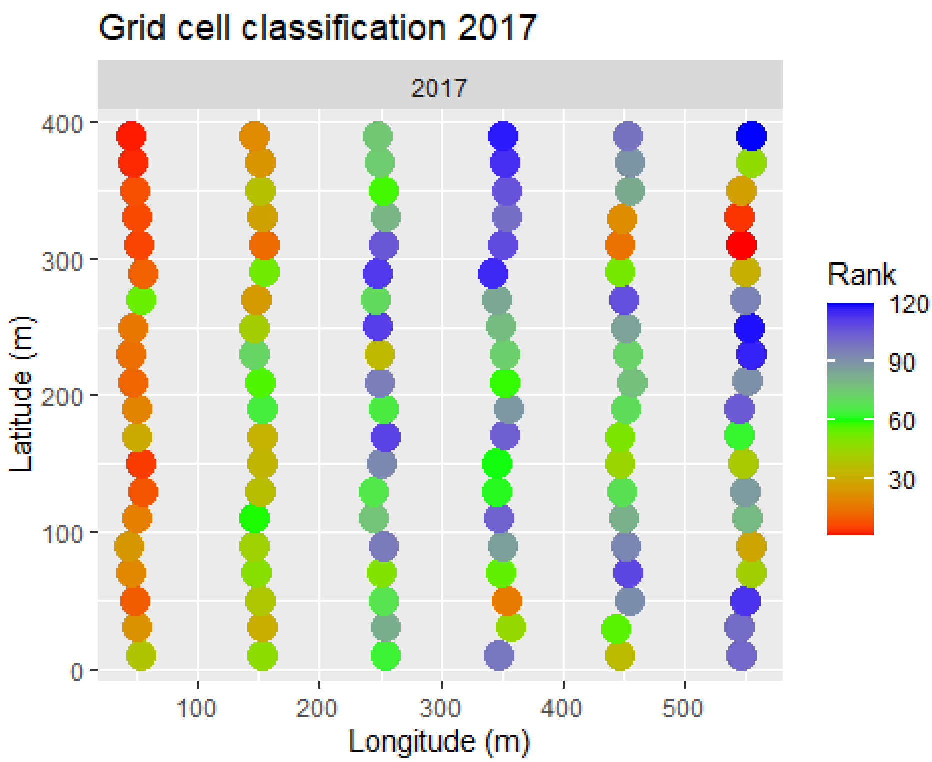

```
ggplot(data = RowHarvest.dat, mapping = aes(x = LongitudeAGG2015, y =
LatitudeAGG2015))+
geom_point(aes(color = rank2015), size = 5)+
scale_colour_gradientn(colours = rainbow(3))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2015) + ggtitle("Grid cell classification 2015")
```



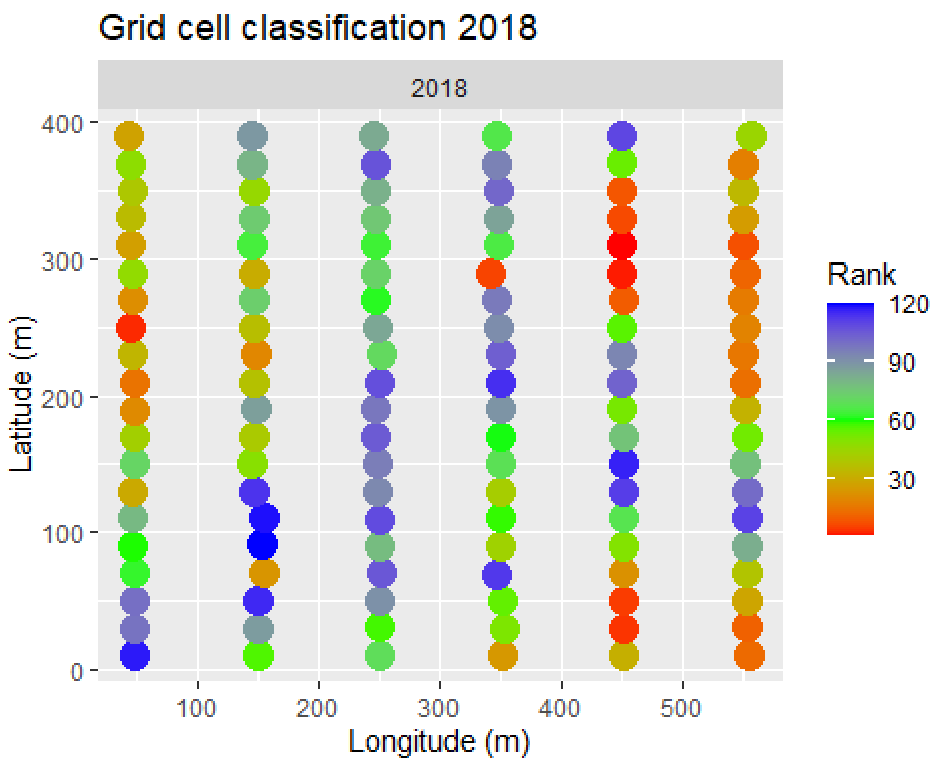

```
ggplot(data = RowHarvest.dat, mapping = aes(x = LongitudeAGG2016, y =
LatitudeAGG2016))+
geom_point(aes(color = rank2016), size = 5)+
scale_colour_gradientn(colours = rainbow(3))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2016) + ggtitle("Grid cell classification 2016")
```




```r
ggplot(data = RowHarvest.dat, mapping = aes(x = LongitudeAGG2017, y =
LatitudeAGG2017))+
geom_point(aes(color = rank2017), size = 5)+
scale_colour_gradientn(colours = rainbow(3))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2017) + ggtitle("Grid cell classification 2017")
```



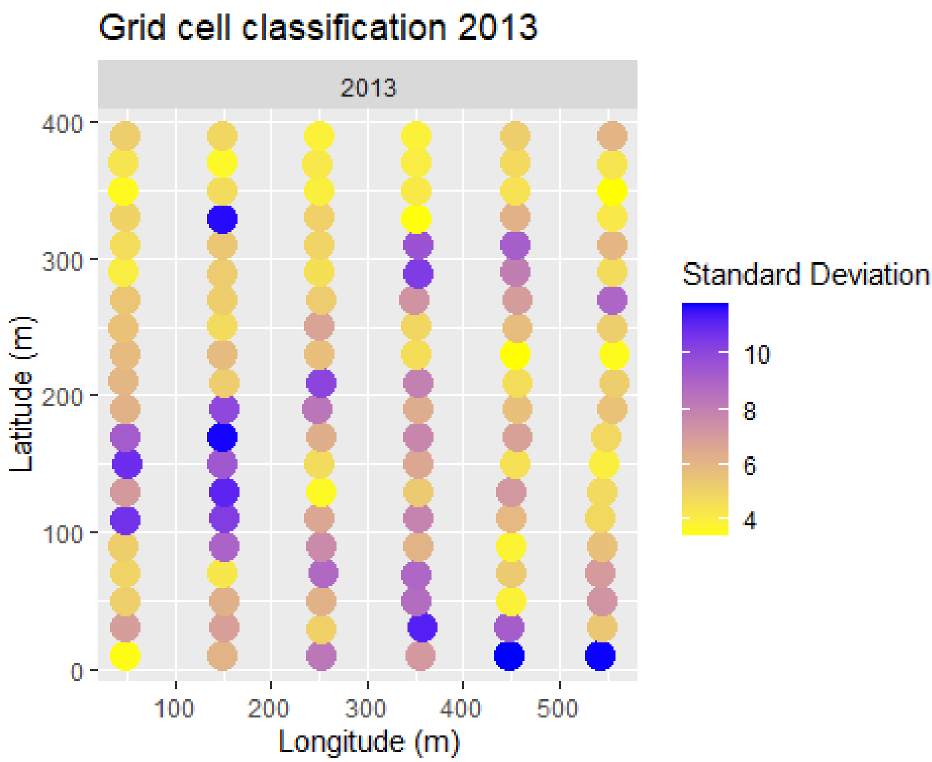

```
ggplot(data = RowHarvest.dat, mapping = aes(x = LongitudeAGG2018, y =
LatitudeAGG2018))+
geom_point(aes(color = rank2018), size = 5)+
scale_colour_gradientn(colours = rainbow(3))+
labs(color = "Rank", x = "Longitude (m)", y = "Latitude (m)") + facet_wrap(~
2018) + ggtitle("Grid cell classification 2018")
```



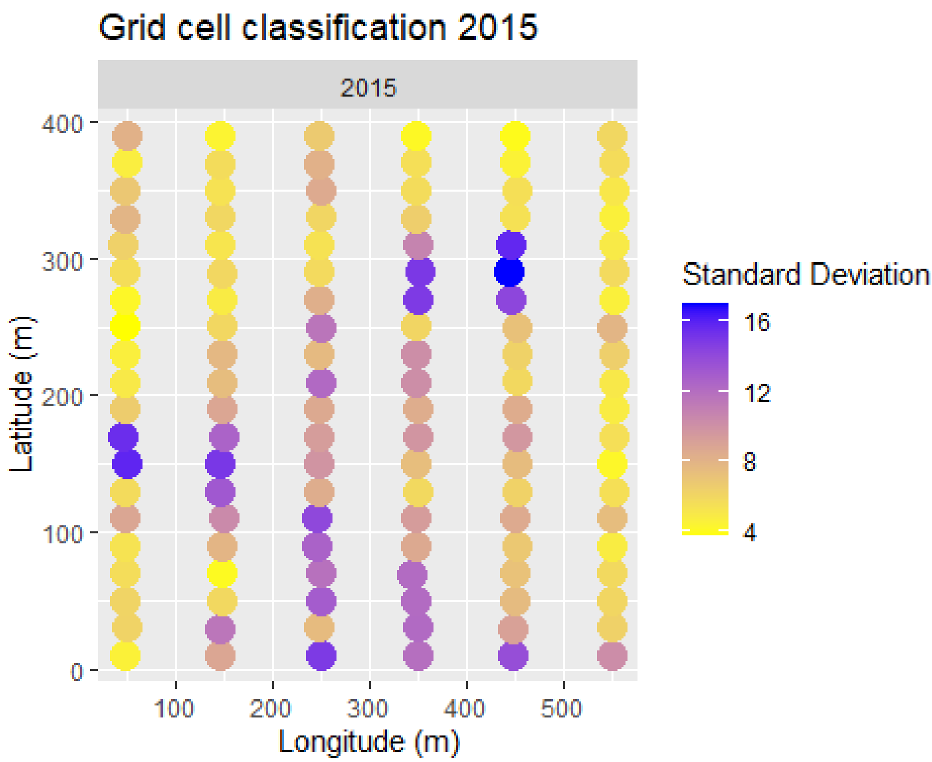
Classification of the standard deviation

Below we are doing a classification according to the standard deviation. The data shows a low to medium standard deviation with patches of areas with high standard deviation.

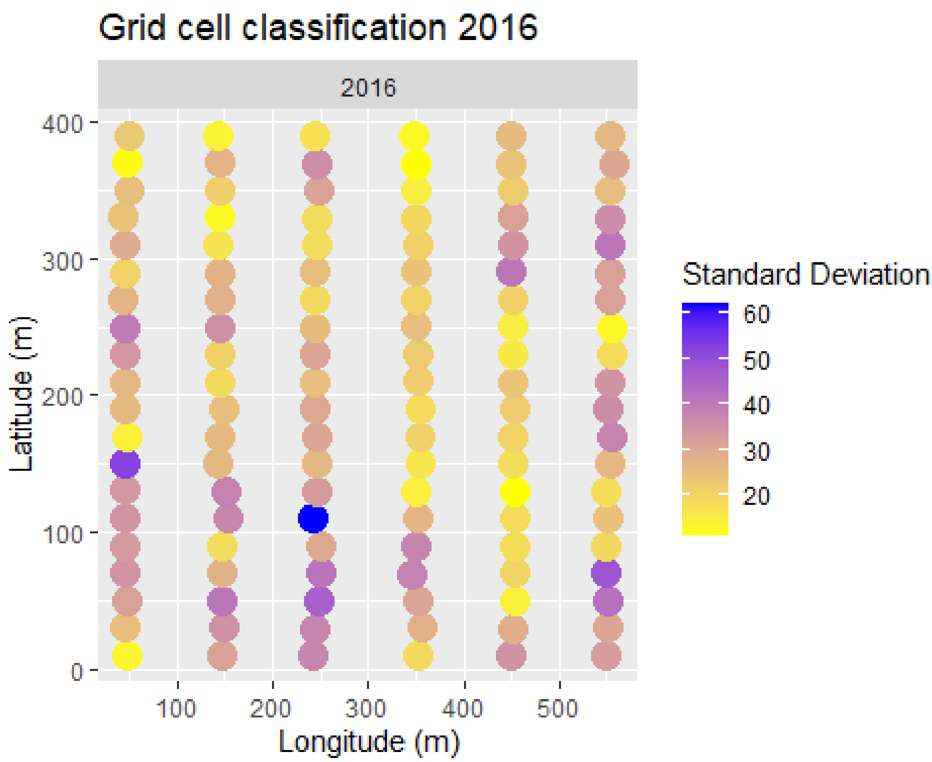
```
library(ggplot2)
ggplot(data = harvestAgg, mapping = aes(x = LongitudeAGG2013, y =
LatitudeAGG2013))+
geom_point(aes(color = SDYield2013), size = 5)+
scale_colour_gradient(low = "yellow", high = "blue") +
labs(color = "Standard Deviation", x = "Longitude (m)", y = "Latitude (m)") +
facet_wrap(~ 2013) + ggtitle("Grid cell classification 2013 ")
```



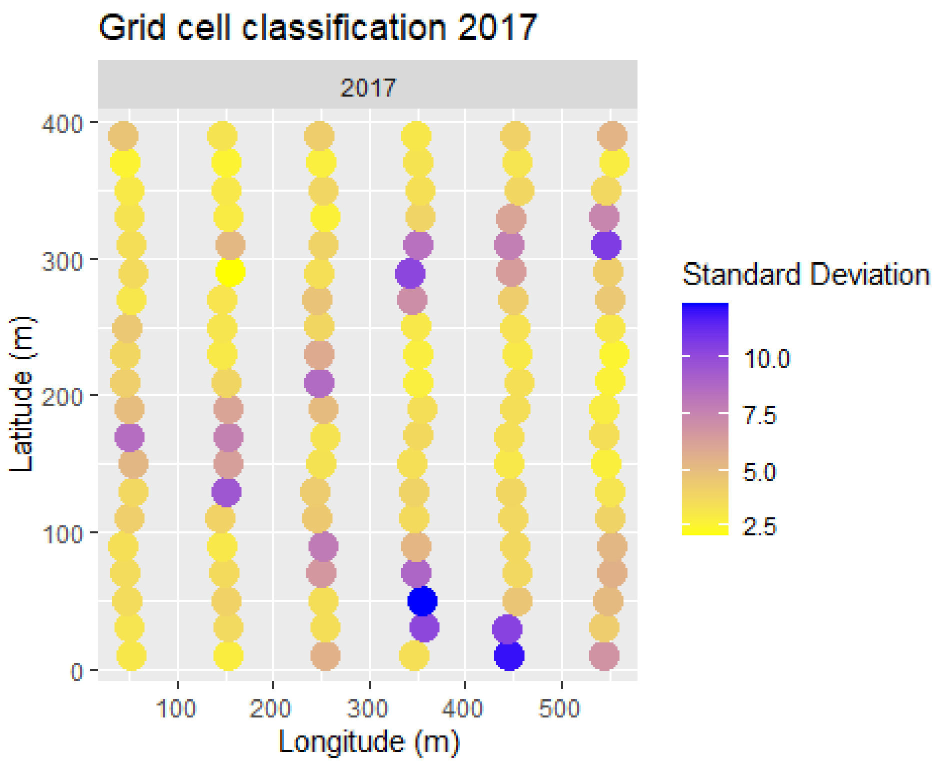

```
ggplot(data = harvestAgg, mapping = aes(x = LongitudeAGG2015, y =
LatitudeAGG2015))+
geom_point(aes(color = SDYield2015), size = 5)+
scale_colour_gradient(low = "yellow", high = "blue") +
labs(color = "Standard Deviation", x = "Longitude (m)", y = "Latitude (m)") +
facet_wrap(~ 2015) + ggtitle("Grid cell classification 2015 ")
```



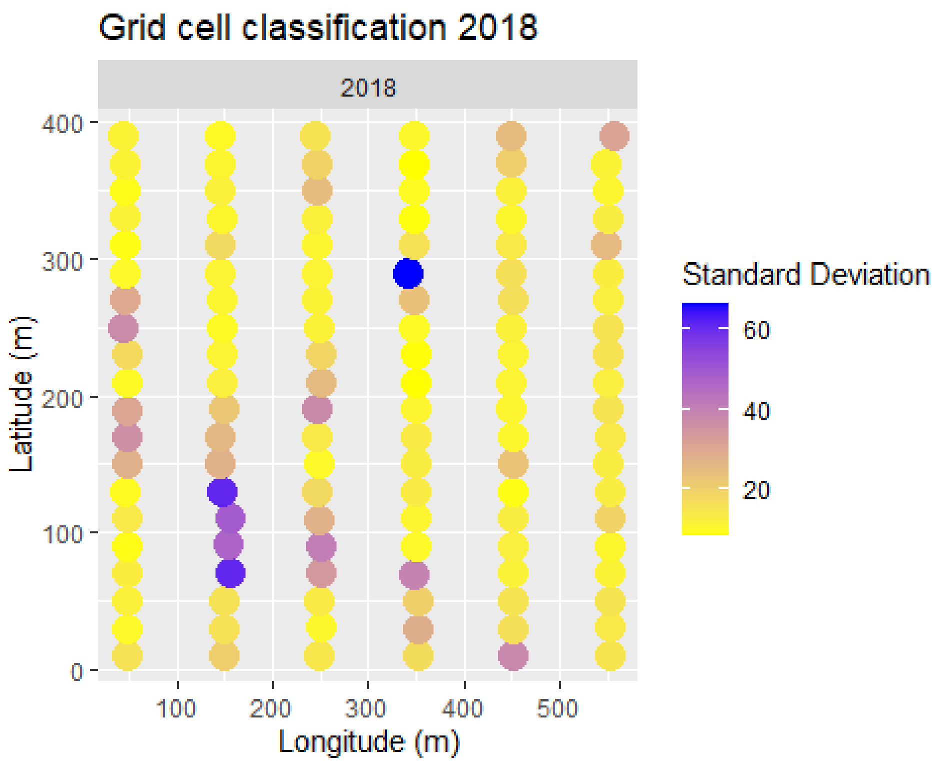

```
ggplot(data = harvestAgg, mapping = aes(x = LongitudeAGG2016, y =
LatitudeAGG2016))+
geom_point(aes(color = SDYield2016), size = 5)+
scale_colour_gradient(low = "yellow", high = "blue") +
labs(color = "Standard Deviation", x = "Longitude (m)", y = "Latitude (m)") +
facet_wrap(~ 2016) + ggtitle("Grid cell classification 2016")
```




```
ggplot(data = harvestAgg, mapping = aes(x = LongitudeAGG2017, y =
LatitudeAGG2017))+
geom_point(aes(color = SDYield2017), size = 5)+
scale_colour_gradient(low = "yellow", high = "blue") +
labs(color = "Standard Deviation", x = "Longitude (m)", y = "Latitude (m)") +
facet_wrap(~ 2017) + ggtitle("Grid cell classification 2017")
```



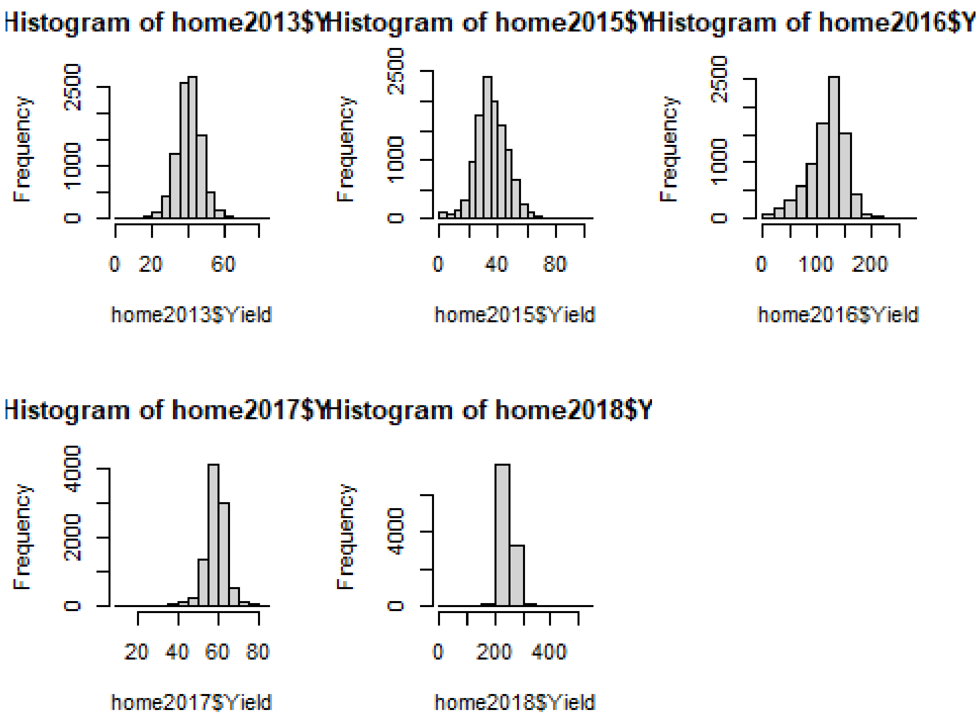

```r
ggplot(data = harvestAgg, mapping = aes(x = LongitudeAGG2018, y =
LatitudeAGG2018))+
geom_point(aes(color = SDYield2018), size = 5)+
scale_colour_gradient(low = "yellow", high = "blue") +
labs(color = "Standard Deviation", x = "Longitude (m)", y = "Latitude (m)") +
facet_wrap(~ 2018) + ggtitle("Grid cell classification 2018")
```



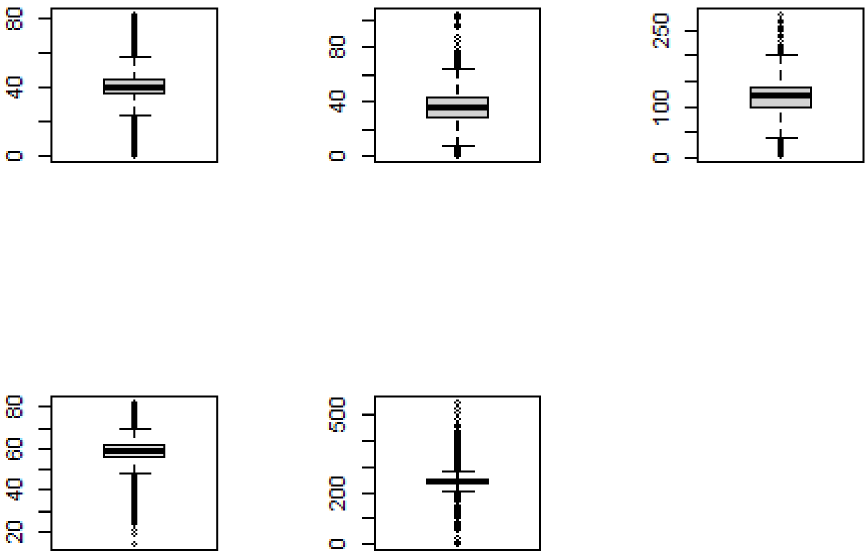
Distribution Plots Before Normalization

Below we are producing distribution plots of the data before cell divisions and normalization. It's easy to see that the data distribution is not normal. This is more evident in the box plots and qqnorms with a good amount of the data falling the outliers region.

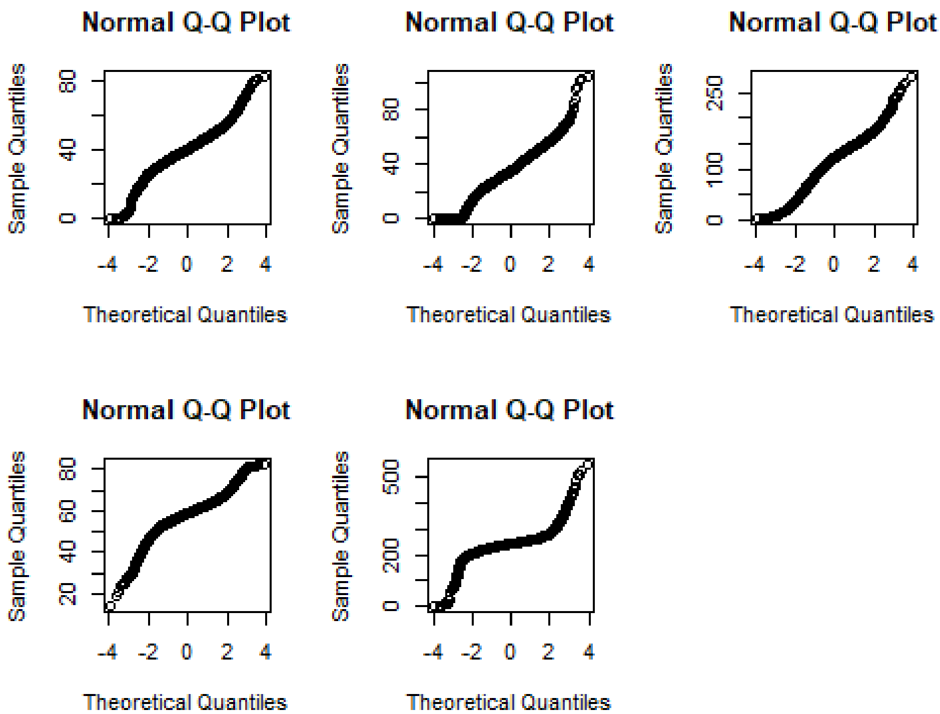
```
par(mfrow=c(2,3))
hist(home2013$Yield)
hist(home2015$Yield)
hist(home2016$Yield)
hist(home2017$Yield)
hist(home2018$Yield)

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




```r
boxplot(home2013$Yield)
boxplot(home2015$Yield)
boxplot(home2016$Yield)
boxplot(home2017$Yield)
boxplot(home2018$Yield)

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



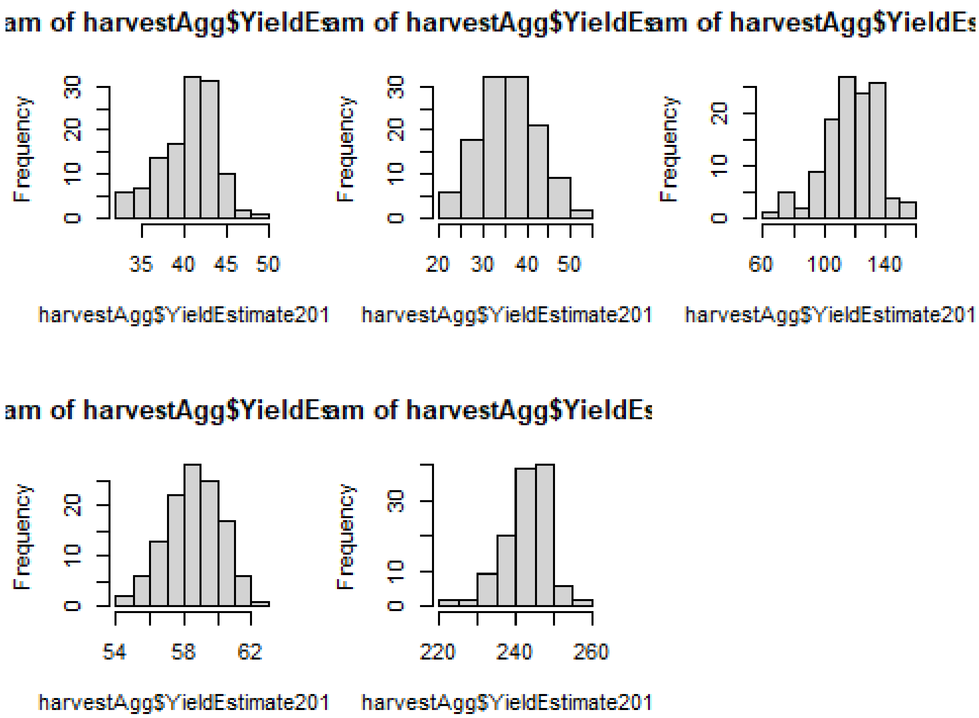

```r
qqnorm(home2013$Yield)
qqnorm(home2015$Yield)
qqnorm(home2016$Yield)
qqnorm(home2017$Yield)
qqnorm(home2018$Yield)
```



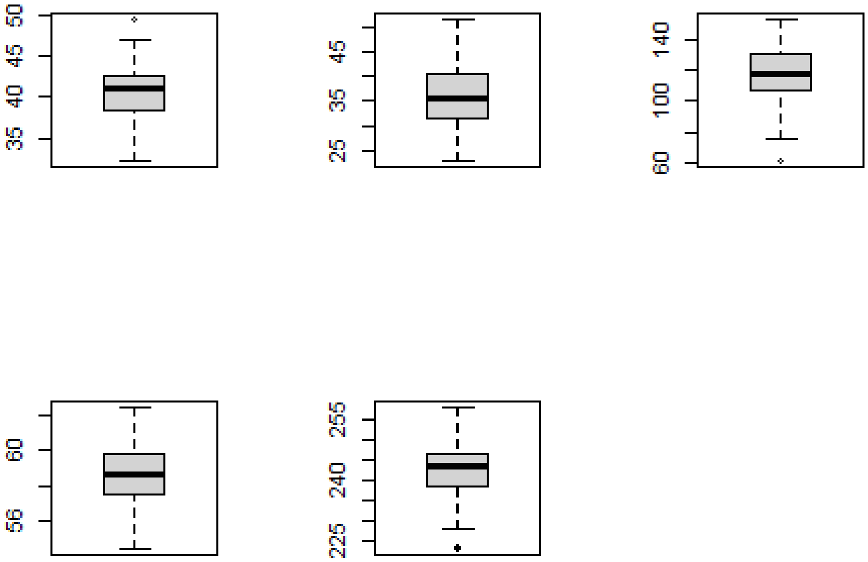
Distribution Plots After Normalization

Below are distribution plots after normalization. The data looks to be a lot closer to being in the normal distribution than before normalization. The distribution of the data is an important element in the data analysis because it's important your data follows a consistent path and not data that is all over the place, which makes it harder to draw conclusions from the data.

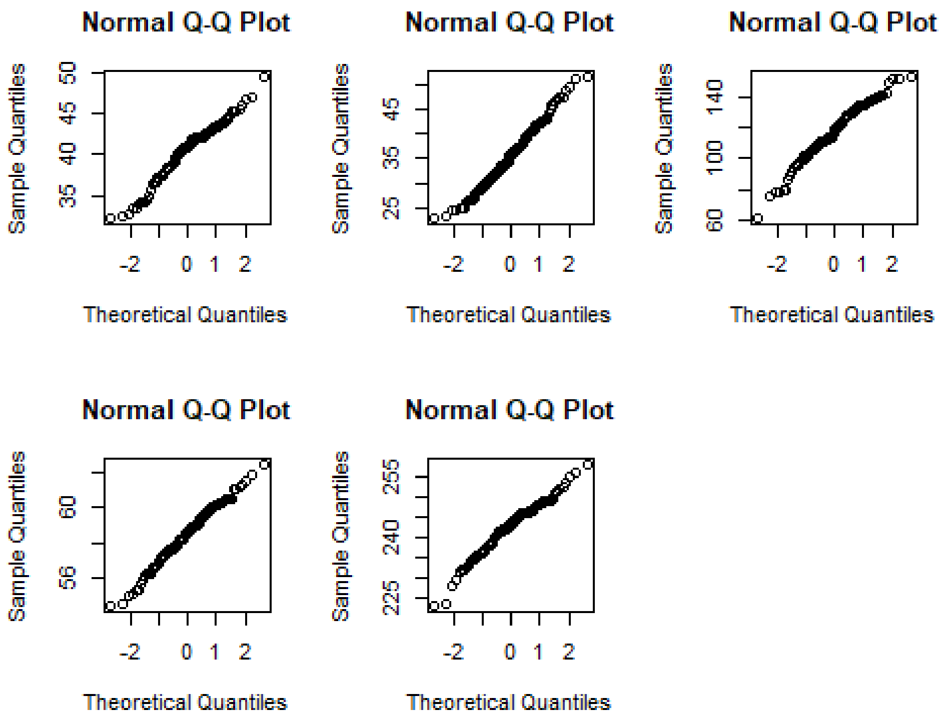
```
par(mfrow=c(2,3))
hist(harvestAgg$YieldEstimate2013)
hist(harvestAgg$YieldEstimate2015)
hist(harvestAgg$YieldEstimate2016)
hist(harvestAgg$YieldEstimate2017)
hist(harvestAgg$YieldEstimate2018)

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




```
boxplot(harvestAgg$YieldEstimate2013)
boxplot(harvestAgg$YieldEstimate2015)
boxplot(harvestAgg$YieldEstimate2016)
boxplot(harvestAgg$YieldEstimate2017)
boxplot(harvestAgg$YieldEstimate2018)

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




```r
qqnorm(harvestAgg$YieldEstimate2013)
qqnorm(harvestAgg$YieldEstimate2015)
qqnorm(harvestAgg$YieldEstimate2016)
qqnorm(harvestAgg$YieldEstimate2017)
qqnorm(harvestAgg$YieldEstimate2018)
```



Conclusion

First, we had to check for timestamps. The time stamps were analyzed to check if each data set was within the one-week interval. This was a constraint that needed to be upheld. The time stamp plots indicated that the field was harvest within or less than seven days.

We then divided the data from each year into 120 grid cells that are 20 by 6 or 100 by 20 meters. This meant averaging the Yields in each data by grid cells. Each grid cell needed to have more than 30 samples of Yield. This was a previous constraint that still needed to be upheld. The lowest samples we had in any grid cell from the 5 years was 58 samples. This allowed us to have a good amount of samples in each cell.

In order to contrast, we plotted the classification before normalization and after and difference was very evident. The further this, we also plotted the distribution of the data before normalization and after normalization and it supported what we saw in the classification. The data became more normal after we divided it into 120 grid cells. This was because by aggregating many samples that were in the bounds of each grid cell, it gave a more wholistic view of the data. Any discrepancies disappeared because we are working with the average of the samples rather the individual samples. To further this even more, we plotted the standard deviation and it showed a low to medium standard deviation after normalization.

Take Aways

After analyzing this data set, we have learned:

1. manipulate data tables (combine and merge)

2. normalize data to have a common scale (rank)

3. How to work with ggplots to classify data

4. How to transform date and time data

5. How to improve the distribution of data by increasing the sample size (in this case aggregating yield values into one grid cell)