Cleaning

Peter Claussen

9/5/2017

This is included to document how I've prepared a sample field.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#0072B2", "#D55E00", "#F0E442","#CC79A7","#
```

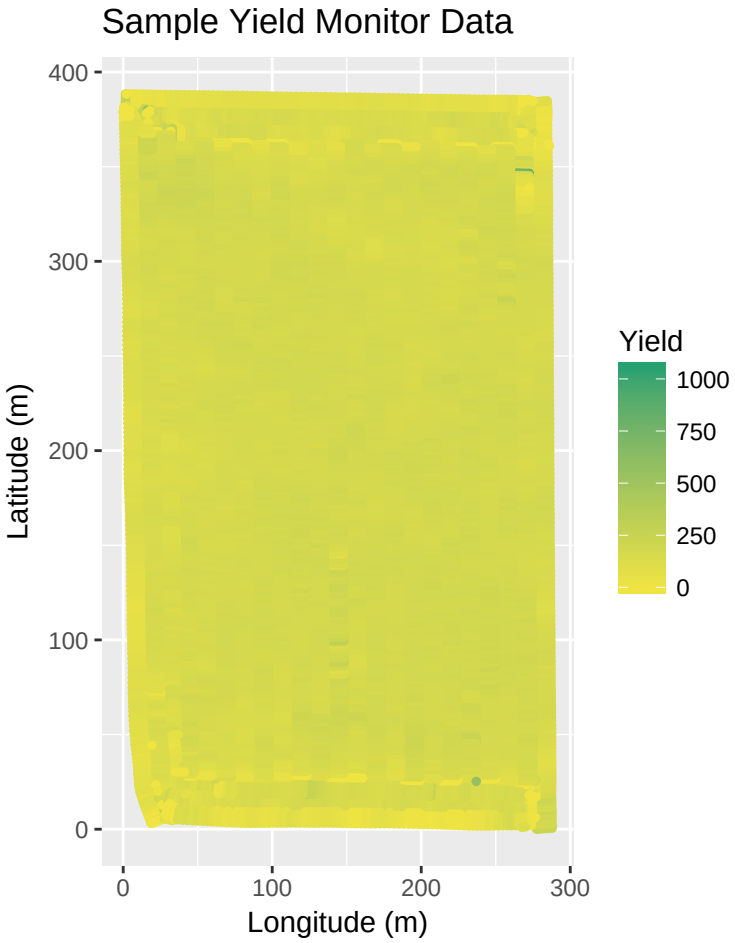
We start with a full field. The data were downloaded from http://myjohndeere.deere.com, imported into SMS Basic, http://www.agleader.com/products/sms-software/basic/, then exported to CSV. I've manually edited the header text to simplify column names.

```
sample.dat <- read.csv("Stevens East-Corn.csv",header=TRUE,comment.char = "#")
head(sample.dat)
```

```
##   Longitude Latitude        Field           Dataset Product ObjId Distance
## 1 -97.59137 44.09556 Stevens East Stevens East-Corn       c     1    2.096
## 2 -97.59135 44.09556 Stevens East Stevens East-Corn       A     2    2.096
## 3 -97.59133 44.09556 Stevens East Stevens East-Corn       A     3    2.096
## 4 -97.59131 44.09556 Stevens East Stevens East-Corn       A     4    2.096
## 5 -97.59129 44.09556 Stevens East Stevens East-Corn       A     5    2.096
## 6 -97.59127 44.09556 Stevens East Stevens East-Corn       A     6    2.096
##   Swath  Yield MarkID YldMassWet Moisture                  Desc Heading
## 1     5 243.57    111      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 2     5 243.57    112      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 3     5 243.57    113      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 4     5 243.57    114      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 5     5 243.57    115      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 6     5 243.57    116      13911    16.66 10/12/2015 4:45:12 PM  354.82
```

```
sample.dat$LonM <- sample.dat$Longitude - min(sample.dat$Longitude)
sample.dat$LatM <- sample.dat$Latitude - min(sample.dat$Latitude)
latMid <- (min(sample.dat$Latitude) + max(sample.dat$Latitude))/2
m_per_deg_lat = 111132.954 - 559.822 * cos( 2.0 * latMid ) + 1.175 * cos( 4.0 * latMid)
m_per_deg_lon = (3.14159265359/180 ) * 6367449 * cos ( latMid )
sample.dat$LonM <- sample.dat$LonM*m_per_deg_lon
sample.dat$LatM <- sample.dat$LatM*m_per_deg_lat
```

```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



We have a large number of duplicated values at slightly different points - the harvester has multiple channels, but each are recording the same yield values. We remove duplicated yields by aggregating over the same sampling interval. I couldn't find a specific date-time field in the SMS import options, so I save the timestamp as `Desc`

```
head(sample.dat)
```

```
##   Longitude Latitude        Field           Dataset Product ObjId Distance
## 1 -97.59137 44.09556 Stevens East Stevens East-Corn       c     1    2.096
## 2 -97.59135 44.09556 Stevens East Stevens East-Corn       A     2    2.096
## 3 -97.59133 44.09556 Stevens East Stevens East-Corn       A     3    2.096
## 4 -97.59131 44.09556 Stevens East Stevens East-Corn       A     4    2.096
## 5 -97.59129 44.09556 Stevens East Stevens East-Corn       A     5    2.096
## 6 -97.59127 44.09556 Stevens East Stevens East-Corn       A     6    2.096
##   Swath  Yield MarkID YldMassWet Moisture                  Desc Heading
## 1     5 243.57    111      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 2     5 243.57    112      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 3     5 243.57    113      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 4     5 243.57    114      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 5     5 243.57    115      13911    16.66 10/12/2015 4:45:12 PM  354.82
## 6     5 243.57    116      13911    16.66 10/12/2015 4:45:12 PM  354.82
##       LonM      LatM
## 1 277.3767 0.0000000
## 2 279.4732 0.1105889
## 3 281.5685 0.2222838
```

```
## 4 283.6650 0.3328727
## 5 285.7604 0.4445675
## 6 287.8569 0.5551564
```

```
aggregate.dat <- aggregate(sample.dat[,-c(3:5,13)], by=list(sample.dat$Desc), FUN=mean, na.rm=TRUE)
head(aggregate.dat)
```

```
##                 Group.1 Longitude Latitude ObjId Distance Swath   Yield
## 1 10/12/2015 4:45:12 PM -97.59132 44.09556   3.5   2.0960     5 243.570
## 2 10/12/2015 4:45:13 PM -97.59132 44.09557  12.5   2.9325     5 228.975
## 3 10/12/2015 4:45:15 PM -97.59132 44.09559  21.5   4.4590     5 167.830
## 4 10/12/2015 4:45:16 PM -97.59132 44.09560  27.5   4.1630     5 193.060
## 5 10/12/2015 4:45:17 PM -97.59132 44.09561  33.5   4.3270     5 173.820
## 6 10/12/2015 4:45:18 PM -97.59132 44.09562  39.5   4.3270     5 202.320
##   MarkID YldMassWet Moisture Heading     LonM      LatM
## 1  113.5    13911.0    16.66 354.820 282.6168 0.2775782
## 2  113.5    13077.5    16.66 356.480 282.5033 1.5261272
## 3  113.5     9585.0    16.66 359.330 282.4839 3.4011626
## 4  113.5    11026.0    16.66 359.540 282.4578 4.6679588
## 5  113.5     9927.5    16.66 359.190 282.4613 5.9810180
## 6  113.5    11555.0    16.66   0.211 282.5073 7.2955518
```

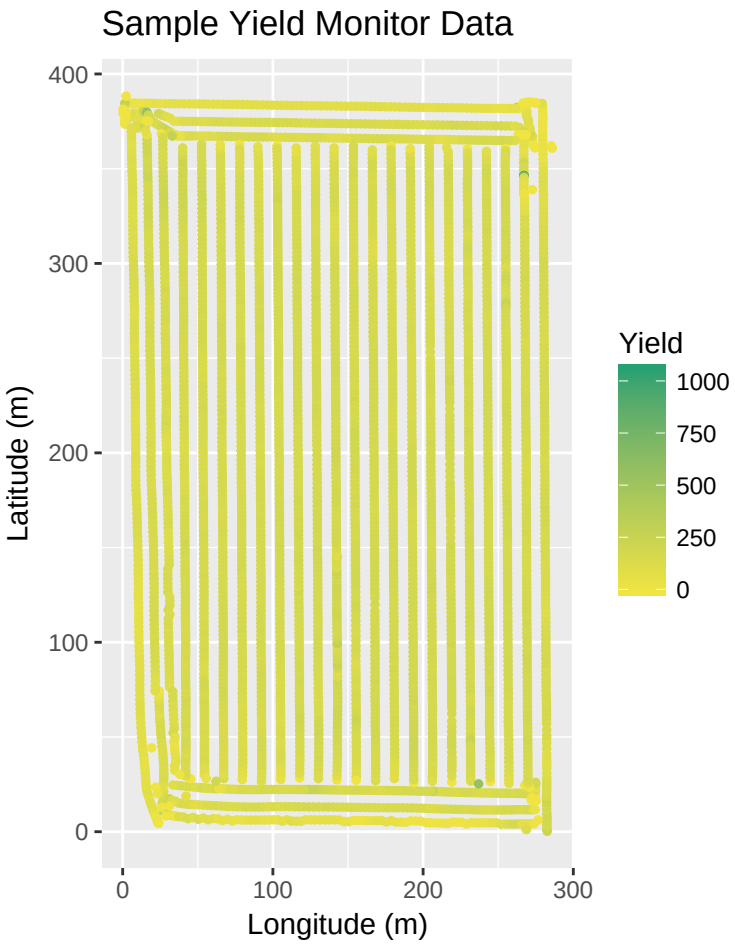
```
aggregate.dat$Product <- aggregate(sample.dat["Product"], by=list(sample.dat$Desc), FUN=function(x){x[
```

Convert the time stamp to seconds.

```
aggregate.dat$DateTime <- as.POSIXct(as.character(aggregate.dat[,1]), format = "%m/%d/%Y %I:%M:%S %p",t
aggregate.dat$Seconds <- aggregate.dat$DateTime - aggregate.dat$DateTime[1]

sample.dat <- aggregate.dat
```

```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```

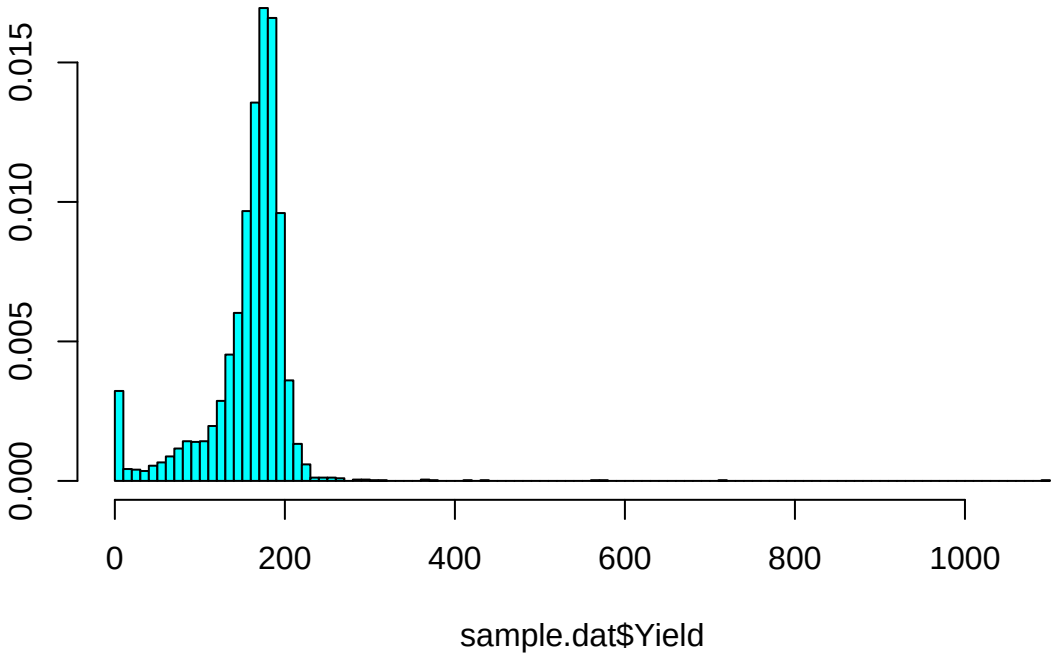


Perform a simple check for outliers and remove excessively large yield estimates.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.3.2
```

```
truehist(sample.dat$Yield)
```

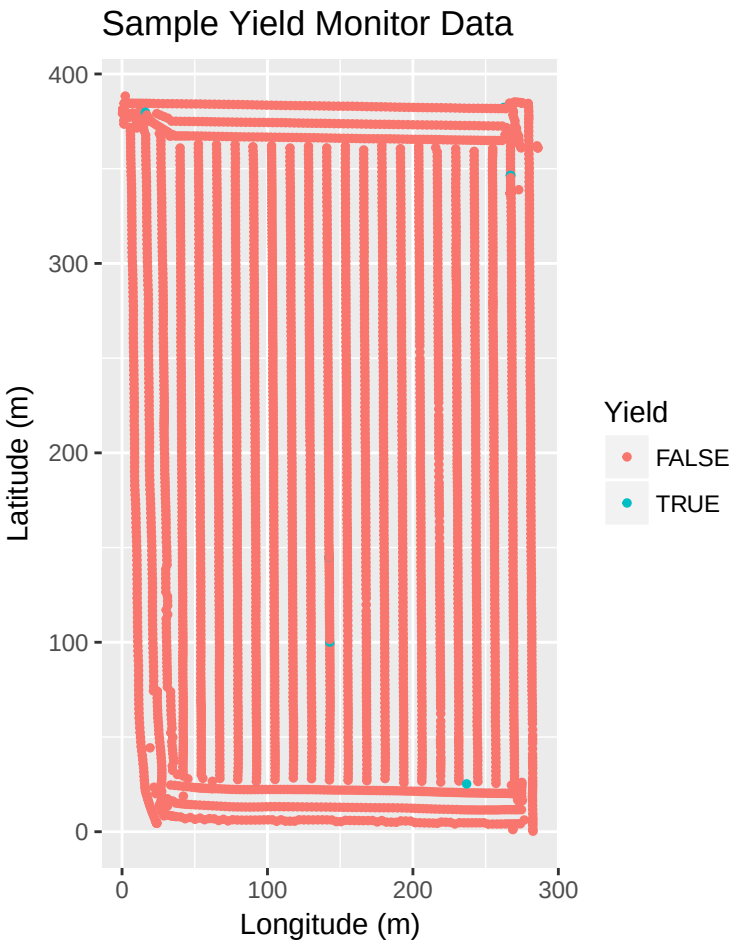



```
max.yield <- 300
sum(sample.dat$Yield>max.yield)
```

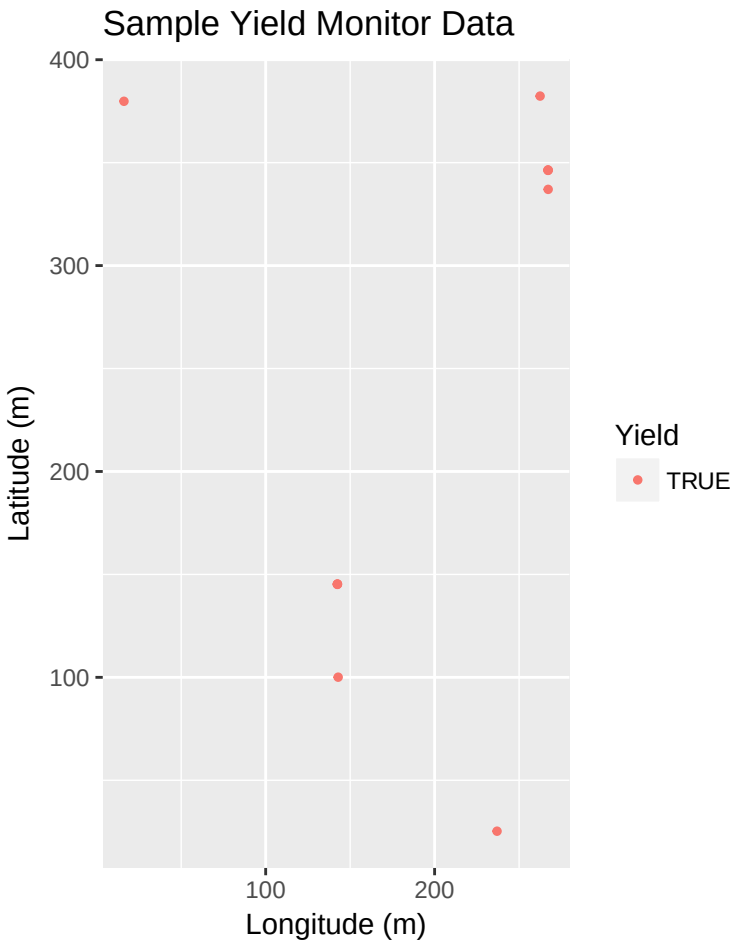
```
## [1] 11
```

Are the outliers randomly distributed?

```
sample.dat$Outlier <- sample.dat$Yield>max.yield
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Outlier),size=1) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```

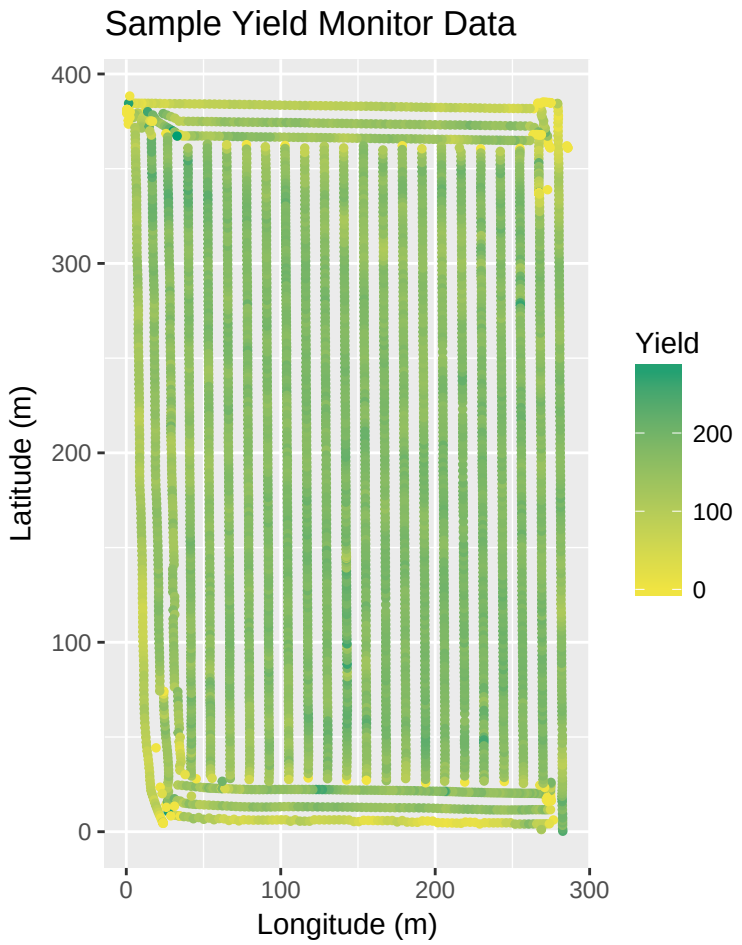



```
ggplot(sample.dat[sample.dat$Outlier,], aes(LonM,LatM)) +
geom_point(aes(colour = Outlier),size=1) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```

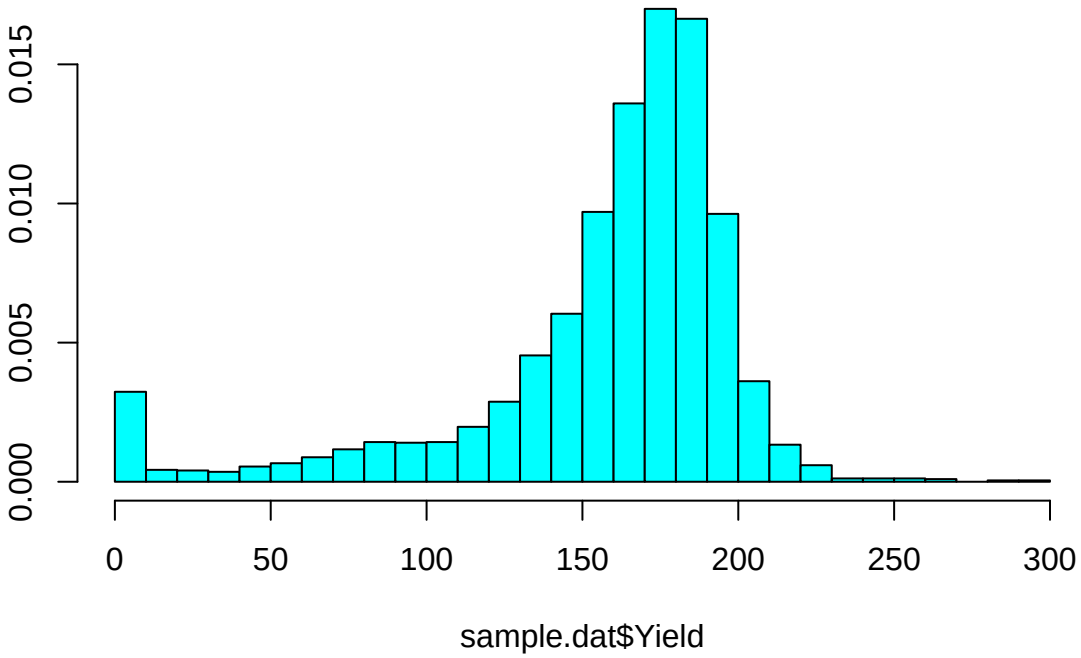


They're mostly scattered, so we can go ahead and drop them

```
sample.dat <- subset(sample.dat,!sample.dat$Outlier)
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



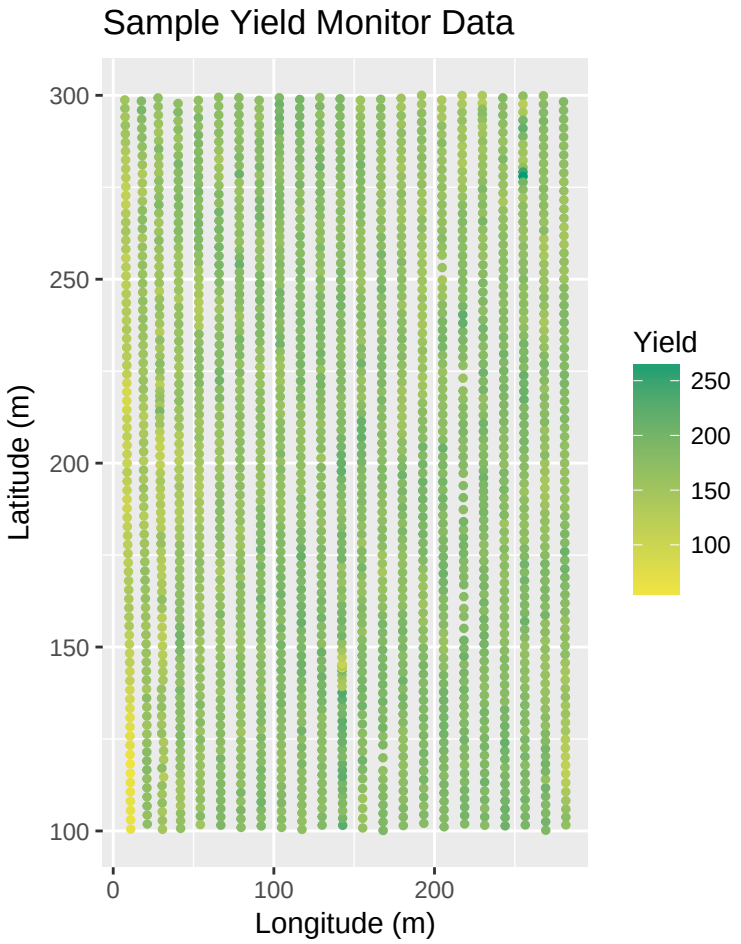

```
truehist(sample.dat$Yield)
```



Now trim 100 meters off top and bottom; this will remove endrows and give us a smaller set to work with

```
sample.dat <- subset(sample.dat,sample.dat$LatM<=300)
sample.dat <- subset(sample.dat,sample.dat$LatM>=100)
```

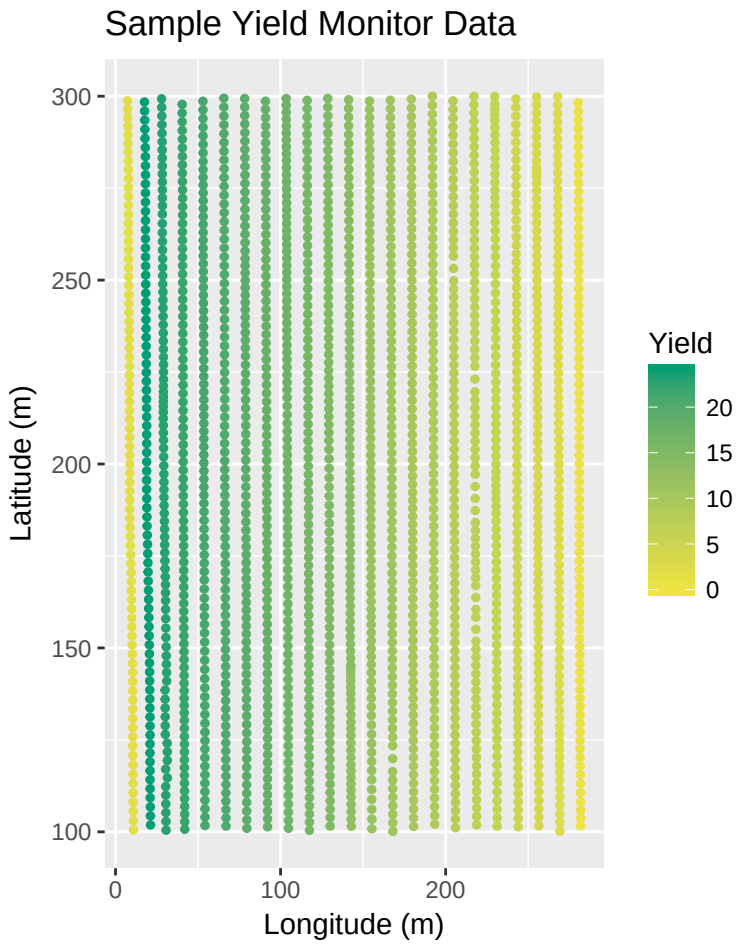
```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



Count up passes.

```
i <- 2
rows <- dim(sample.dat)[1]
PassNum <- 1
sample.dat$PassNum<- 0
while(i<=rows) {
  currentTime <- sample.dat$Seconds[i]
  previousTime <- sample.dat$Seconds[i-1]
  if((currentTime-previousTime)>5) {
    PassNum <- PassNum+1
  }
  sample.dat$PassNum[i] <- PassNum
  i <- i+1
}
```

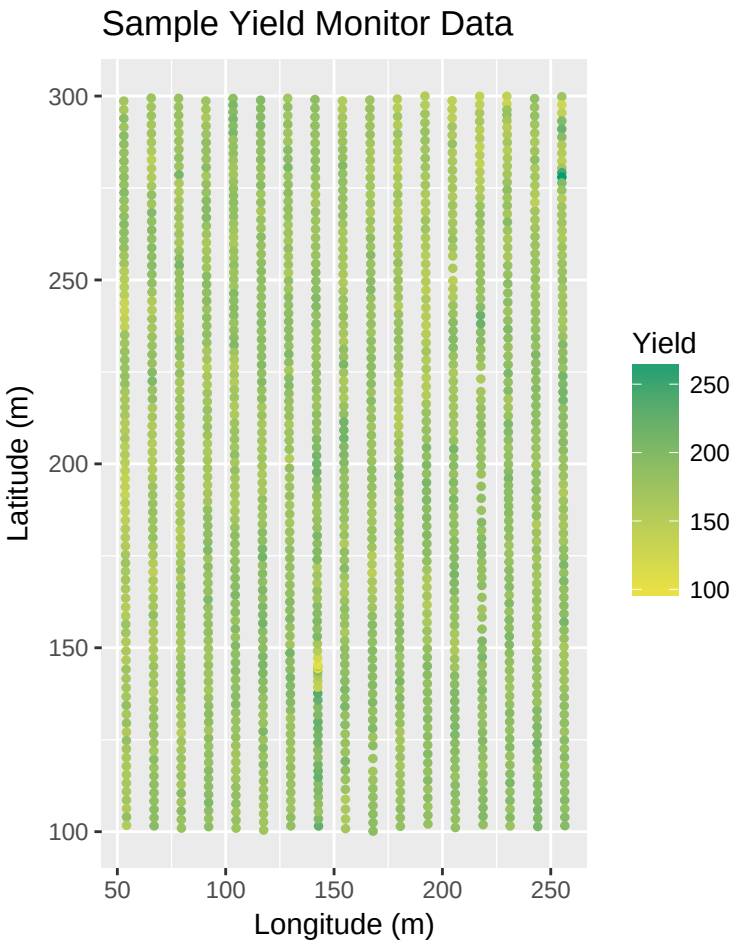
```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = PassNum),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



Take out the first 3 and last 3 passes; this will remove border rows. It's not exactly balanced, but it's close enough for our purposes.

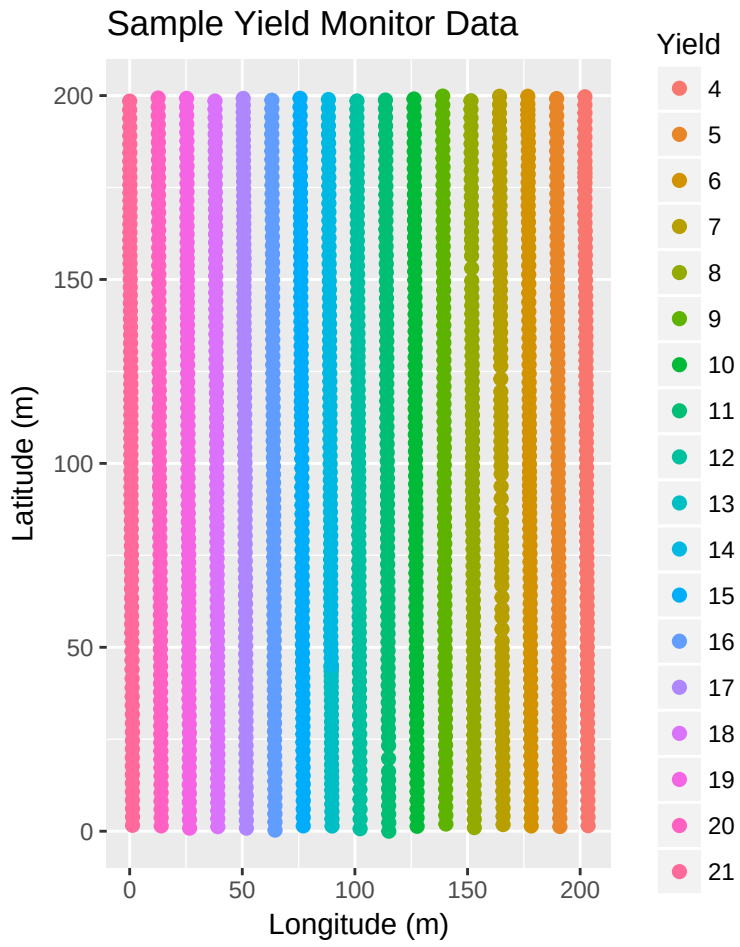
```
sample.dat <- subset(sample.dat,sample.dat$PassNum<(max(sample.dat$PassNum)-2))
sample.dat <- subset(sample.dat,sample.dat$PassNum>3)
sample.dat$Pass <- as.factor(sample.dat$PassNum)
```

```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```

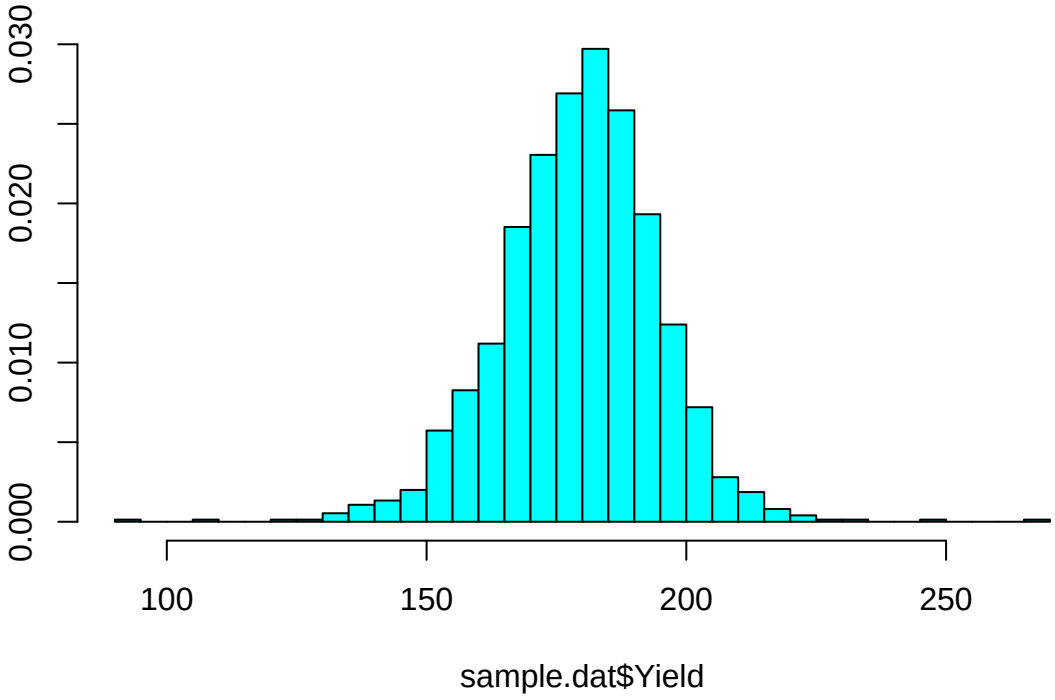



```
sample.dat$LonM <- sample.dat$LonM - min(sample.dat$LonM)
sample.dat$LatM <- sample.dat$LatM - min(sample.dat$LatM)
```

```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Pass),size=2) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



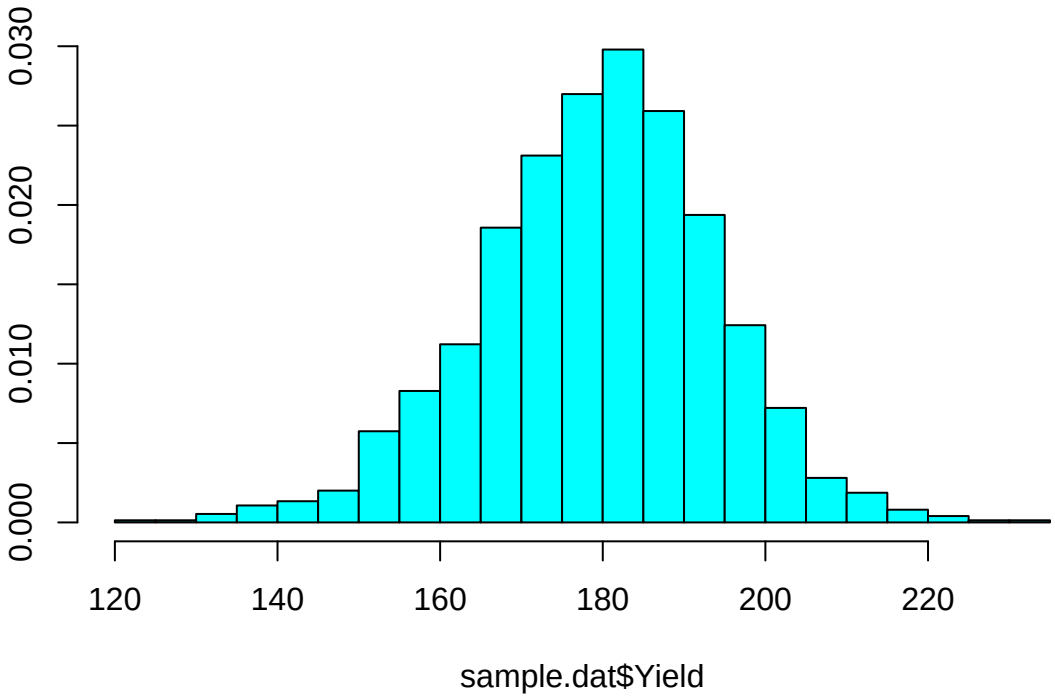

```
truehist(sample.dat$Yield)
```



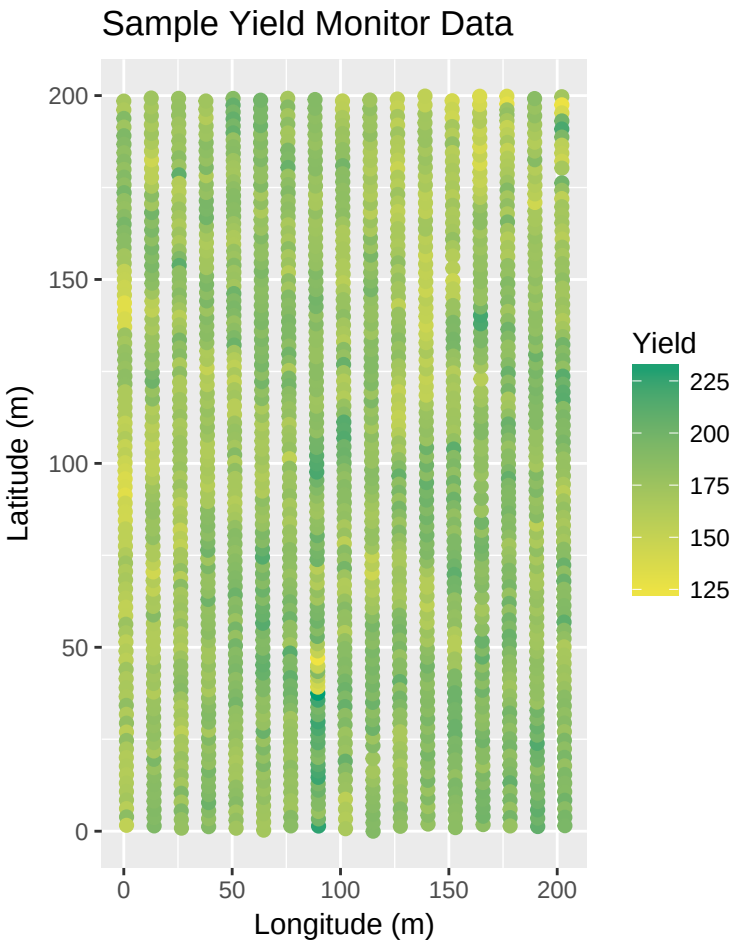
There might be four points that we could consider outliers; let's remove those as well.

```
sample.dat <- subset(sample.dat,sample.dat$Yield<=240)
sample.dat <- subset(sample.dat,sample.dat$Yield>=120)
```

```
truehist(sample.dat$Yield)
```




```
ggplot(sample.dat, aes(LonM,LatM)) +
geom_point(aes(colour = Yield),size=2) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data")
```



Save the data.

```
head(sample.dat)
```

```
##                   Group.1 Longitude Latitude  ObjId Distance Swath  Yield
## 1274 10/12/2015 5:14:52 PM -97.59157 44.09827 7076.5    7.247     5 172.51
## 1275 10/12/2015 5:14:53 PM -97.59157 44.09825 7082.5    7.247     5 125.27
## 1276 10/12/2015 5:14:54 PM -97.59157 44.09823 7088.5    7.247     5 140.30
## 1277 10/12/2015 5:14:55 PM -97.59157 44.09821 7094.5    7.214     5 199.43
## 1278 10/12/2015 5:14:56 PM -97.59157 44.09819 7100.5    7.247     5 217.90
## 1279 10/12/2015 5:14:57 PM -97.59157 44.09817 7106.5    7.247     5 193.78
##      MarkID YldMassWet Moisture Heading     LonM     LatM Product
## 1274  113.5     9705.7     15.4  179.85 202.1152 199.6867       A
## 1275  113.5     7047.8     15.4  180.04 202.1249 197.4888       A
## 1276  113.5     7893.7     15.4  179.94 202.1334 195.2949       A
## 1277  113.5    11220.0     15.4  180.21 202.1372 193.1049       A
## 1278  113.5    12260.0     15.4  180.15 202.1104 190.9106       A
## 1279  113.5    10903.0     15.4  179.92 202.1234 188.7148       A
##                  DateTime   Seconds Outlier PassNum Pass
## 1274 2015-10-12 17:14:52 1780 secs   FALSE       4    4
## 1275 2015-10-12 17:14:53 1781 secs   FALSE       4    4
## 1276 2015-10-12 17:14:54 1782 secs   FALSE       4    4
## 1277 2015-10-12 17:14:55 1783 secs   FALSE       4    4
## 1278 2015-10-12 17:14:56 1784 secs   FALSE       4    4
## 1279 2015-10-12 17:14:57 1785 secs   FALSE       4    4
```

```r
save(sample.dat,file="sample.dat.Rda")
write.csv(sample.dat,file="sample.csv")
```