Case Study 1, Part 1

Peter Claussen

10/3/2017

```
library(MASS)
```

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

```
library(ggplot2)
```

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

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

We wish to combine four yield maps from a single field.

Processing

The first step is to load each file and process

```
Corn2013.dat <- read.csv("Corn 2013.csv",header=TRUE,comment.char = "#")
Soybean2014.dat <- read.csv("Soybeans 2014.csv",header=TRUE,comment.char = "#")
Corn2015.dat <- read.csv("Corn 2015.csv",header=TRUE,comment.char = "#")
Soybean2016.dat <- read.csv("Soybeans 2016.csv",header=TRUE,comment.char = "#")
```

We write a function to standardize each file.

```
metric_from_gps <- function(longitude, latitude, origin=c(0,0)) {
  if(origin[1]==0) {
    origin <- c(min(longitude),min(latitude))
  }
  mid_latitude <- (origin[2] + max(latitude))/2

  easting <- longitude - origin[1]
  northing <- latitude - origin[2]

  m_per_deg_lat = 111132.954 - 559.822 * cos(2.0 * mid_latitude) + 1.175 * cos(4.0 * mid_latitude)
  m_per_deg_lon = (3.14159265359/180 ) * 6367449 * cos(mid_latitude)
  easting <- easting*m_per_deg_lon
  northing <- northing*m_per_deg_lat

  return(list(northing=northing,
              easting=easting,
              origin=origin,
              m_per_deg_lon=m_per_deg_lon,
              m_per_deg_lat=m_per_deg_lat))
}

standardize.field <- function(field.dat,origin=c(0,0)) {

  #we will aggregate those values which can be averaged.
  col.names = names(field.dat)
```

```
  drop.columns = c(
    which(col.names=="Field"),
    which(col.names=="Dataset"),
    which(col.names=="Product"),
    which(col.names=="Desc")
  )
  aggregate.dat <- aggregate(field.dat[,-drop.columns],
                             by=list(field.dat$Desc),
                             FUN=mean, na.rm=TRUE)
  aggregate.dat$Product <- aggregate(field.dat[,"Product"],
                                     by=list(field.dat$Desc), FUN=function(x){x[1]})[,2]
  aggregate.dat$DateTime <- as.POSIXct(as.character(aggregate.dat[,1]),
                                       format = "%m/%d/%Y %I:%M:%S %p",tz = "")
  aggregate.dat$Seconds <- aggregate.dat$DateTime - aggregate.dat$DateTime[1]
  aggregate.dat <- aggregate.dat[,-1]

  #remove all non-trivial values
  aggregate.dat <- subset(aggregate.dat,aggregate.dat$Yield!=0)

  meters <- metric_from_gps(aggregate.dat$Longitude,
                            aggregate.dat$Latitude,
                            origin)
  #convert longitude and latitude to meters
  aggregate.dat$Easting <- meters$easting
  aggregate.dat$Northing <- meters$northing

  aggregate.dat$Rank <- rank(aggregate.dat$Yield)
  aggregate.dat$Quantile <- aggregate.dat$Rank/length(aggregate.dat$Rank)
  aggregate.dat$Percent <- 100*aggregate.dat$Yield/median(aggregate.dat$Yield)
  return(aggregate.dat)
}
```

We will anchor each field to the same origin, so

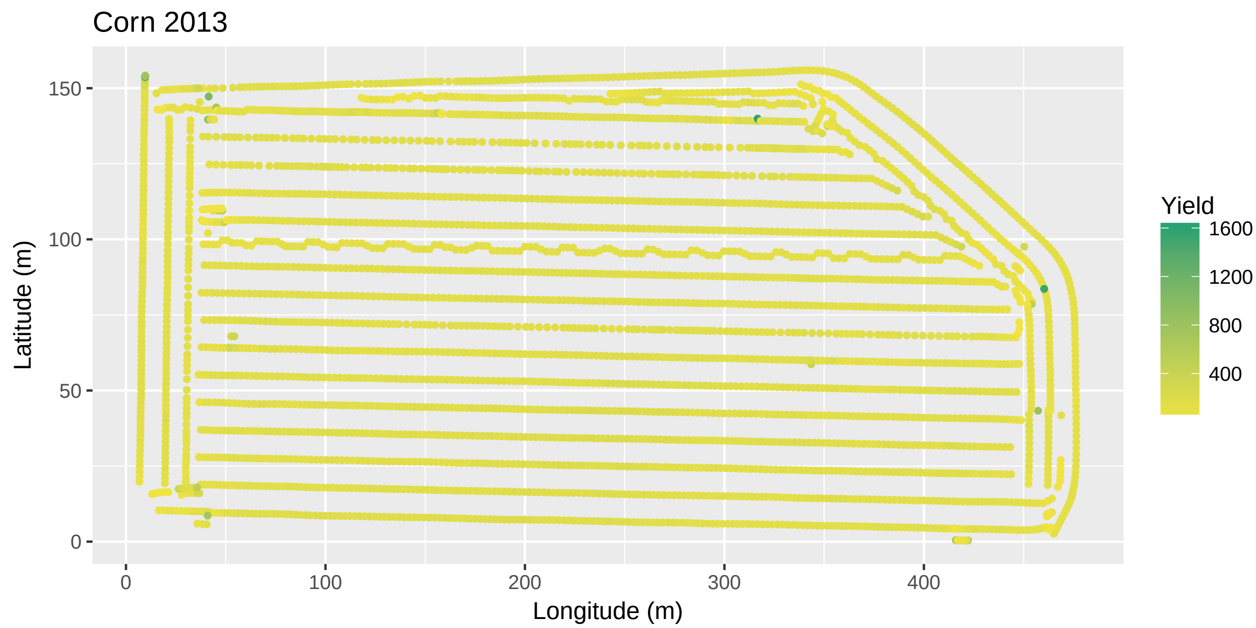
```
origin <- c(
  min(Corn2013.dat$Longitude, Soybean2014.dat$Longitude, Corn2015.dat$Longitude, Soybean2016.dat$Longit
  min(Corn2013.dat$Latitude, Soybean2014.dat$Latitude, Corn2015.dat$Latitude, Soybean2016.dat$Latitude)
)
origin
```

```
## [1] -97.59976  44.09562
```

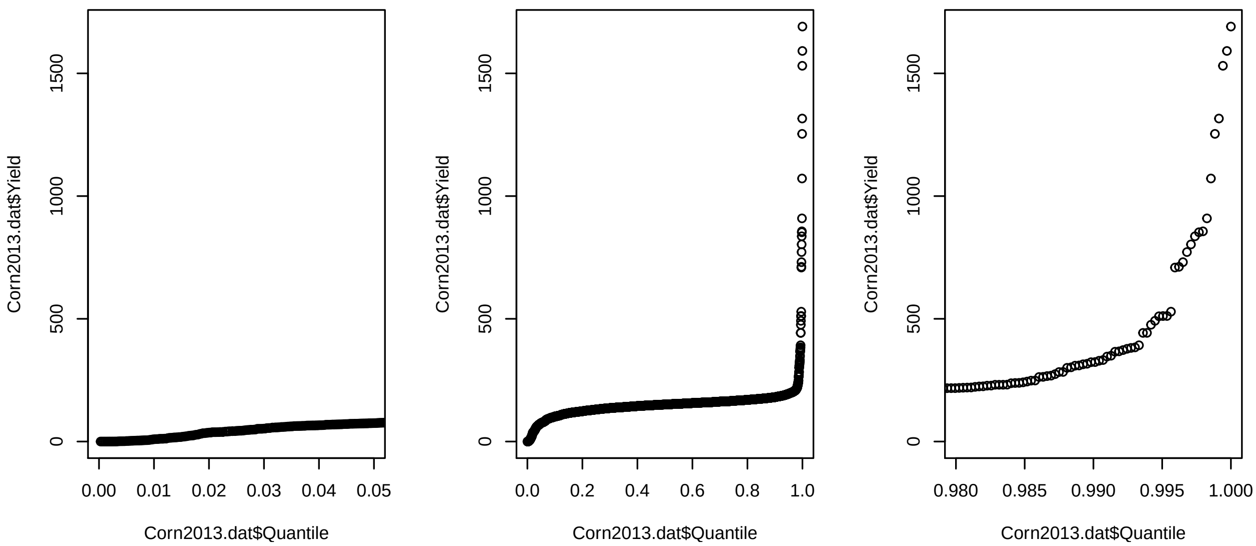
Corn, 2013

```
Corn2013.dat <- standardize.field(Corn2013.dat,origin)
```

```
ggplot(Corn2013.dat, aes(Easting,Northing)) +
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 = "Corn 2013")
```

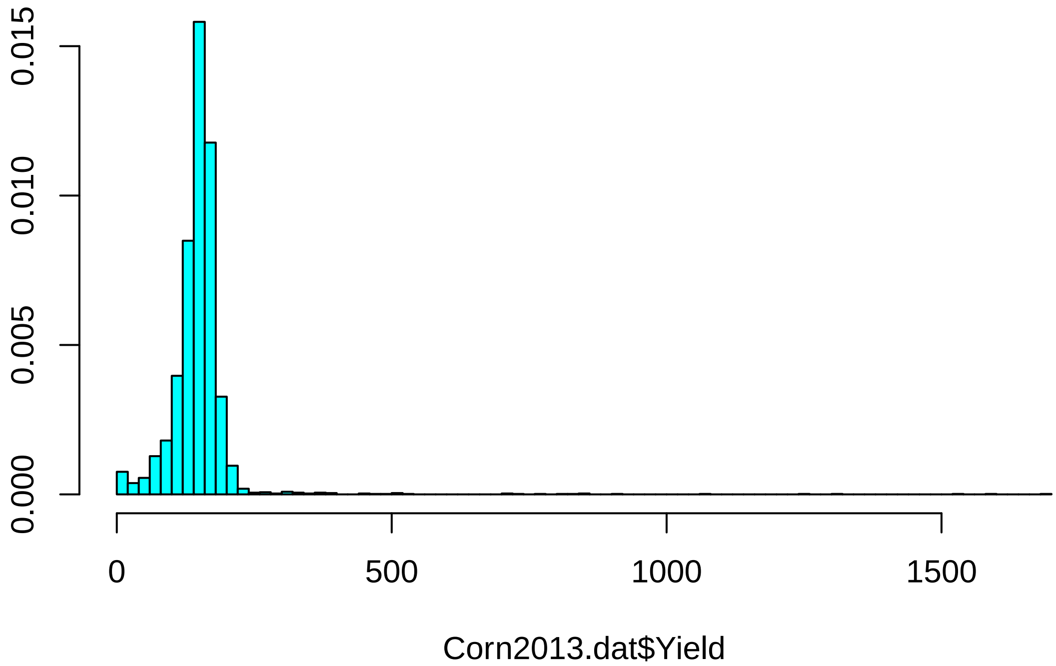



```r
par(mfrow=c(1,3))
plot(Corn2013.dat$Quantile,Corn2013.dat$Yield,xlim=c(0,.05))
plot(Corn2013.dat$Quantile,Corn2013.dat$Yield)
plot(Corn2013.dat$Quantile,Corn2013.dat$Yield,xlim=c(0.98,1))
```

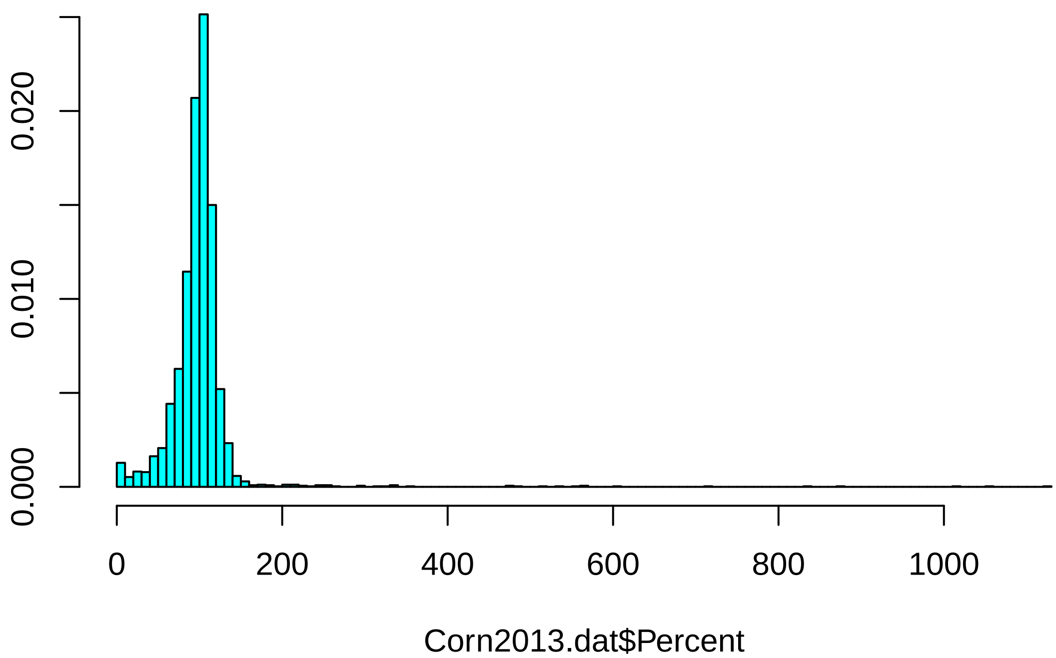



```r
par(mfrow=c(1,1))
```

```r
truehist(Corn2013.dat$Yield)
```




```r
truehist(Corn2013.dat$Percent)
```



We've removed 0, but are there any other duplicates at the minimum rank?

```r
min(Corn2013.dat$Rank)
```

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

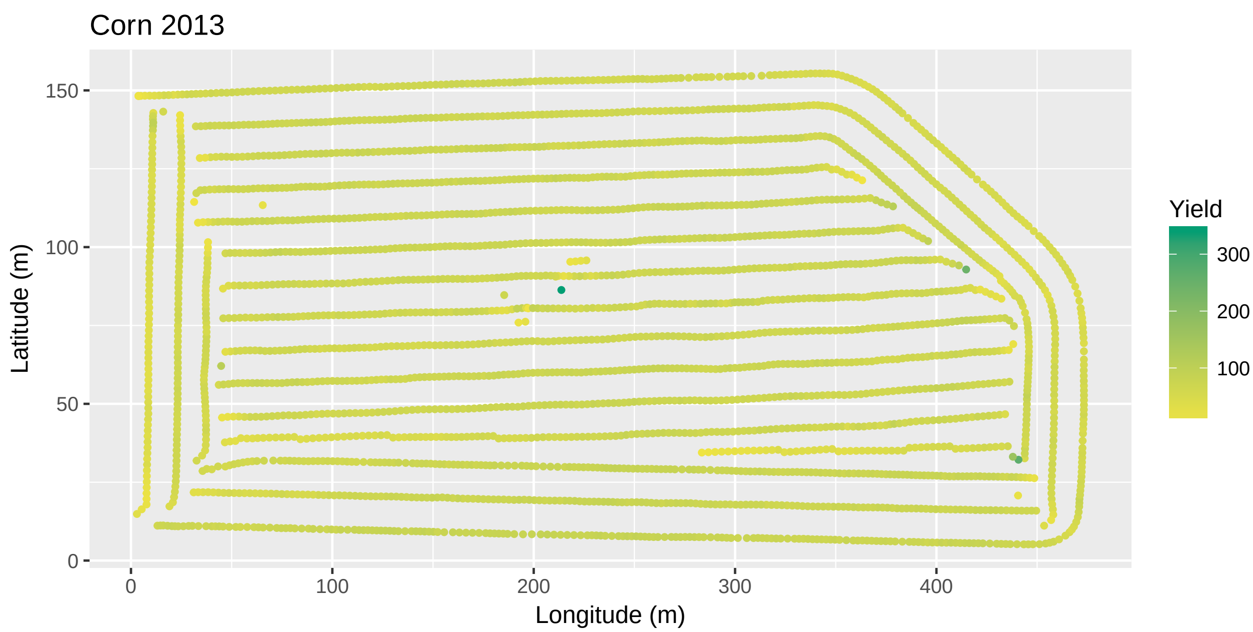
```r
sum(Corn2013.dat$Rank==min(Corn2013.dat$Rank))
```

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

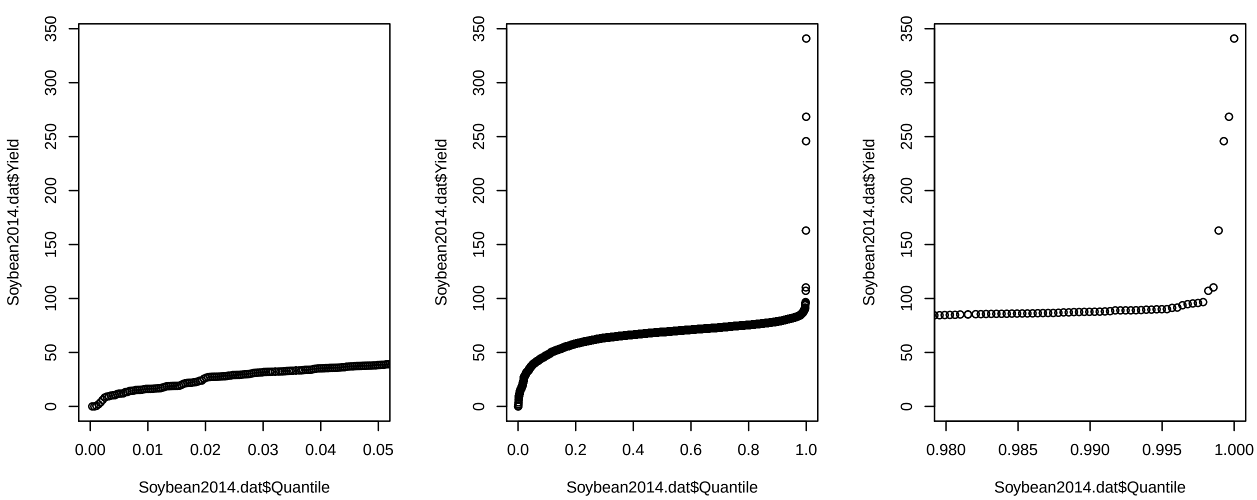
Soybeans, 2014

```
Soybean2014.dat <- standardize.field(Soybean2014.dat,origin)
```

```
ggplot(Soybean2014.dat, aes(Easting,Northing)) +
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 = "Corn 2013")
```




```
par(mfrow=c(1,3))
plot(Soybean2014.dat$Quantile,Soybean2014.dat$Yield,xlim=c(0,.05))
plot(Soybean2014.dat$Quantile,Soybean2014.dat$Yield)
plot(Soybean2014.dat$Quantile,Soybean2014.dat$Yield,xlim=c(0.98,1))
```

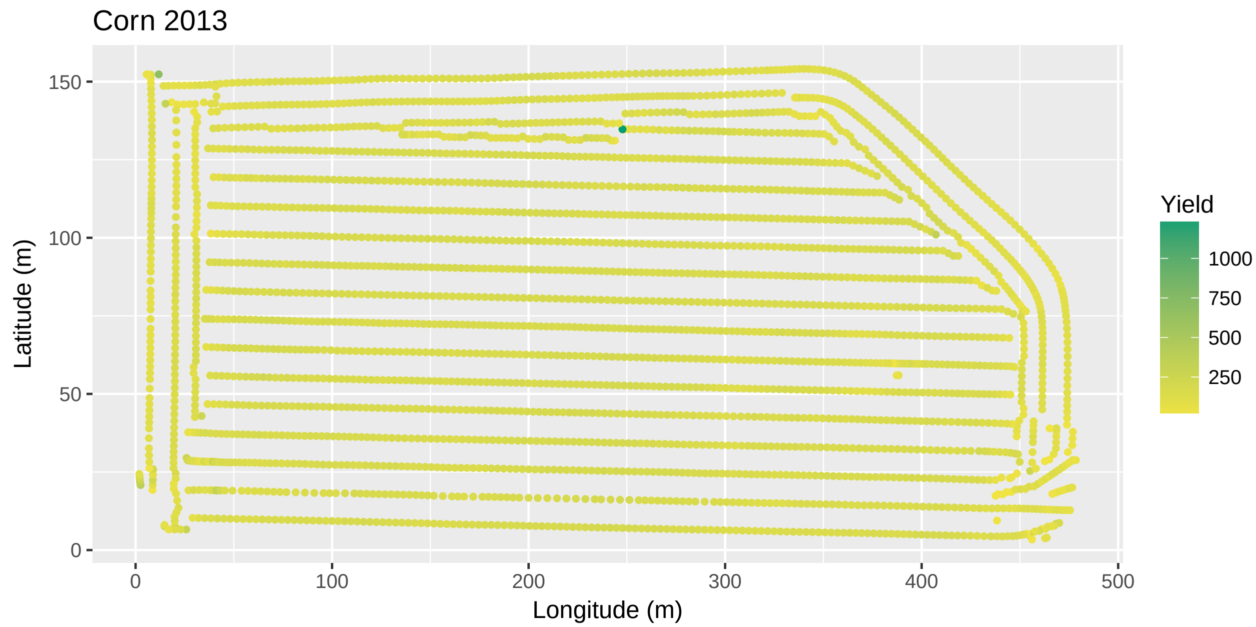



```
par(mfrow=c(1,1))
```

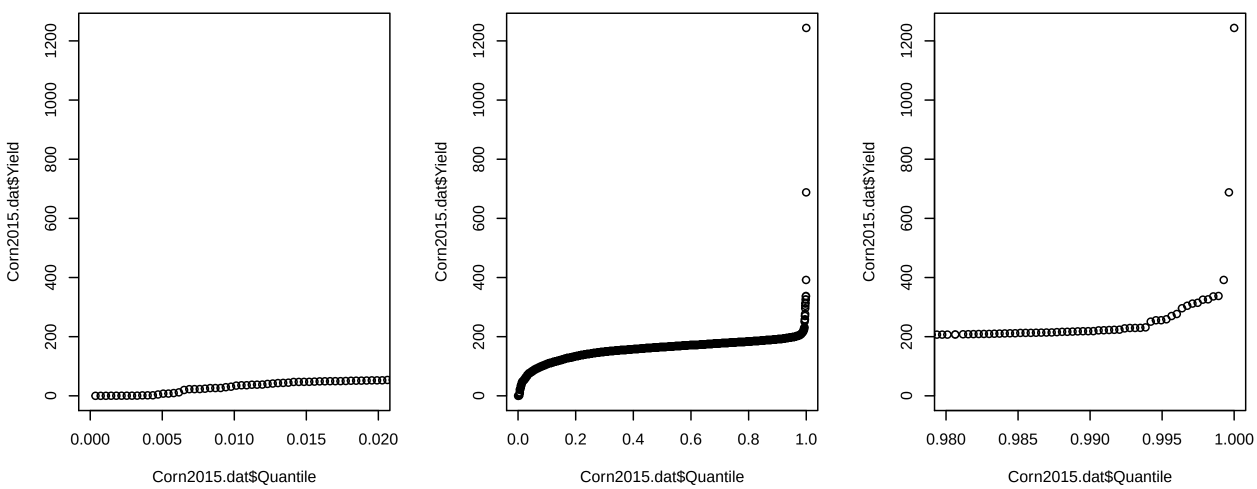
Corn, 2015

```
Corn2015.dat <- standardize.field(Corn2015.dat,origin)
```

```
ggplot(Corn2015.dat, aes(Easting,Northing)) +
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 = "Corn 2013")
```




```
par(mfrow=c(1,3))
plot(Corn2015.dat$Quantile,Corn2015.dat$Yield,xlim=c(0,.02))
plot(Corn2015.dat$Quantile,Corn2015.dat$Yield)
plot(Corn2015.dat$Quantile,Corn2015.dat$Yield,xlim=c(0.98,1))
```

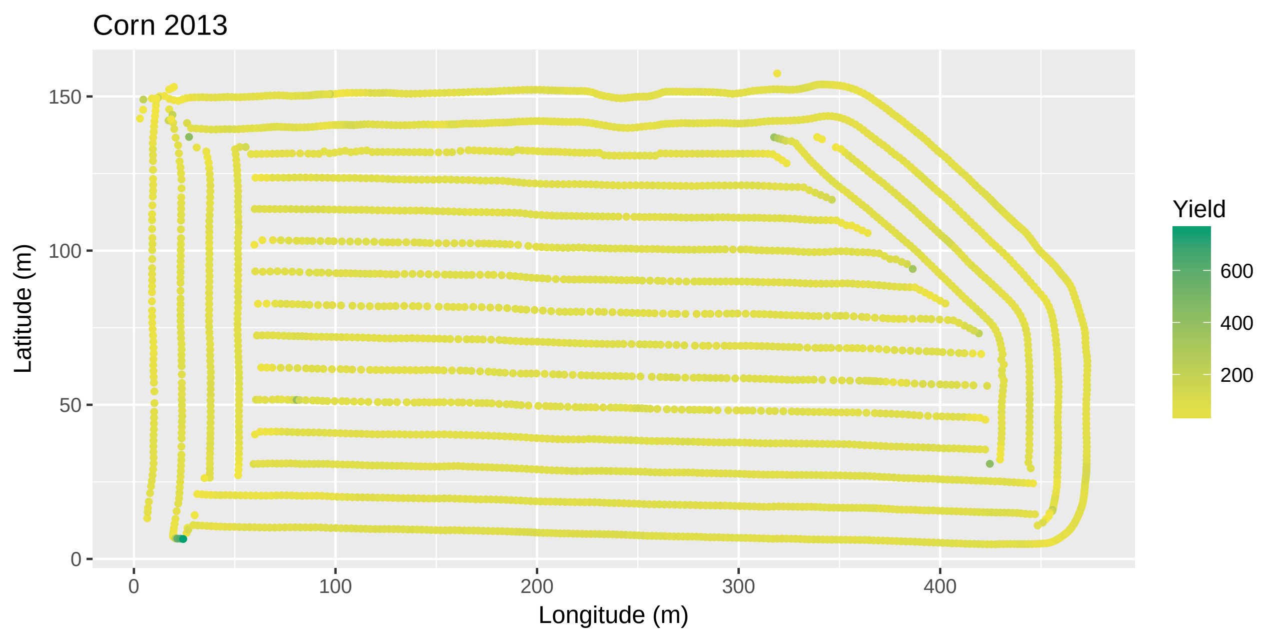



```
par(mfrow=c(1,1))
```

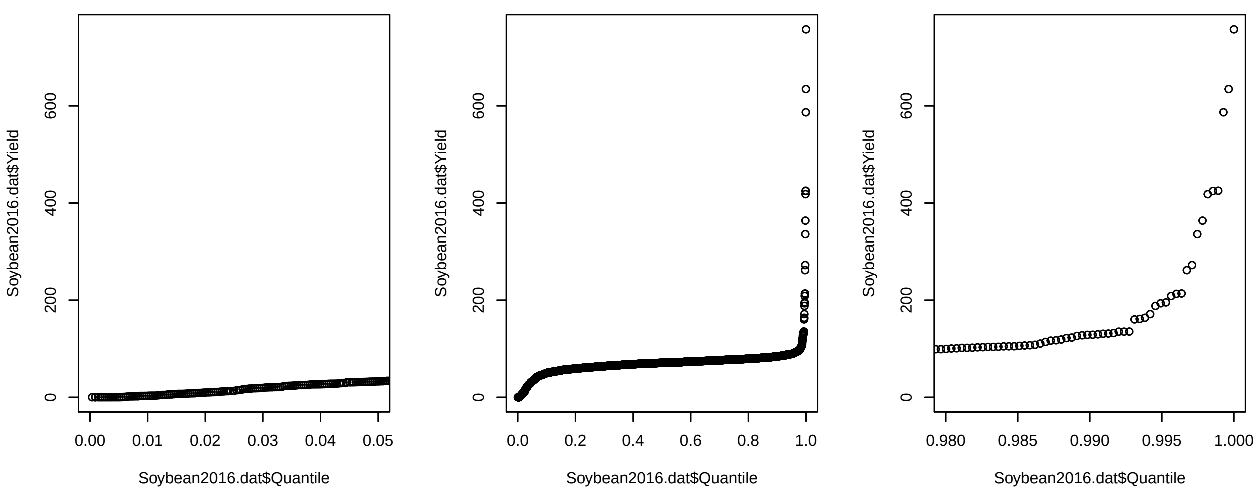
Soybeans, 2016

```
Soybean2016.dat <- standardize.field(Soybean2016.dat,origin)
```

```
ggplot(Soybean2016.dat, aes(Easting,Northing)) +
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 = "Corn 2013")
```




```
par(mfrow=c(1,3))
plot(Soybean2016.dat$Quantile,Soybean2016.dat$Yield,xlim=c(0,.05))
plot(Soybean2016.dat$Quantile,Soybean2016.dat$Yield)
plot(Soybean2016.dat$Quantile,Soybean2016.dat$Yield,xlim=c(0.98,1))
```




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

Spatial Statitics, Uncleaned data

```
summary.spatial <- function(field.dat) {
  #Create Distance Neighbors
  pooled.dists <- as.matrix(dist(cbind(field.dat$Easting, field.dat$Northing)))
  pooled.dists <- 1/pooled.dists
  diag(pooled.dists) <- 0
  #use ape to compute Moran I, it's faster
  require(ape)
  MoranIYield <- Moran.I(field.dat$Yield, pooled.dists)
  MoranIPercent <- Moran.I(field.dat$Percent, pooled.dists)
  MoranIQuantile <- Moran.I(field.dat$Quantile, pooled.dists)

  # for simplicity, correlogram instead of variogram
  require(ncf)
  CorrelogYield <- correlog(field.dat$Easting, field.dat$Northing, field.dat$Yield,
                          increment=3, resamp=0, quiet=TRUE)
  CorrelogPercent <- correlog(field.dat$Easting, field.dat$Northing, field.dat$Percent,
                          increment=3, resamp=0, quiet=TRUE)
  CorrelogQuantile <- correlog(field.dat$Easting, field.dat$Northing, field.dat$Quantile,
                          increment=3, resamp=0, quiet=TRUE)

   return(list(MoranIYield=MoranIYield,
               MoranIPercent=MoranIPercent,
               MoranIQuantile=MoranIQuantile,
               CorrelogYield=CorrelogYield,
               CorrelogPercent=CorrelogPercent,
               CorrelogQuantile=CorrelogQuantile))
}
#override correlog plot to make it easier to modify
plot.correlog <- function (x, main="Correlogram",...)
{
    obj <- x
    plot(obj$mean.of.class, obj$correlation, ylab = "correlation",
        xlab = "distance (mean-of-class)",...)
    lines(obj$mean.of.class, obj$correlation)
    if (!is.null(obj$p)) {
        points(obj$mean.of.class[obj$p < 0.025], obj$correlation[obj$p <
            0.025], pch = 21, bg = "black")
    }
    title(main)
}
```

What spatial variable should we use?

```
Corn2013.summary <- summary.spatial(Corn2013.dat)
```

```
## Loading required package: ape
## Warning: package 'ape' was built under R version 3.3.2
## Loading required package: ncf
```

```
##
## Attaching package: 'ncf'

## The following object is masked _by_ '.GlobalEnv':
##
##     plot.correlog

## The following object is masked from 'package:ape':
##
##     mantel.test
```

```
Corn2013.summary$MoranIYield
```

```
## $observed
## [1] 0.1717702
##
## $expected
## [1] -0.0002907822
##
## $sd
## [1] 0.001038807
##
## $p.value
## [1] 0
```

```
Corn2013.summary$MoranIPercent
```

```
## $observed
## [1] 0.1717702
##
## $expected
## [1] -0.0002907822
##
## $sd
## [1] 0.001038807
##
## $p.value
## [1] 0
```

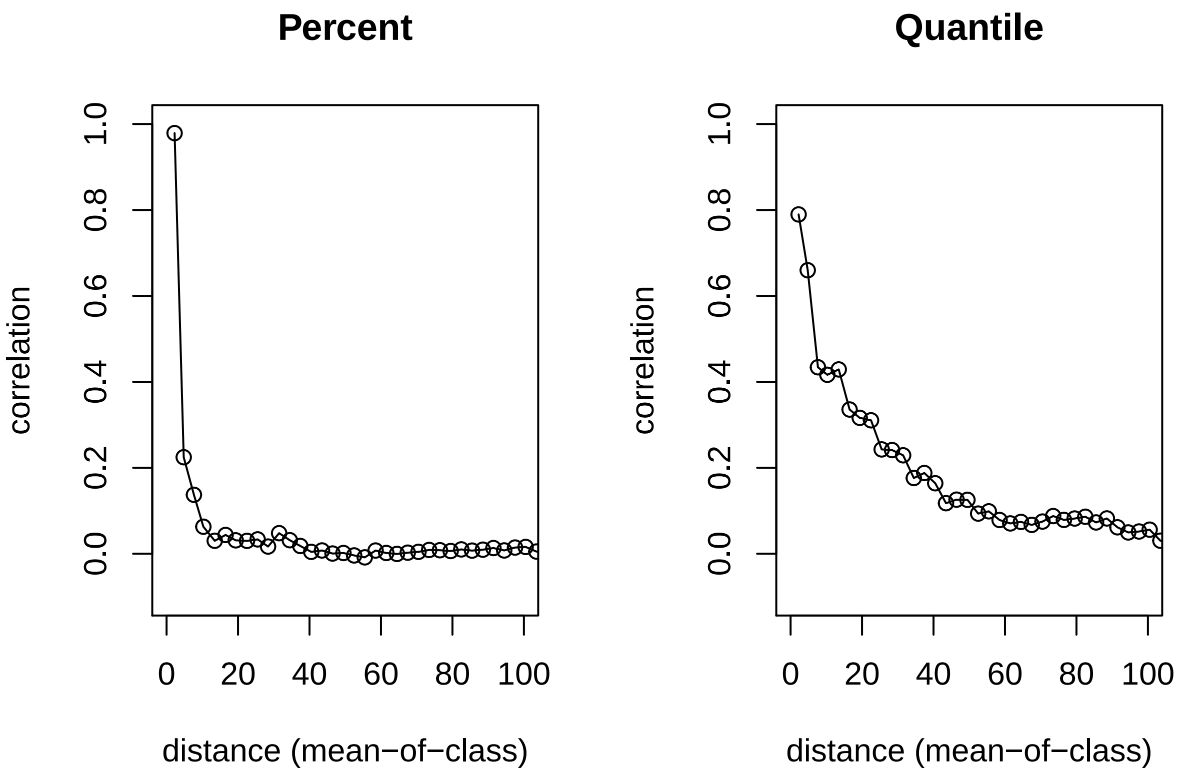
```
Corn2013.summary$MoranIQuantile
```

```
## $observed
## [1] 0.1333456
##
## $expected
## [1] -0.0002907822
##
## $sd
## [1] 0.001065378
##
## $p.value
## [1] 0
```

There is no difference in I between `Yield` and `Percent`, which should be expected. There is a reduction in I with `Quantile`, which may not be a good thing - remember, an I near 0 implies no spatial correlation, and that our data is generated by white noise.

What is the range of correlation, and does it change with measure?

```
par(mfrow=c(1,2))
plot(Corn2013.summary$CorrelogPercent,main='Percent',xlim=c(0,100),ylim=c(-0.1,1))
plot(Corn2013.summary$CorrelogQuantile,main='Quantile',xlim=c(0,100),ylim=c(-0.1,1))
```




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

Is spatial correlation similar for each year?

```
Soybean2014.summary <- summary.spatial(Soybean2014.dat)
Soybean2014.summary$MoranIPercent
```

```
## $observed
## [1] 0.09115397
##
## $expected
## [1] -0.0003589375
##
## $sd
## [1] 0.001082365
##
## $p.value
## [1] 0
```

```
Soybean2014.summary$MoranIQuantile
```

```
## $observed
## [1] 0.129403
```

```
##
## $expected
## [1] -0.0003589375
##
## $sd
## [1] 0.001093114
##
## $p.value
## [1] 0
```

```
Corn2015.summary <- summary.spatial(Corn2015.dat)
Corn2015.summary$MoranIPercent
```

```
## $observed
## [1] 0.1122946
##
## $expected
## [1] -0.0003620565
##
## $sd
## [1] 0.001109414
##
## $p.value
## [1] 0
```

```
Corn2015.summary$MoranIQuantile
```

```
## $observed
## [1] 0.1507307
##
## $expected
## [1] -0.0003620565
##
## $sd
## [1] 0.001139721
##
## $p.value
## [1] 0
```

```
Soybean2016.summary <- summary.spatial(Soybean2016.dat)
Soybean2016.summary$MoranIPercent
```

```
## $observed
## [1] 0.03914923
##
## $expected
## [1] -0.0003633721
##
## $sd
## [1] 0.001042099
##
## $p.value
## [1] 0
```

```
Soybean2016.summary$MoranIQuantile
```

```
## $observed
```

```
## [1] 0.1053377
##
## $expected
## [1] -0.0003633721
##
## $sd
## [1] 0.001077997
##
## $p.value
## [1] 0
```

Soybean 2016 was a problem year, with a lot of *Kochia*. This may have introduced some noise to the yield pattern ($I = 0.039$, versus, $\{0.172, 0.091, 0.112\}$) for the other three years. However, using`Quantile` would reduce the range of I ($\{0.105 - 0.151\}$ versus $\{0.039 - 0.172\}$). It also worth noting that 2013 was planted with two varieties split between the north and south halves. This field showed the largest I at 0.172 for `Percent` but not for `Quantile`. Estimated dry yield averages for the two varieties for 2013 were 149.66 bu/ac vs 145.94 bu/ac, while moisture averaged at 14.31% vs 16.15%.

Other spatial estimates

`spdep` is a very comprehensive package, but I hesitate to use it for some analysis because it can be quite slow (I think there is a tradeoff between flexibility in the data structures and computational speed). We'll compare different measures for Soybean 2016 to see if they're worth the extra computing time.

```
pooled.dists <- as.matrix(dist(cbind(Soybean2016.dat$Easting, Soybean2016.dat$Northing)))
pooled.dists <- 1/pooled.dists
diag(pooled.dists) <- 0
library(spdep)
```

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

## Loading required package: sp

## Warning: package 'sp' was built under R version 3.3.2

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.3.2

##
## Attaching package: 'spdep'

## The following object is masked from 'package:ape':
##
##     plot.mst
```

```
pooled.dists.listw <-mat2listw(pooled.dists)
```

```
neighborhood30 <- dnearneigh(cbind(Soybean2016.dat$Easting, Soybean2016.dat$Northing), 0, 30)
neighborhood30.listw <- nb2listw(neighborhood30, style="B")
```

```
moran.test(Soybean2016.dat$Yield,pooled.dists.listw,alternative = "two.sided")
```

```
##
##  Moran I test under randomisation
##
## data:  Soybean2016.dat$Yield
## weights: pooled.dists.listw
##
```

```
## Moran I statistic standard deviate = 22.305, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      2.968512e-02     -3.633721e-04      1.814881e-06
```

```r
moran.test(Soybean2016.dat$Quantile,pooled.dists.listw,alternative = "two.sided")
```

```
##
##  Moran I test under randomisation
##
## data:  Soybean2016.dat$Quantile
## weights: pooled.dists.listw
##
## Moran I statistic standard deviate = 72.053, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      1.000421e-01     -3.633721e-04      1.941824e-06
```

```r
geary.test(Soybean2016.dat$Yield,pooled.dists.listw,alternative = "two.sided")
```

```
##
##  Geary C test under randomisation
##
## data:  Soybean2016.dat$Yield
## weights: pooled.dists.listw
##
## Geary C statistic standard deviate = 5.284, p-value = 1.264e-07
## alternative hypothesis: two.sided
## sample estimates:
## Geary C statistic       Expectation          Variance
##      0.8761069025      1.0000000000      0.0005497563
```

```r
geary.test(Soybean2016.dat$Quantile,pooled.dists.listw,alternative = "two.sided")
```

```
##
##  Geary C test under randomisation
##
## data:  Soybean2016.dat$Quantile
## weights: pooled.dists.listw
##
## Geary C statistic standard deviate = 55.619, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
## Geary C statistic       Expectation          Variance
##      8.838228e-01      1.000000e+00      4.363177e-06
```

```r
globalG.test(Soybean2016.dat$Yield,neighborhood30.listw, alternative = "two.sided")
```

```
##
##  Getis-Ord global G statistic
##
## data:  Soybean2016.dat$Yield
## weights: neighborhood30.listw
##
## standard deviate = 1.6583, p-value = 0.09726
```

```
## alternative hypothesis: two.sided
## sample estimates:
## Global G statistic        Expectation           Variance
##       3.483265e-02       3.466277e-02       1.049553e-08
```

```
globalG.test(Soybean2016.dat$Quantile,neighborhood30.listw, alternative = "two.sided")
```

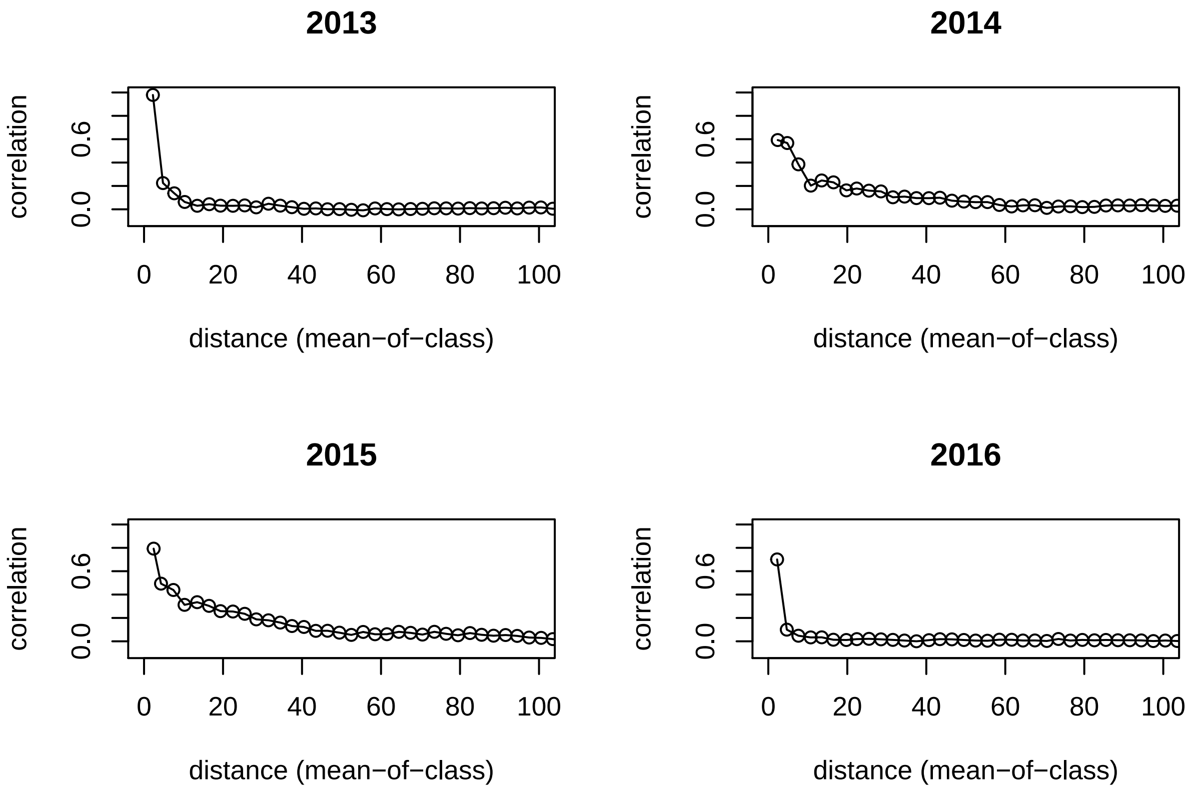
```
##
##  Getis-Ord global G statistic
##
## data:  Soybean2016.dat$Quantile
## weights: neighborhood30.listw
##
## standard deviate = 19.426, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
## Global G statistic        Expectation           Variance
##       3.727472e-02       3.466277e-02       1.807910e-08
```

Correlograms, Percent

```
head(Corn2013.summary$CorrelogPercent$p)
```

```
## NULL
```

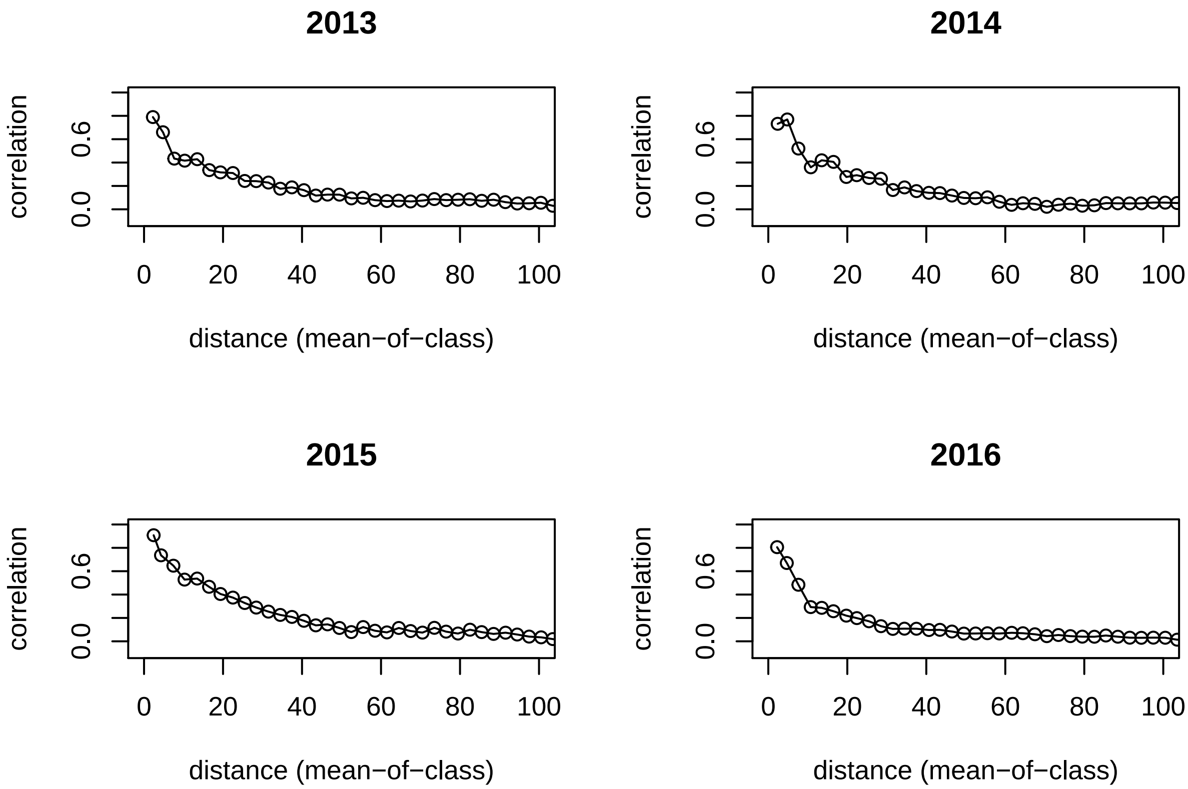
```
par(mfrow=c(2,2))
plot(Corn2013.summary$CorrelogPercent,main='2013',xlim=c(0,100),ylim=c(-0.1,1))
plot(Soybean2014.summary$CorrelogPercent,main='2014',xlim=c(0,100),ylim=c(-0.1,1))
plot(Corn2015.summary$CorrelogPercent,main='2015',xlim=c(0,100),ylim=c(-0.1,1))
plot(Soybean2016.summary$CorrelogPercent,main='2016',xlim=c(0,100),ylim=c(-0.1,1))
```




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

Correlograms, Quantiles

```
par(mfrow=c(2,2))
plot(Corn2013.summary$CorrelogQuantile,main='2013',xlim=c(0,100),ylim=c(-0.1,1))
plot(Soybean2014.summary$CorrelogQuantile,main='2014',xlim=c(0,100),ylim=c(-0.1,1))
plot(Corn2015.summary$CorrelogQuantile,main='2015',xlim=c(0,100),ylim=c(-0.1,1))
plot(Soybean2016.summary$CorrelogQuantile,main='2016',xlim=c(0,100),ylim=c(-0.1,1))
```

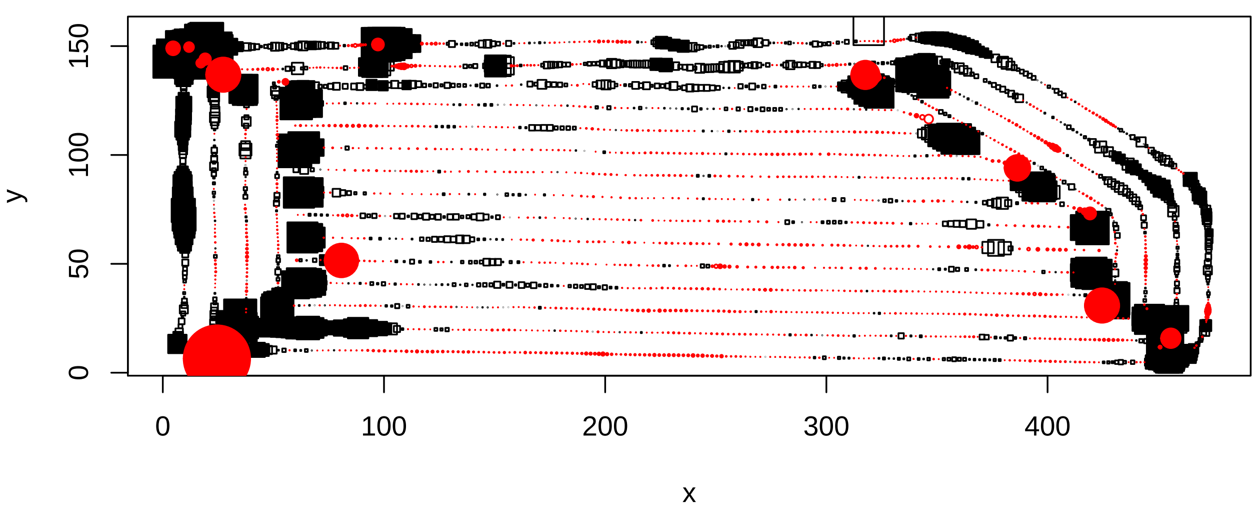



```
par(mfrow=c(1,1))
```

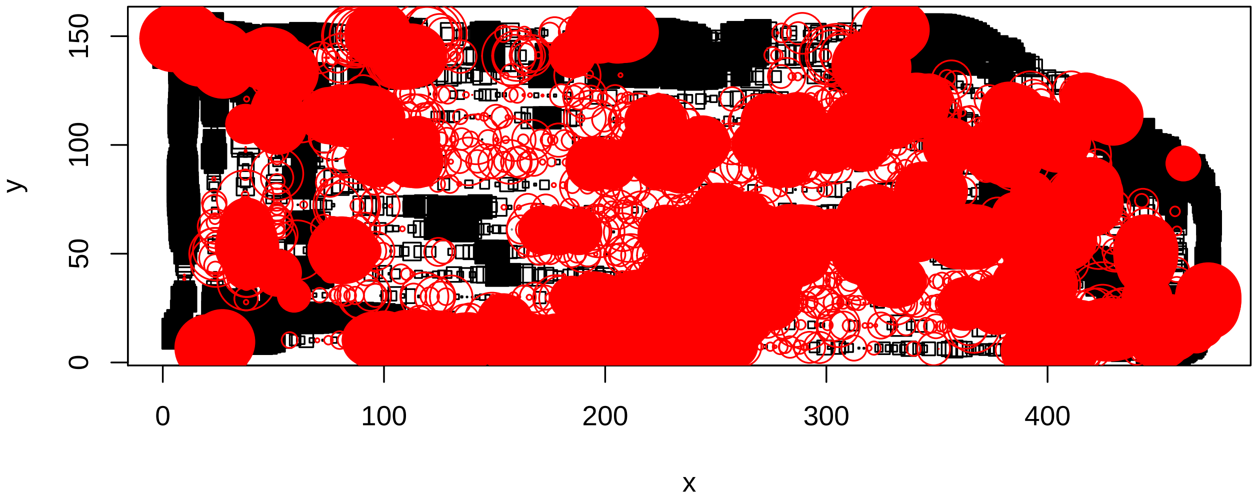
LISA Plots

The 2016 field seems to be the most problematic, so we compare LISA plots for `Percent` and `Quantile` at 10m and 30m neighborhoods

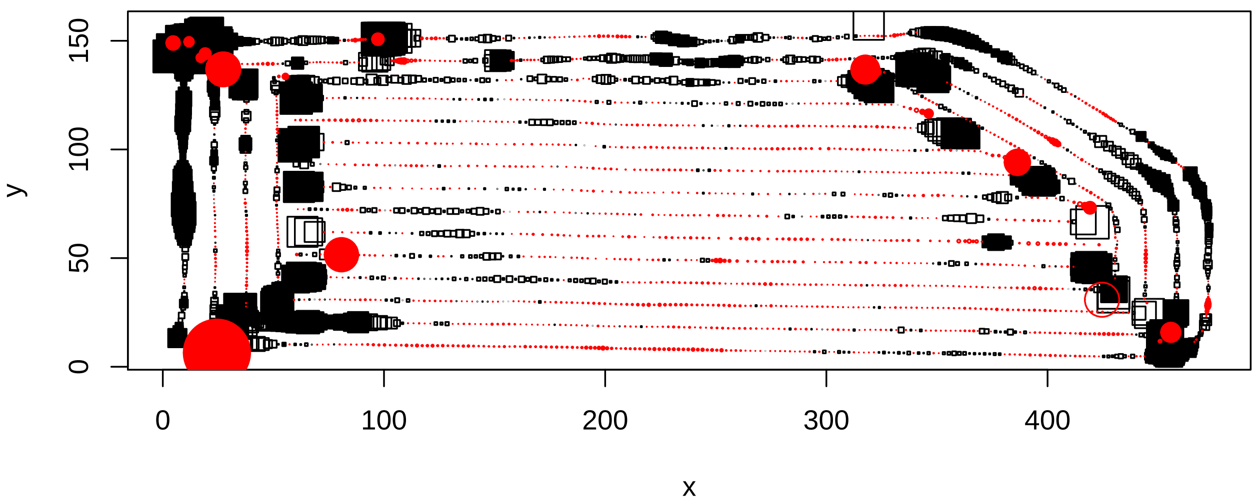
```
Soybean2016P10.lisa <- lisa(Soybean2016.dat$Easting, Soybean2016.dat$Northing, Soybean2016.dat$Percent,
                            neigh=10, resamp=500, quiet=TRUE)
plot.lisa(Soybean2016P10.lisa, negh.mean=FALSE)
```



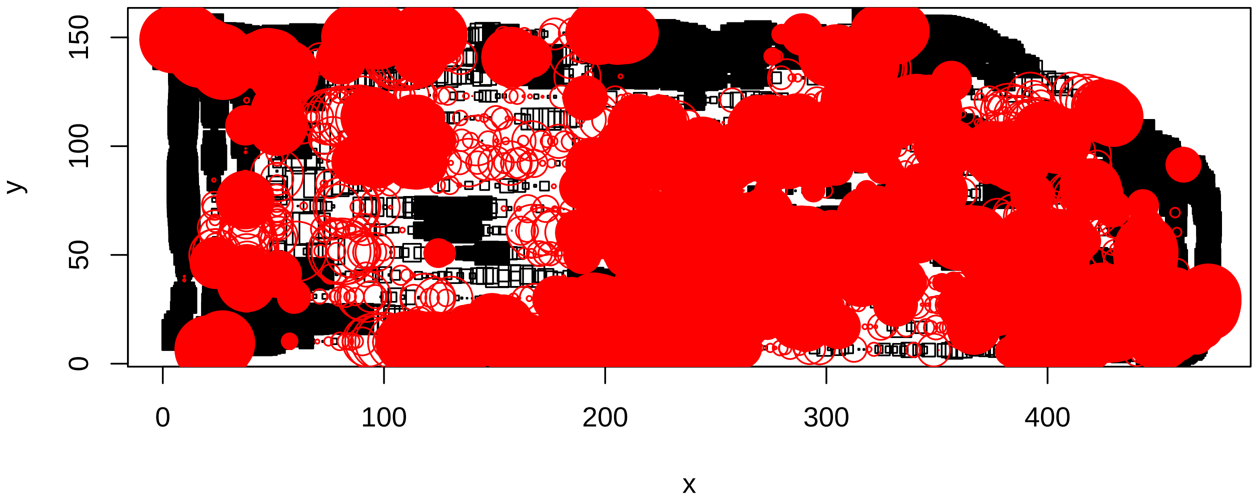
```r
Soybean2016Q10.lisa <- lisa(Soybean2016.dat$Easting, Soybean2016.dat$Northing, Soybean2016.dat$Quantile
                         neigh=10, resamp=500, quiet=TRUE)
plot.lisa(Soybean2016Q10.lisa, negh.mean=FALSE)
```



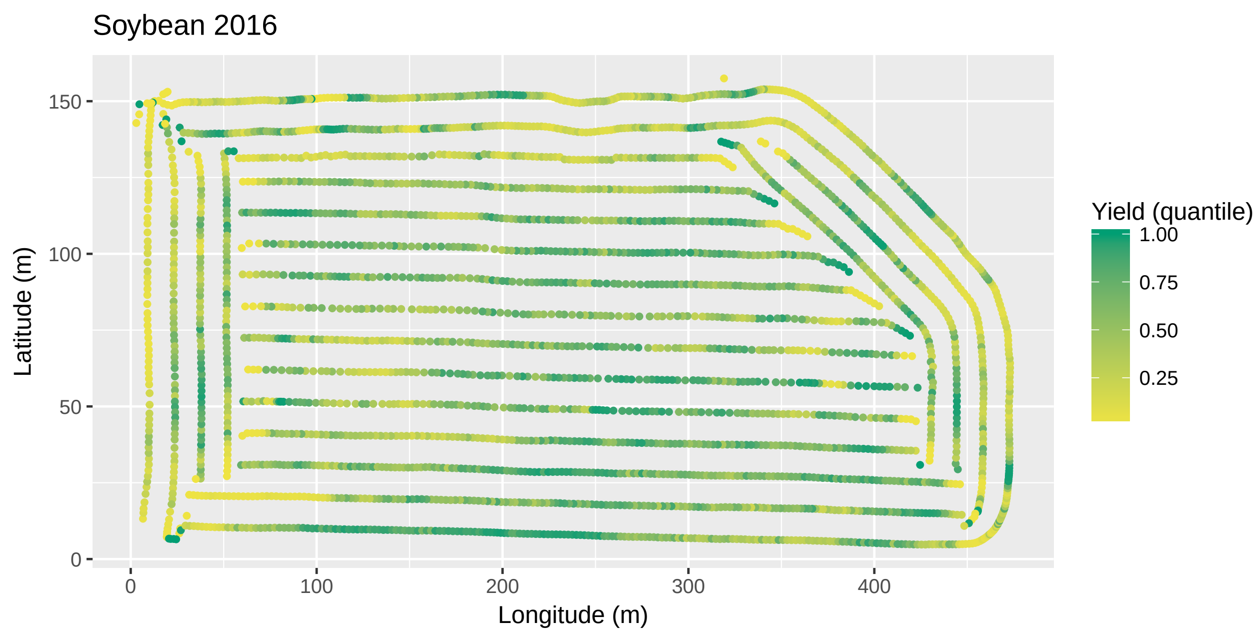

```r
Soybean2016P30.lisa <- lisa(Soybean2016.dat$Easting, Soybean2016.dat$Northing, Soybean2016.dat$Percent,
                         neigh=30, resamp=500, quiet=TRUE)
plot.lisa(Soybean2016P30.lisa, negh.mean=FALSE)
```




```r
Soybean2016Q30.lisa <- lisa(Soybean2016.dat$Easting, Soybean2016.dat$Northing, Soybean2016.dat$Quantile
                         neigh=30, resamp=500, quiet=TRUE)
plot.lisa(Soybean2016Q30.lisa, negh.mean=FALSE)
```

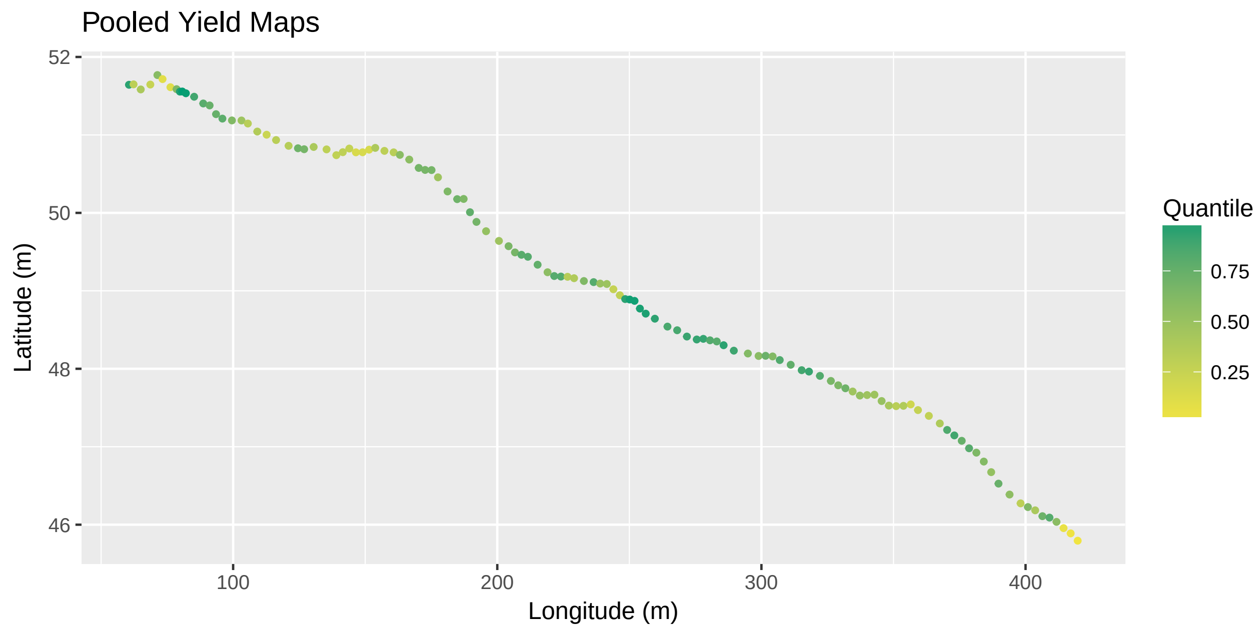



```r
ggplot(Soybean2016.dat, aes(Easting,Northing)) +
geom_point(aes(colour = Quantile),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Yield (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Soybean 2016")
```

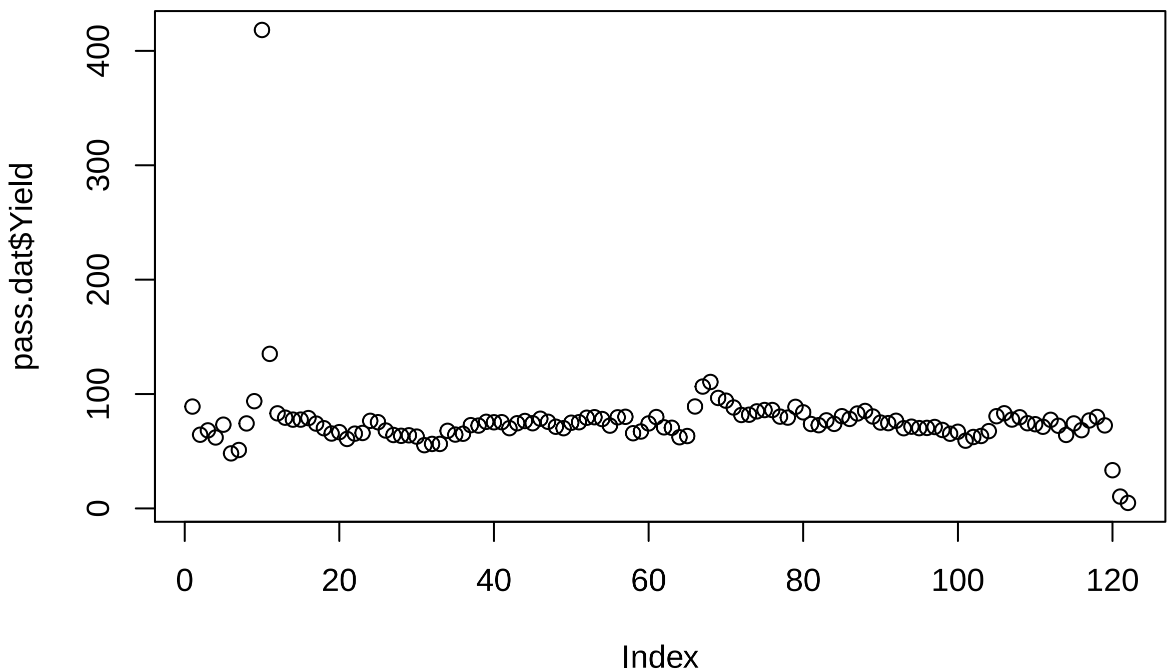



```r
pass.dat <- subset(Soybean2016.dat,Soybean2016.dat$Northing<55)
pass.dat <- subset(pass.dat,pass.dat$Northing>45)
pass.dat <- subset(pass.dat,pass.dat$Easting>55)
pass.dat <- subset(pass.dat,pass.dat$Easting<420)
```

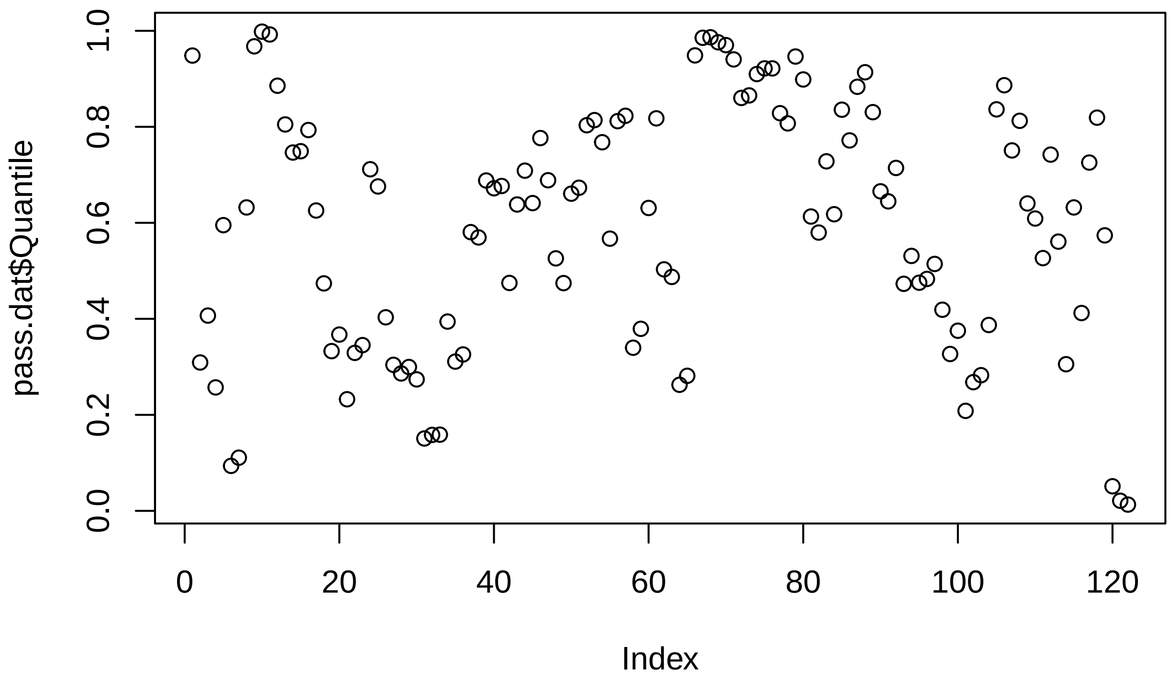
```r
ggplot(pass.dat, aes(Easting,Northing)) +
geom_point(aes(colour = Quantile),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Quantile", x="Longitude (m)", y="Latitude (m)", title = "Pooled Yield Maps")
```



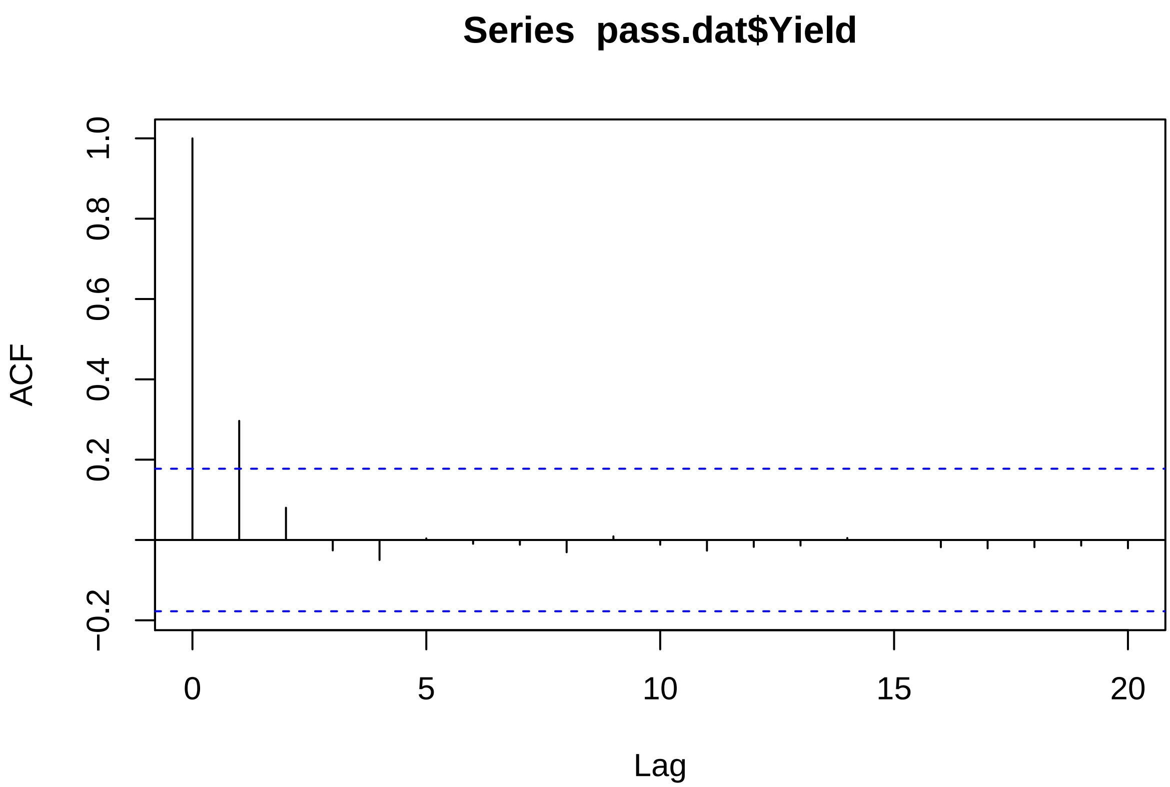

```r
plot(pass.dat$Yield)
```



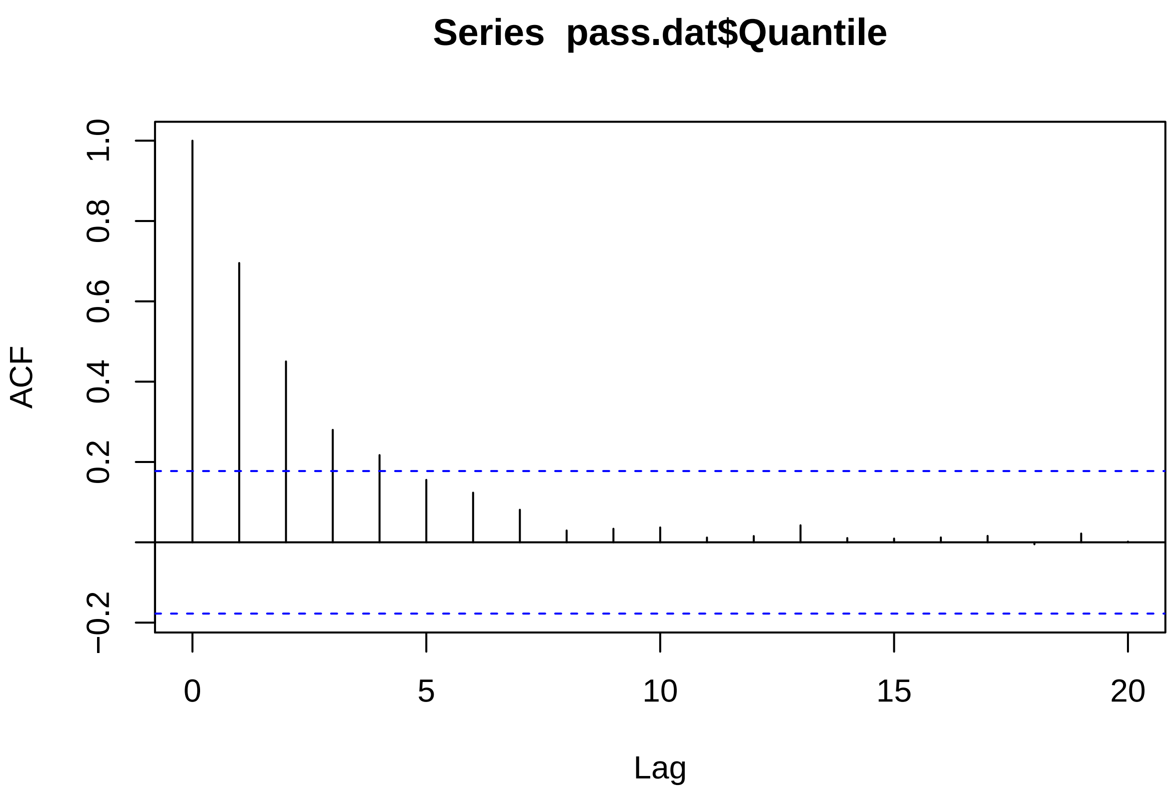

```r
plot(pass.dat$Quantile)
```




```r
acf(pass.dat$Yield)
```




```r
acf(pass.dat$Quantile)
```



What is the best fitting autoregressive model, along a single pass?

```
library(forecast)
```

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

```
auto.arima(pass.dat$Yield,stepwise=FALSE,approximation=FALSE)
```

```
## Series: pass.dat$Yield
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
##       0.3053  75.7815
## s.e.  0.0875   4.2438
##
## sigma^2 estimated as 1085:  log likelihood=-598.52
## AIC=1203.03   AICc=1203.23   BIC=1211.44
```

```
mean(pass.dat$Yield)
```

```
## [1] 75.98918
```

```
auto.arima(pass.dat$Quantile,stepwise=FALSE,approximation=FALSE)
```

```
## Series: pass.dat$Quantile
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.7324  0.5889
## s.e.  0.0641  0.0585
```

```
##
## sigma^2 estimated as 0.0317:  log likelihood=38.05
## AIC=-70.11   AICc=-69.9   BIC=-61.69
```

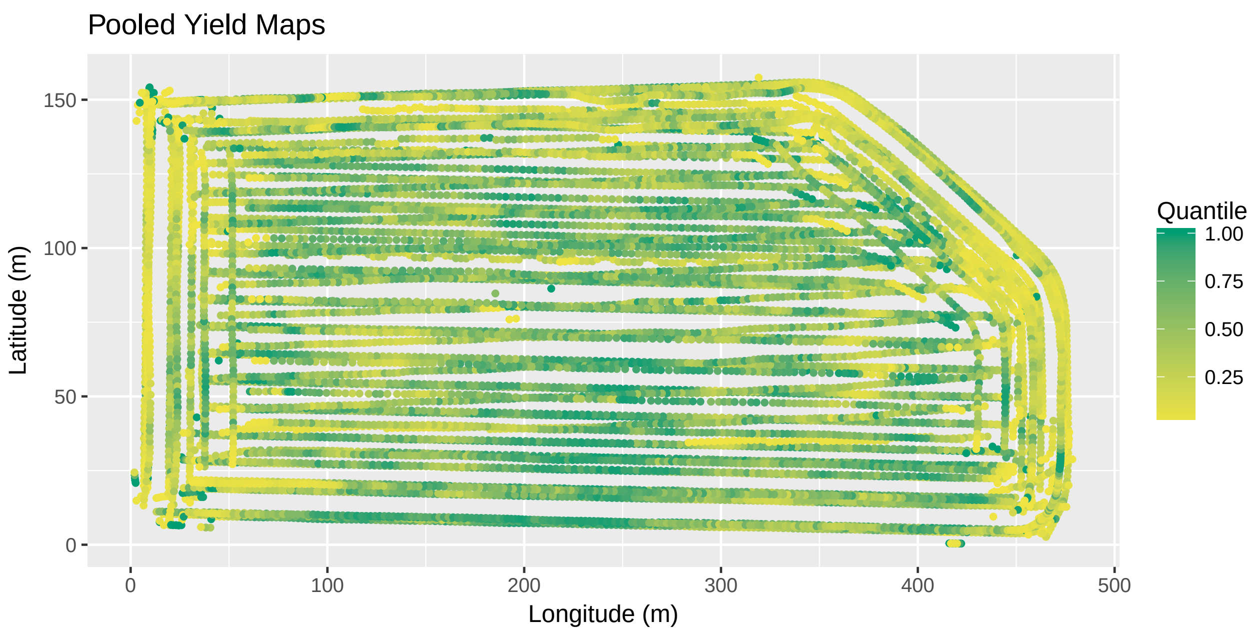
```
mean(pass.dat$Quantile)
```

```
## [1] 0.5935835
```

```
Corn2013.dat$Year <- 2013
Corn2015.dat$Year <- 2015
Soybean2014.dat$Year <- 2014
Soybean2016.dat$Year <- 2016
```

```
Pooled.dat <- rbind(Corn2013.dat,Corn2015.dat,Soybean2014.dat,Soybean2016.dat)
```

```
ggplot(Pooled.dat, aes(Easting,Northing)) +
geom_point(aes(colour = Quantile),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
labs(colour = "Quantile", x="Longitude (m)", y="Latitude (m)", title = "Pooled Yield Maps")
```




```
save(Corn2013.dat, Corn2015.dat, Soybean2014.dat ,Soybean2016.dat,file="Pooled.Rda")
save(Corn2013.summary, Soybean2014.summary, Corn2015.summary, Soybean2016.summary,file="Summaries.Rda")
save(Soybean2016P10.lisa,
     Soybean2016Q10.lisa,
     Soybean2016P30.lisa,
     Soybean2016Q30.lisa,file="LISA.Rda")
```