# Grid Cells

*Peter Claussen*

*9/5/2017*

## Libraries

```
library(MASS)
```

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

```
library(ncf)
library(ggplot2)
```

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

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#0072B2", "#D55E00", "#F0E442","#CC79A7","#0
```

## Data

```
load(file="sample.dat.Rda")
```

Here, we compare three methods for estimating yield over a fixed network of cells.

### Grid Cell Means

We divide our yield map into equally sized, square cells, of 20x20m and calculate simple means of all yield monitor estimates within the bounds of each cell.

### Trend Surface Interpolation.

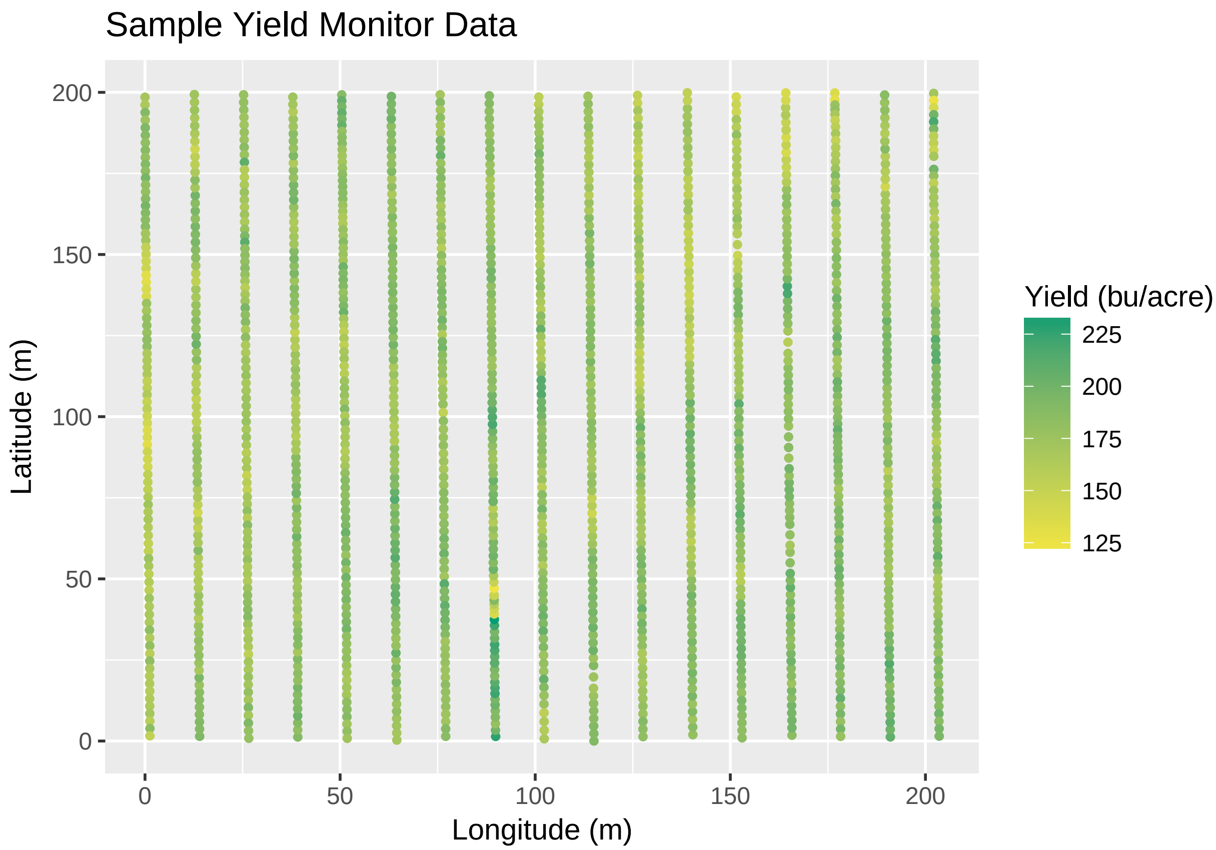
We fit our yeild map to a two-dimensional polynomial and then predict yield at fixed, unsampled locations within a 20x20m network. We will interpolate four points in each cell and compute an average for the points to provide an estimate of yield for each grid cell.

### Kriging

We fit a variogram model and krige to impute yield at fixed locations within the 20x20m network. As with Trend Surface Interpolation, we calculate an average of four kriged points to estimate yield for each grid cell.

Start with map data.

```
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 (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Sample Yield Monitor Data
```




We'll divide the field into 20 m quadrats. I add 0.1 to push min(LatM) and min(LonM) into a grid cell, instead of being on the border (Row 0 or Column 0). This will keep max less than the boundary.

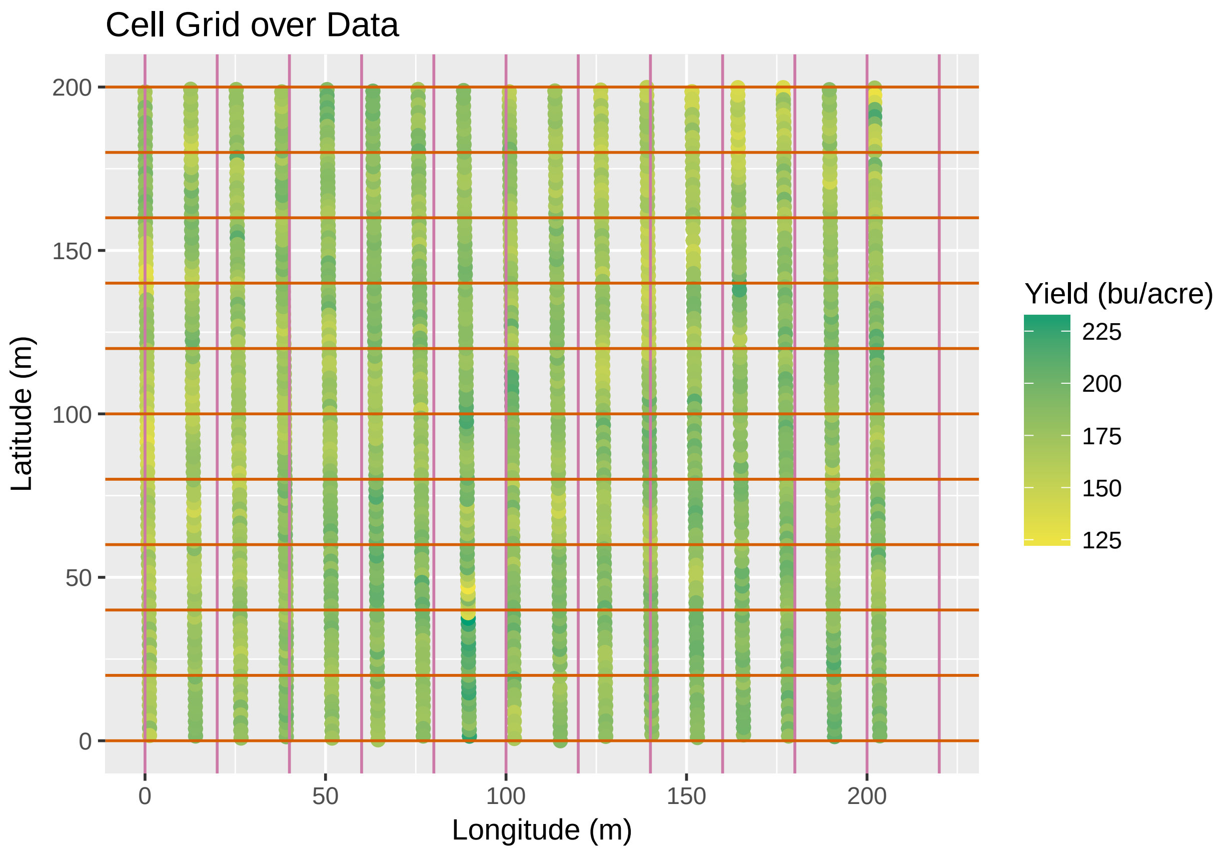
### Cell Indexes

```
sample.dat$row <- ceiling((sample.dat$LatM+0.1)/20)
sample.dat$col <- ceiling((sample.dat$LonM+0.1)/20)
sample.dat$cell <- as.factor(sample.dat$row):as.factor(sample.dat$col)
```

### Cell Borders

```
rowPoints <- c(0,20*(1:ceiling(max(sample.dat$LatM+0.1)/20)))
colPoints <- c(0,20*(1:ceiling(max(sample.dat$LonM+0.1)/20)))
```

```
library(ggplot2)
ggplot(sample.dat, aes(LonM, LatM)) +
  geom_point(aes(colour = Yield),size=2) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Cell Grid over Data")
```

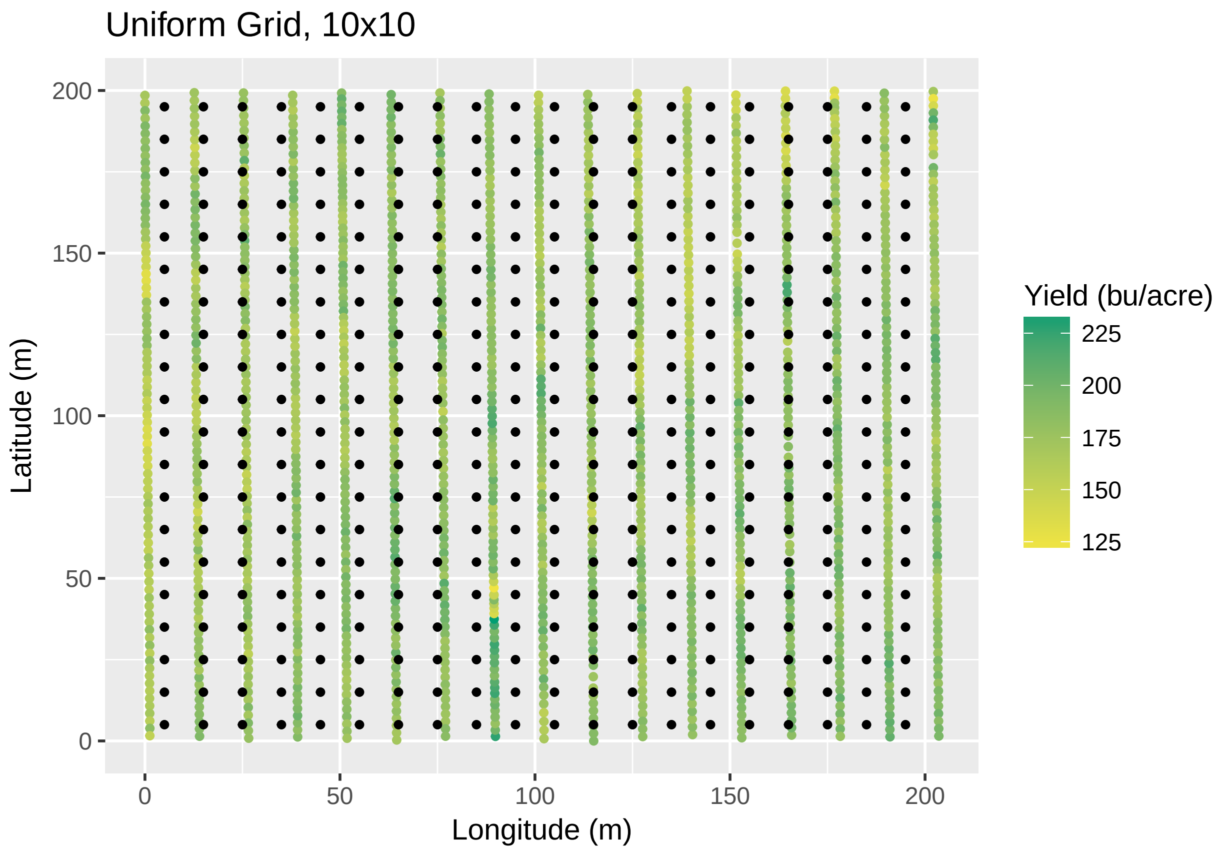



## Sample Grid

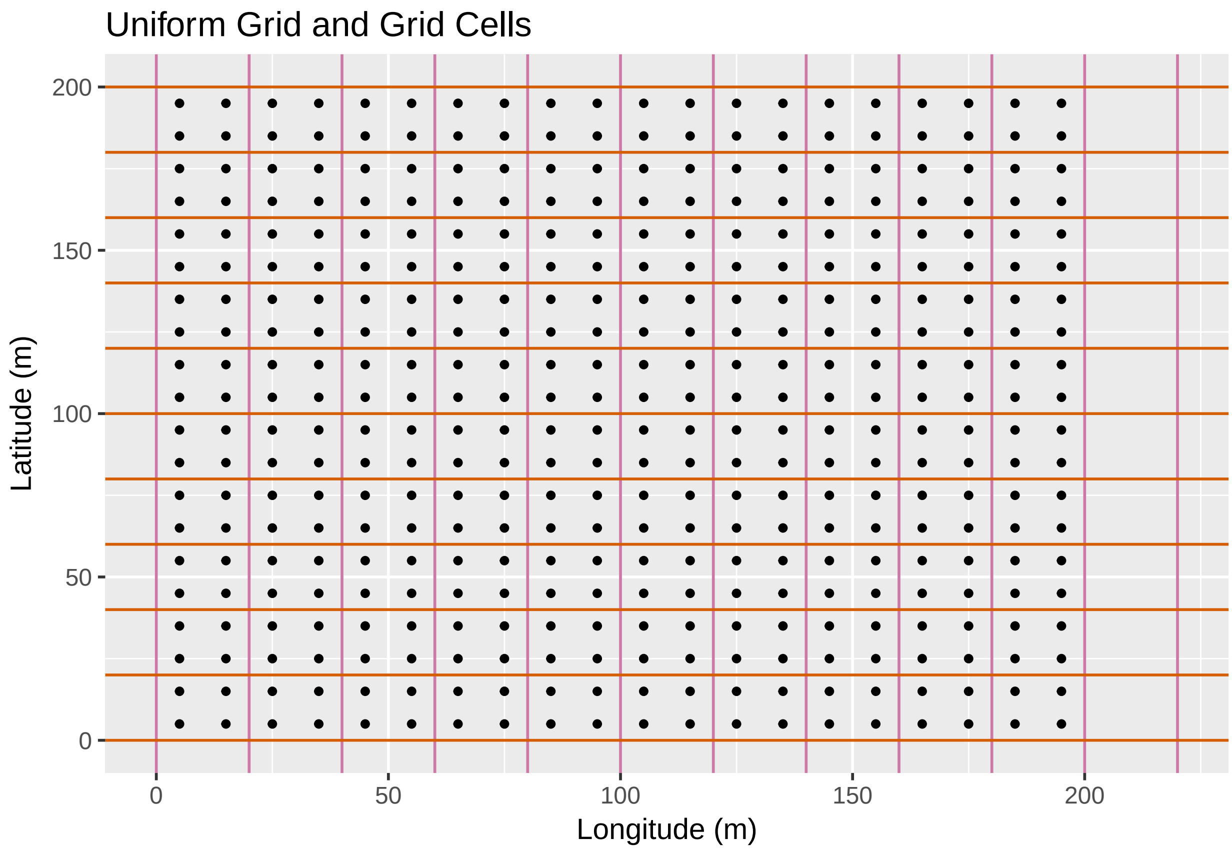
We will need a set of fixed points for interpolation. Since we have a 20x20m grid, we place points at 10m intervals, and offset by 5m from the origin.

```
columns <- floor(max(sample.dat$LonM)/10)
rows <- ceiling(max(sample.dat$LatM)/10)
sample4.grd <- expand.grid(LatM=((1:rows)*10)-5,
                           LonM=((1:columns)*10)-5)
sample4.grd$row <- ceiling((sample4.grd$LatM-5+0.1)/20)
sample4.grd$col <- ceiling((sample4.grd$LonM-5+0.1)/20)
sample4.grd$cell <- as.factor(sample4.grd$row):as.factor(sample4.grd$col)
```

```
ggplot(sample.dat, aes(LonM, LatM)) +
  geom_point(aes(colour = Yield),size=1) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_point(aes(LonM, LatM),data=sample4.grd, size=1) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Uniform Grid, 10x10")
```




```r
ggplot(sample.dat, aes(LonM, LatM)) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_point(aes(LonM, LatM),data=sample4.grd, size=1) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Uniform Grid and Grid C
```




# Kriging

### Variogram

We use a variogram from earlier,

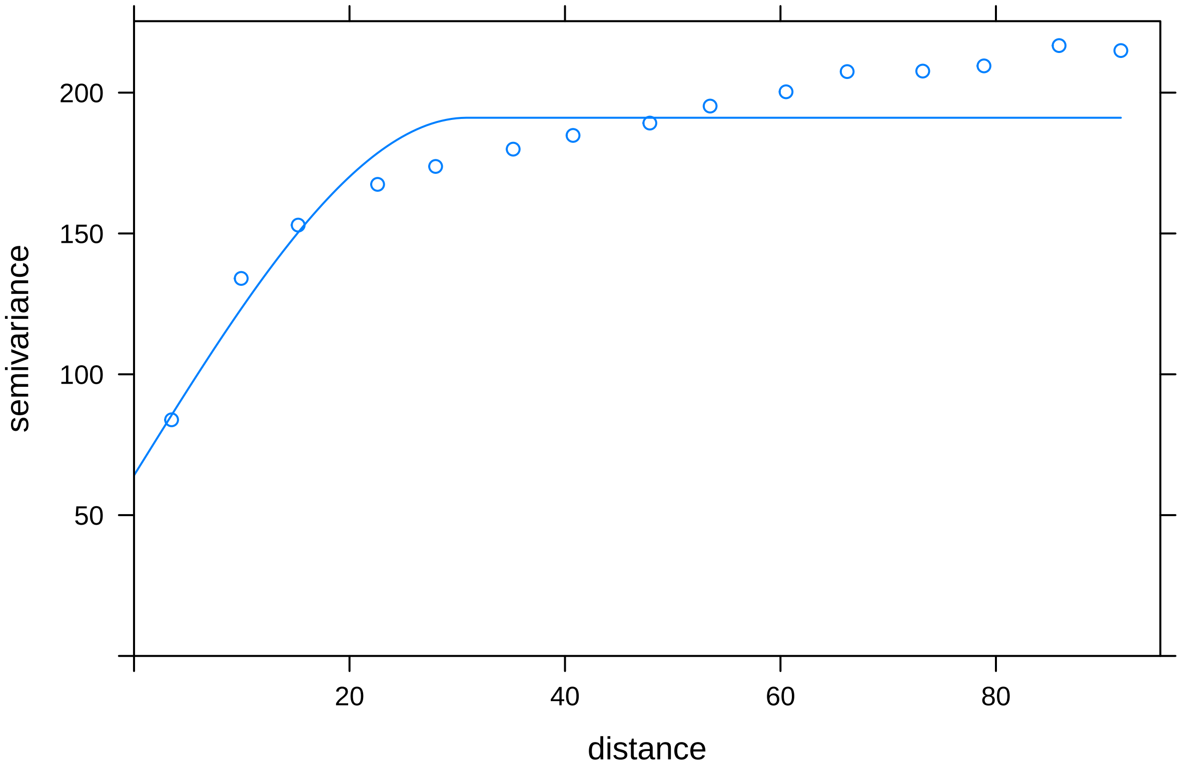
```
library(gstat)
```

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

```
Yield.var <- variogram(Yield~1,
                       locations=~LonM+LatM,
                       data=sample.dat)
print(Yield.vgm <- fit.variogram(Yield.var, vgm(200,"Sph",40,50)))
```

```
##   model     psill    range
## 1   Nug  64.12461  0.00000
## 2   Sph 126.97065 30.91816
```

```
plot(Yield.var, model=Yield.vgm)
```



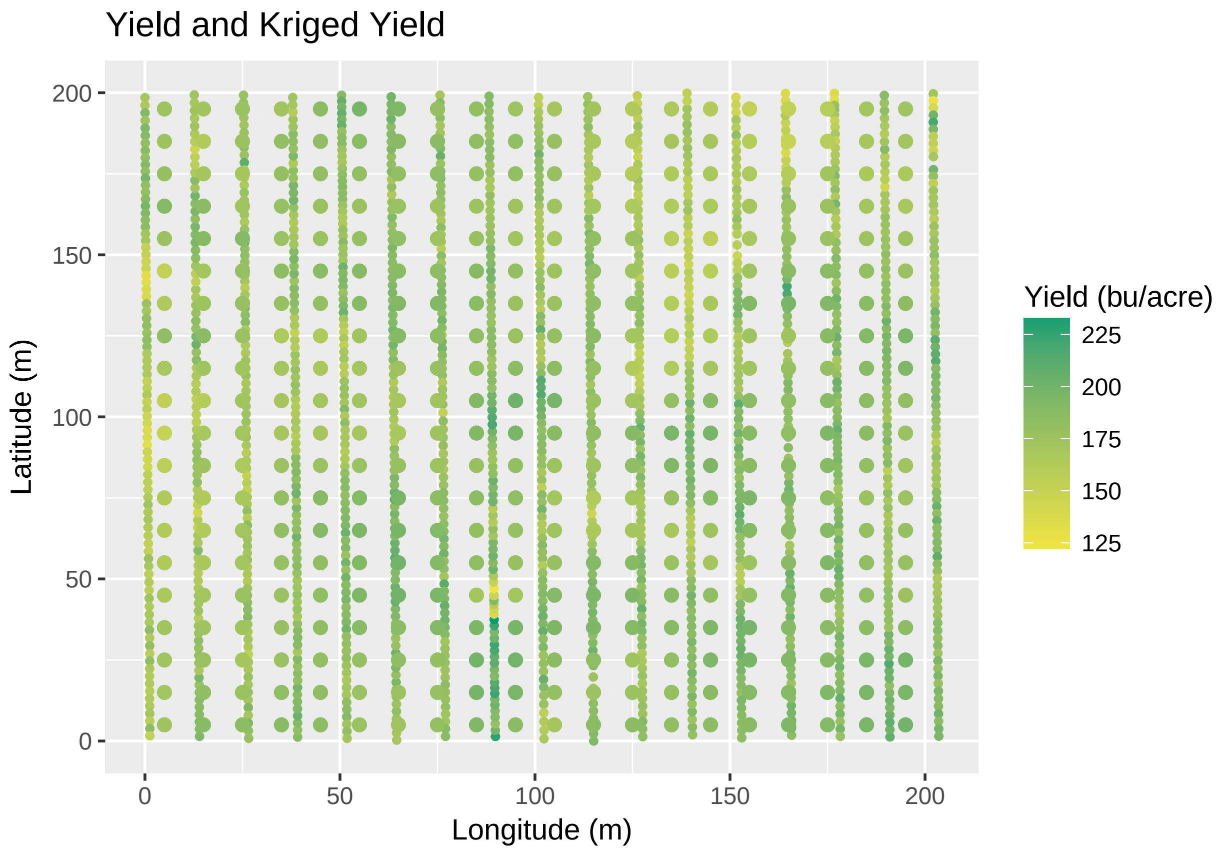

```
Yield.krig <- krige(id="Yield",
                     formula=Yield~1,
                     data = sample.dat,
                     newdata = sample4.grd,
                     model = Yield.vgm,
                     maxdist = 100,
  locations=~LatM + LonM)
```

```
## [using ordinary kriging]
```

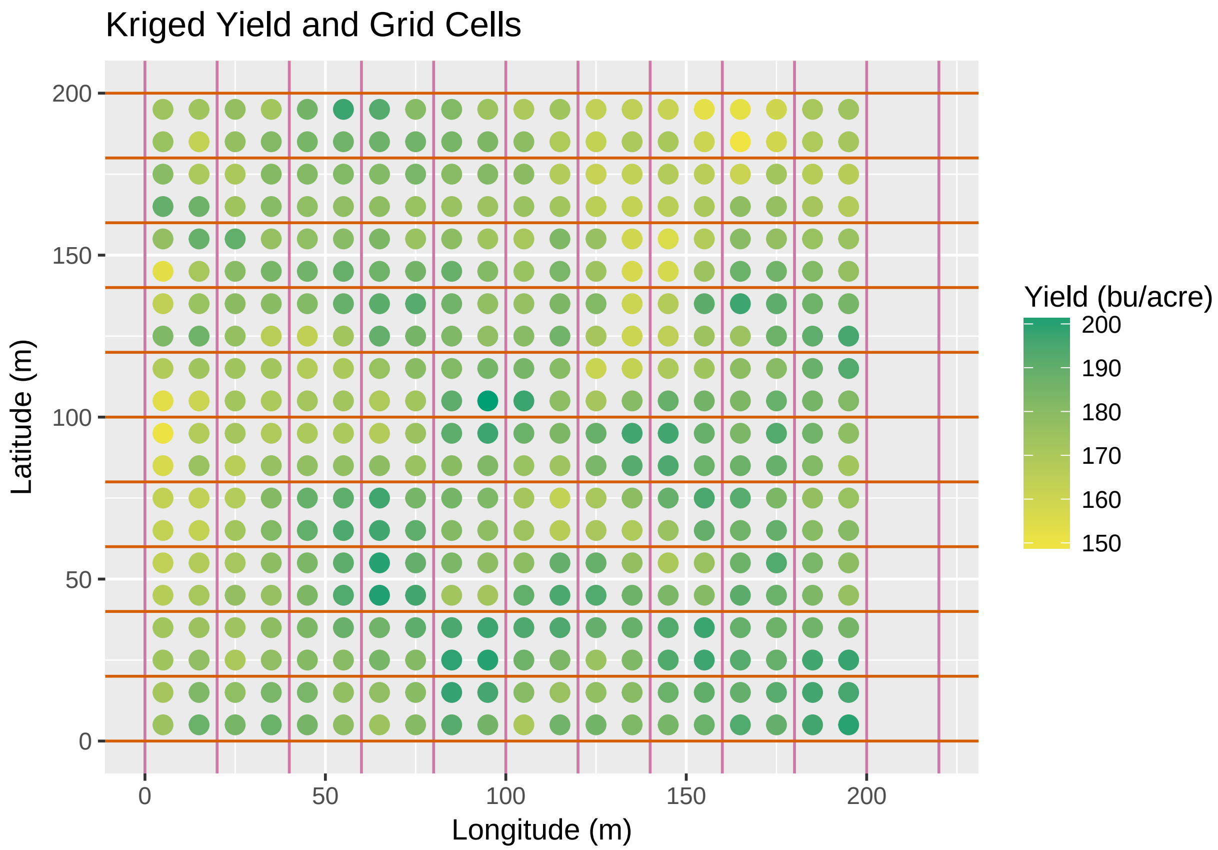
```
sample4.grd$Yield.krig <- Yield.krig$Yield.pred
```

**Plot the kriged estimates**

```
ggplot(sample.dat, aes(LonM, LatM)) +
  geom_point(aes(colour = Yield),size=1) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_point(aes(LonM, LatM,color=Yield.krig), data=sample4.grd,size=2) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Yield and Kriged Yield
```




```
ggplot(sample4.grd, aes(LonM, LatM)) +
geom_point(aes(colour = Yield.krig),size=3) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Kriged Yield and Grid C
```

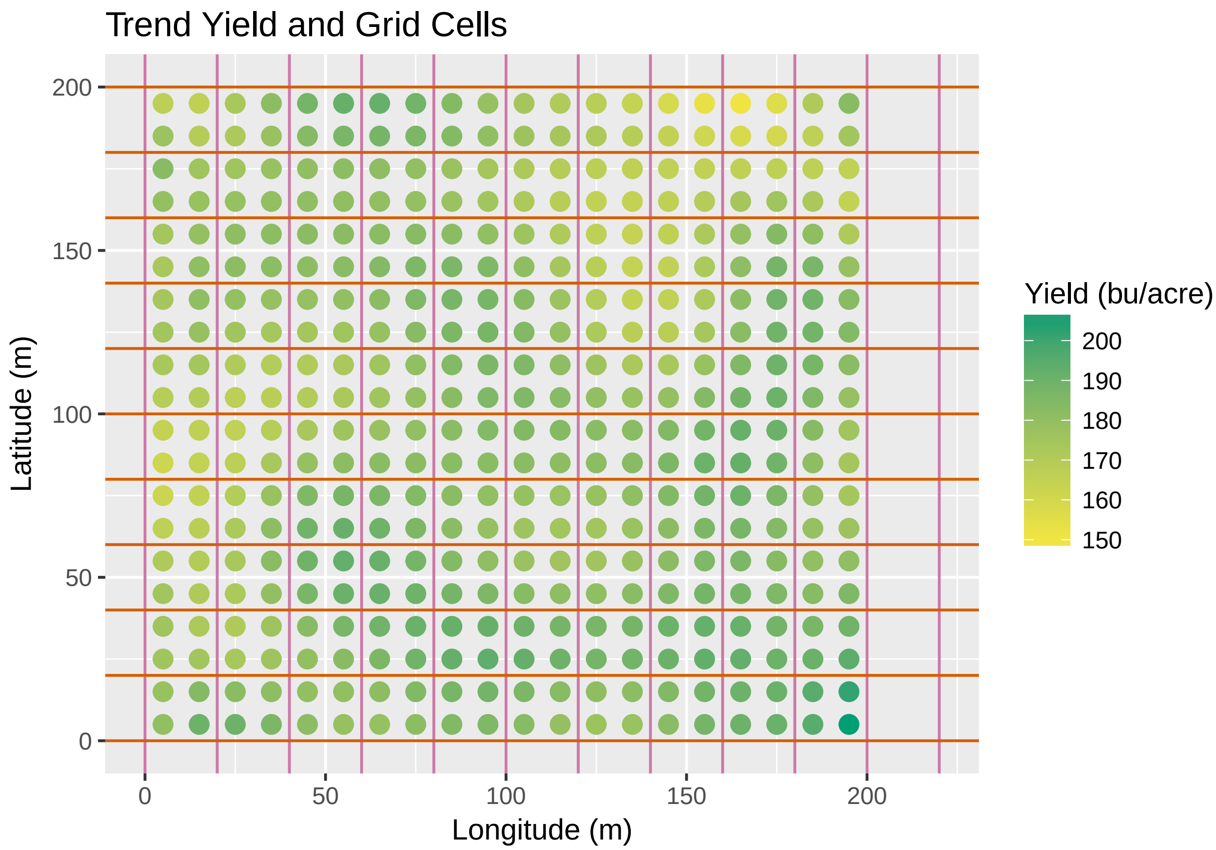



## Trend Surface Analysis

We use an 11th degree polynomial to fit our data.

```
Yield11.lm <-lm(Yield ~poly(LonM, LatM, degree=11), data=sample.dat)
sample4.grd$Yield.trend <- predict(Yield11.lm,sample4.grd)
```

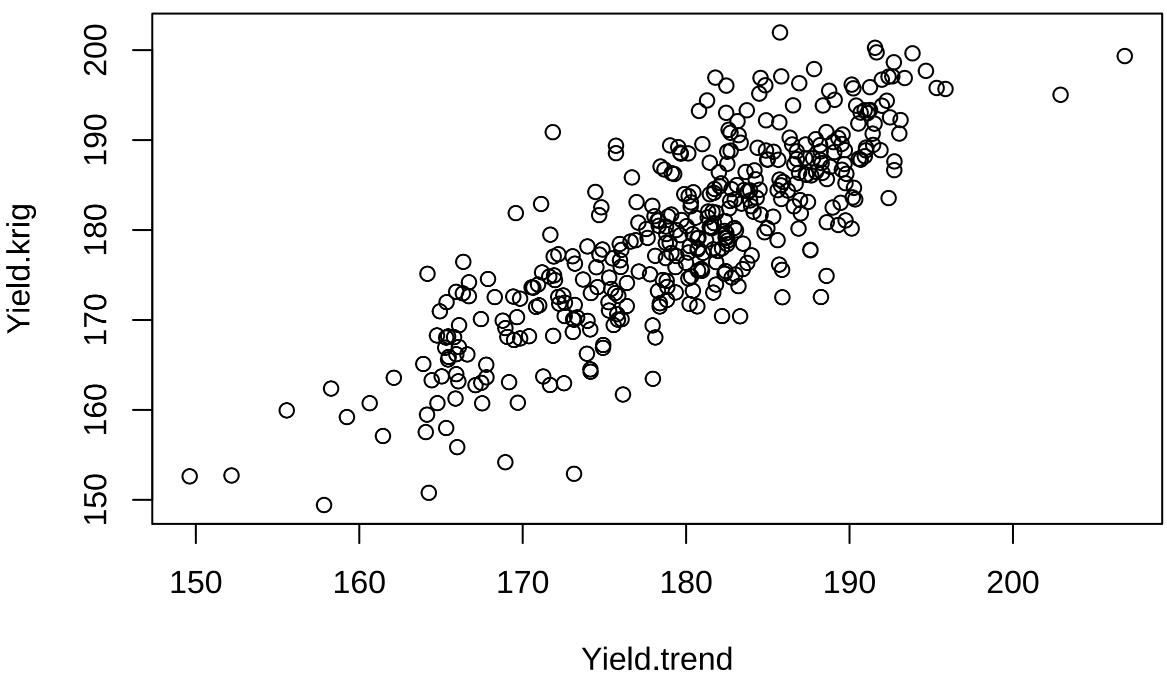
```
ggplot(sample4.grd, aes(LonM, LatM)) +
  geom_point(aes(colour = Yield.trend),size=3) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Trend Yield and Grid Ce
```




**Compare Estimates**

A simple plot to compare the two methods.

```
plot(Yield.krig ~ Yield.trend,data=sample4.grd)
```



## Cell Means

Previously, we indentified unique combinations of rows and columns as cells. We use this as an index for summary statistics, in this case simply means.

First, since kriged means and trend means are based on the same data grid, we'll compute those as we create a data frame. Since there are an equal number of observations, we can simply compute the average of latitude and longitude to provide a center point for each cell.

We need to call `as.vector` to strip row names and allow data to work with `lisa` plots

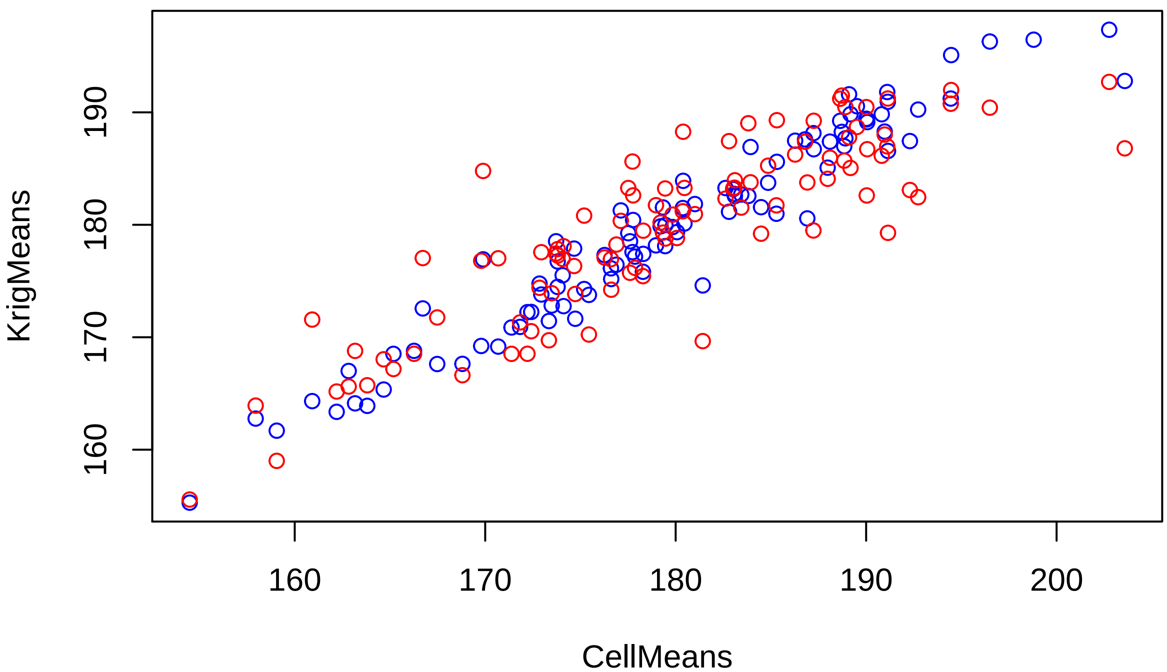
```
KrigMeans = tapply(sample4.grd$Yield.krig,list(sample4.grd$cell),mean,na.rm=TRUE)
TrendMeans = tapply(sample4.grd$Yield.trend,list(sample4.grd$cell),mean,na.rm=TRUE)
Grid.dat <- data.frame(
   Cell = levels(sample4.grd$cell),
   KrigMeans = as.vector(KrigMeans),
   TrendMeans = as.vector(TrendMeans),
   LonM = as.vector(tapply(sample4.grd$LonM,list(sample4.grd$cell),mean,na.rm=TRUE)),
   LatM = as.vector(tapply(sample4.grd$LatM,list(sample4.grd$cell),mean,na.rm=TRUE))
)
```

We take an extra step in computing cell means to index using `Cell`. This is necessary if we are not sure that the raw data is arranged in the same row-column order as the sample grid.

```
CellMeans = tapply(sample.dat$Yield,list(sample.dat$cell),mean,na.rm=TRUE)
Grid.dat$CellMeans <- as.vector(CellMeans[as.character(Grid.dat$Cell)])
```

And a simple plot to compare the three methods.

```
plot(KrigMeans~CellMeans,data=Grid.dat,col="blue")
points(TrendMeans~CellMeans,data=Grid.dat,col="red")
```



# Model Comparions

## Global spatial correlation.

We consider global spatial metrics of the three grids using Moran's $I$.

```
grid.dists <- as.matrix(dist(cbind(Grid.dat$LonM, Grid.dat$LatM)))
grid.dists <- 1/grid.dists
diag(grid.dists) <- 0
library(ape)
```

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

```
##
## Attaching package: 'ape'
```

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

```
Moran.I(Grid.dat$CellMeans, grid.dists)
```

```
## $observed
## [1] 0.09819354
##
## $expected
## [1] -0.01010101
##
## $sd
## [1] 0.01044322
##
## $p.value
## [1] 0
```

```
Moran.I(Grid.dat$KrigMeans, grid.dists)
```

```
## $observed
## [1] 0.1195487
##
## $expected
## [1] -0.01010101
##
## $sd
## [1] 0.01045457
##
## $p.value
## [1] 0
```

```
Moran.I(Grid.dat$TrendMeans, grid.dists)
```

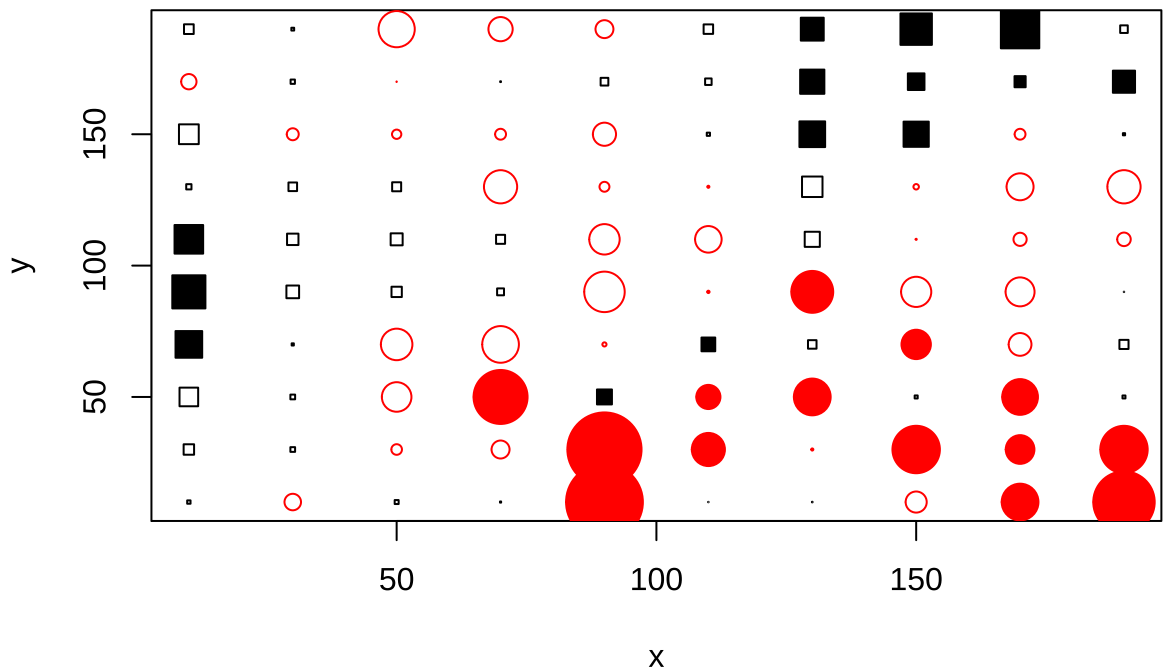
```
## $observed
## [1] 0.1430269
##
## $expected
## [1] -0.01010101
##
## $sd
```

```
## [1] 0.01042885
##
## $p.value
## [1] 0
```

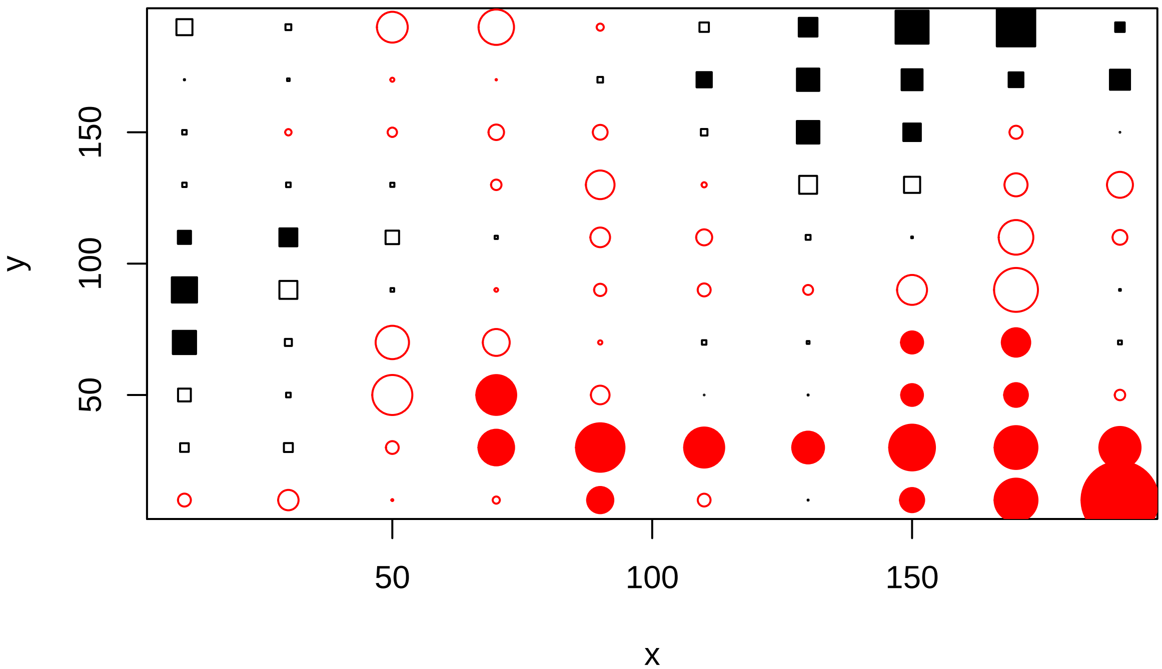
### Local spatial correlation.

We also compute local spatial measures. We use a neighborhood of 60 - this gives us 3 grid widths.

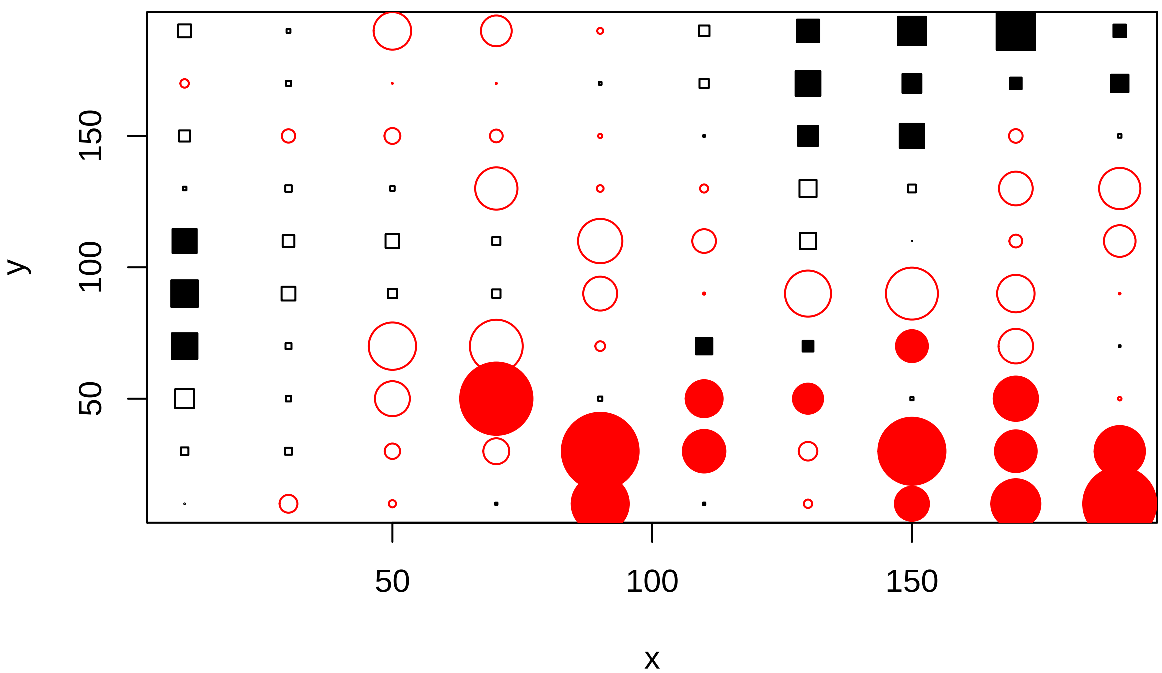
```r
library(ncf)
CellMeans.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$CellMeans,
                        neigh=60, resamp=500, quiet=TRUE)
plot.lisa(CellMeans.lisa, negh.mean=FALSE)
```




```r
TrendMeans.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$TrendMeans,
                        neigh=60, resamp=500, quiet=TRUE)
plot.lisa(TrendMeans.lisa, negh.mean=FALSE)
```




```
KrigMeans.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$KrigMeans,
                       neigh=60, resamp=500, quiet=TRUE)
plot.lisa(KrigMeans.lisa, negh.mean=FALSE)
```



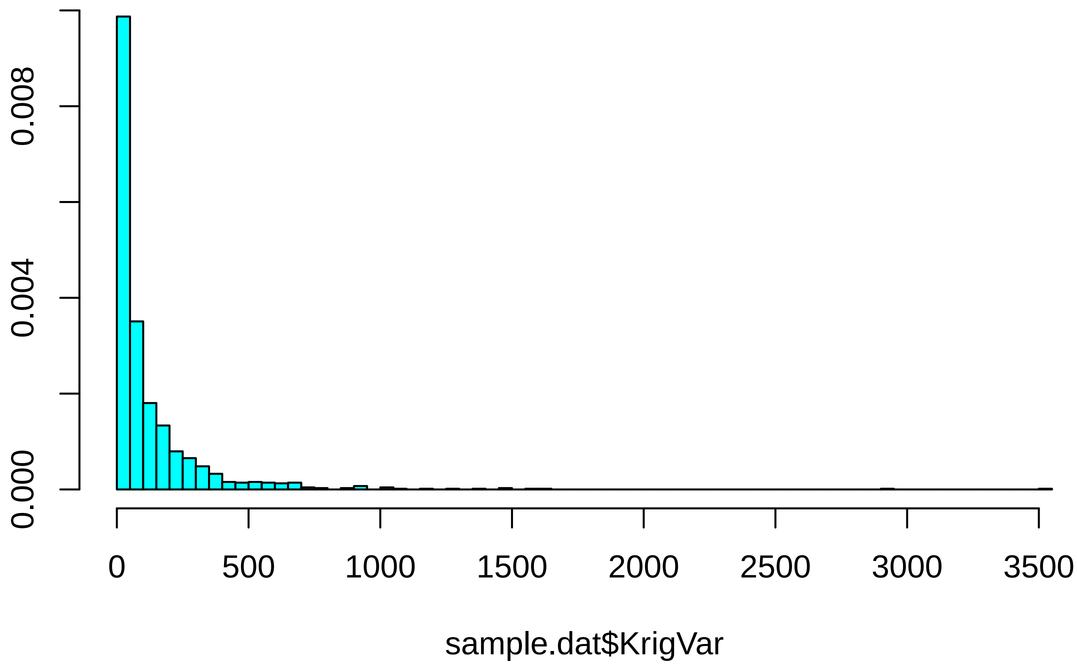

## Residuals

We also consider the spatial correlation among residuals. We compute a variance (residual squared) since we expect the sum of differences to be 0 for the cell means method. We should also note that the cell means method computes a minimum variance estimate for the mean yield of each grid cell. Both kriging and trend surface modeling with increase the variance of grid means, relative to the observations within each grid, but that's to be expected, since both methods include information from outside the grid.

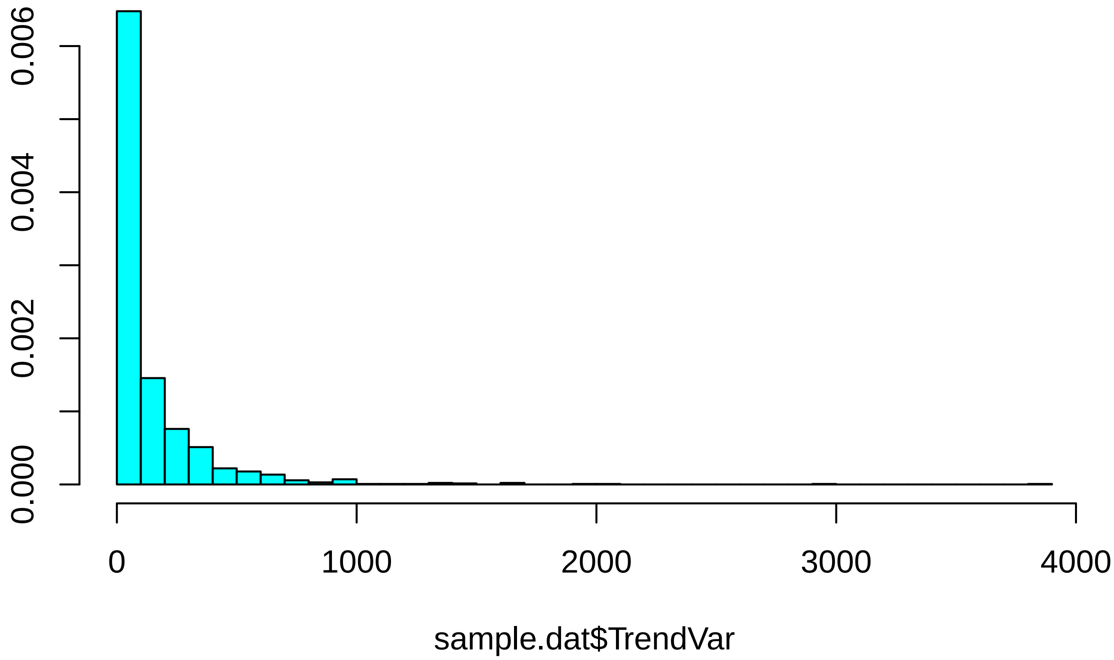
### Global spatial correlation.

```
sample.dat$KrigVar <- as.vector(sample.dat$Yield - KrigMeans[as.character(sample.dat$cell)])^2
sample.dat$TrendVar <- as.vector(sample.dat$Yield - TrendMeans[as.character(sample.dat$cell)])^2
sample.dat$CellVar <- as.vector(sample.dat$Yield - CellMeans[as.character(sample.dat$cell)])^2
```

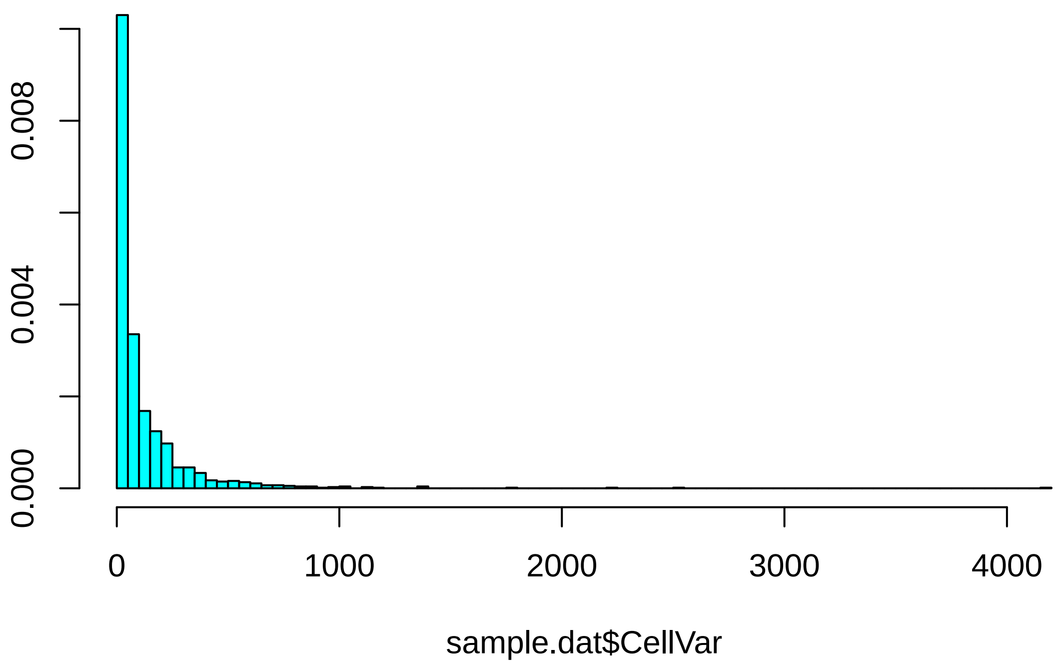
```
truehist(sample.dat$KrigVar)
```




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



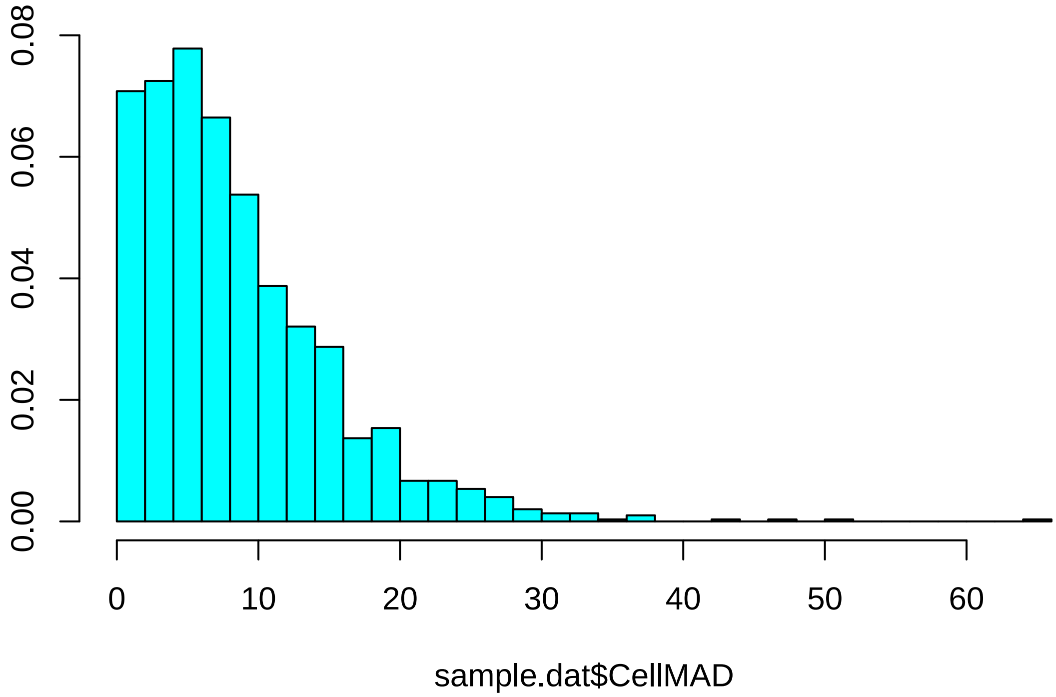

```
truehist(sample.dat$CellVar)
```



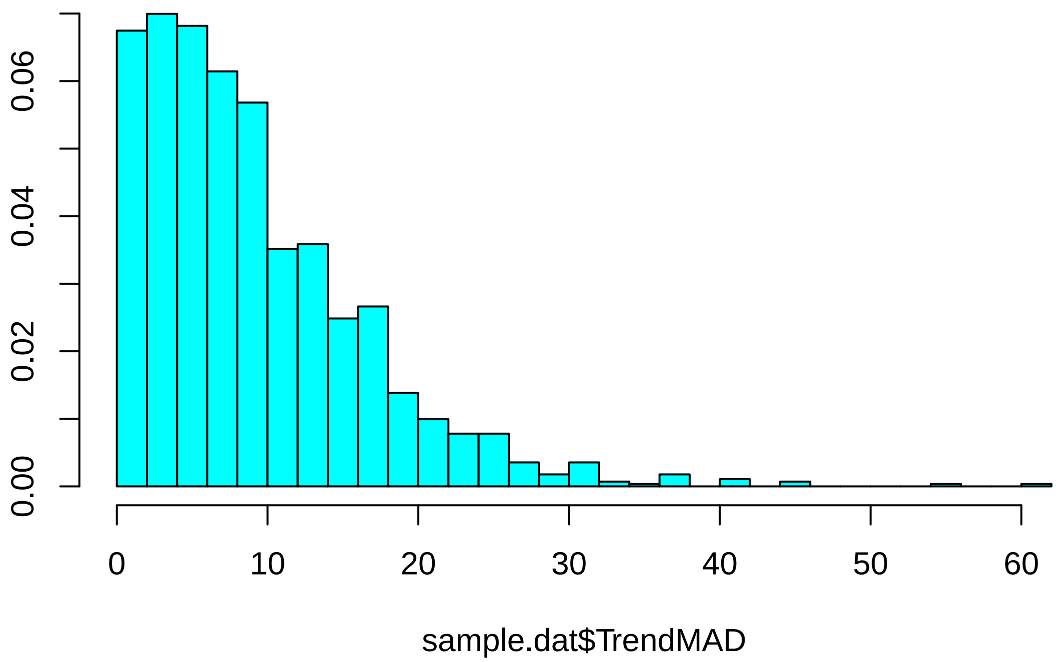

Mean absolute deviation is also valid measure of difference from a central tendency.

```
sample.dat$KrigMAD <- as.vector(abs(sample.dat$Yield - KrigMeans[as.character(sample.dat$cell)]))
sample.dat$TrendMAD <- as.vector(abs(sample.dat$Yield - TrendMeans[as.character(sample.dat$cell)]))
sample.dat$CellMAD <- as.vector(abs(sample.dat$Yield - CellMeans[as.character(sample.dat$cell)]))
```

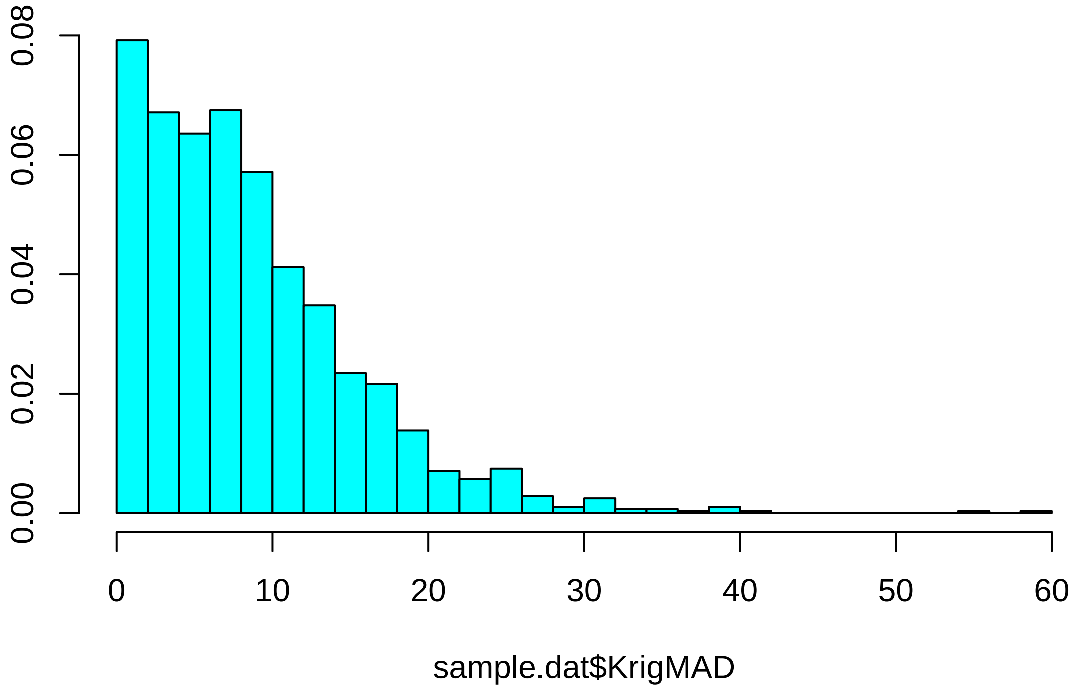
```
truehist(sample.dat$CellMAD)
```




```r
truehist(sample.dat$TrendMAD)
```



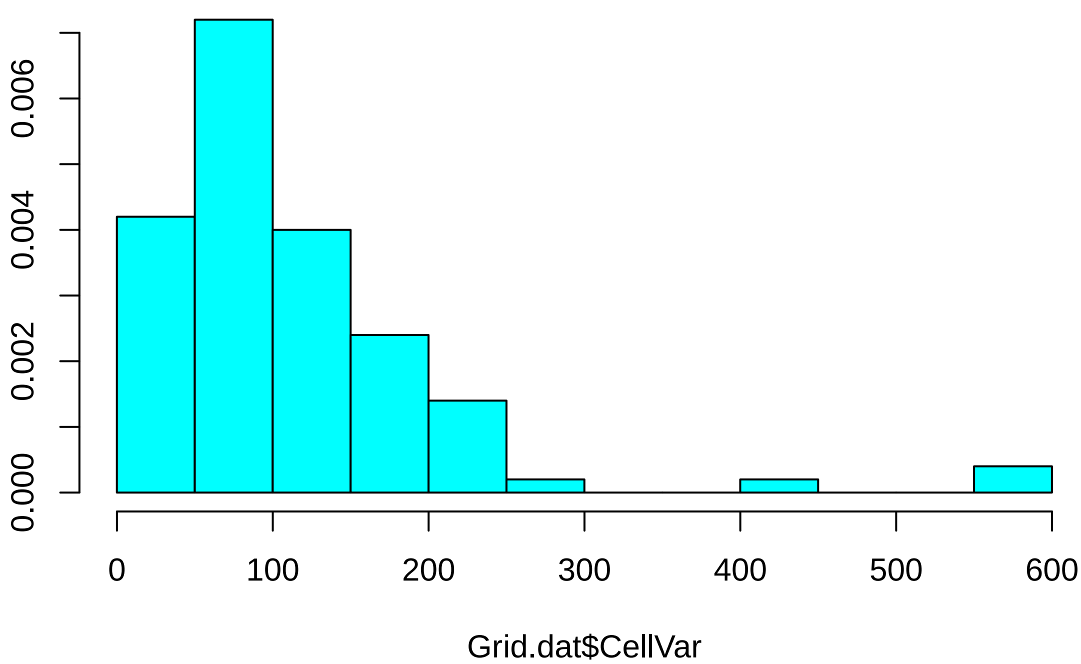

```r
truehist(sample.dat$KrigMAD)
```



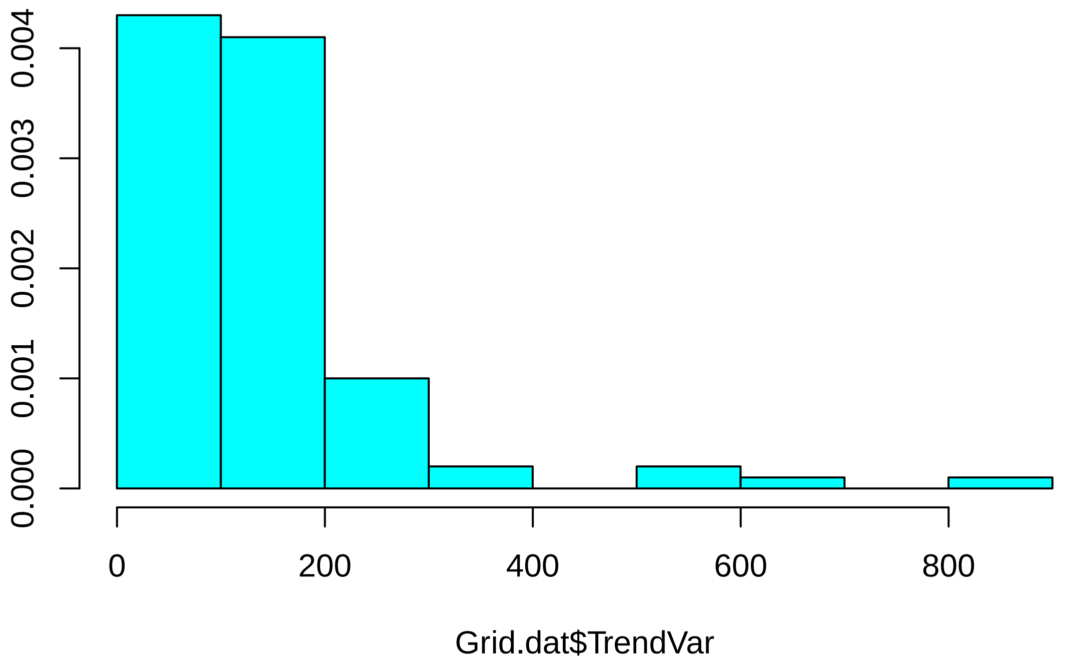

```
Grid.dat$CellVar <- as.vector(tapply(sample.dat$CellVar,list(sample.dat$cell),mean,na.rm=TRUE)[as.chara
Grid.dat$TrendVar <- as.vector(tapply(sample.dat$TrendVar,list(sample.dat$cell),mean,na.rm=TRUE)[as.cha
Grid.dat$KrigVar <- as.vector(tapply(sample.dat$KrigVar,list(sample.dat$cell),mean,na.rm=TRUE)[as.chara

Grid.dat$CellMAD <- as.vector(tapply(sample.dat$CellMAD,list(sample.dat$cell),mean,na.rm=TRUE)[as.chara
Grid.dat$TrendMAD <- as.vector(tapply(sample.dat$TrendMAD,list(sample.dat$cell),mean,na.rm=TRUE)[as.cha
Grid.dat$KrigMAD <- as.vector(tapply(sample.dat$KrigMAD,list(sample.dat$cell),mean,na.rm=TRUE)[as.chara
```

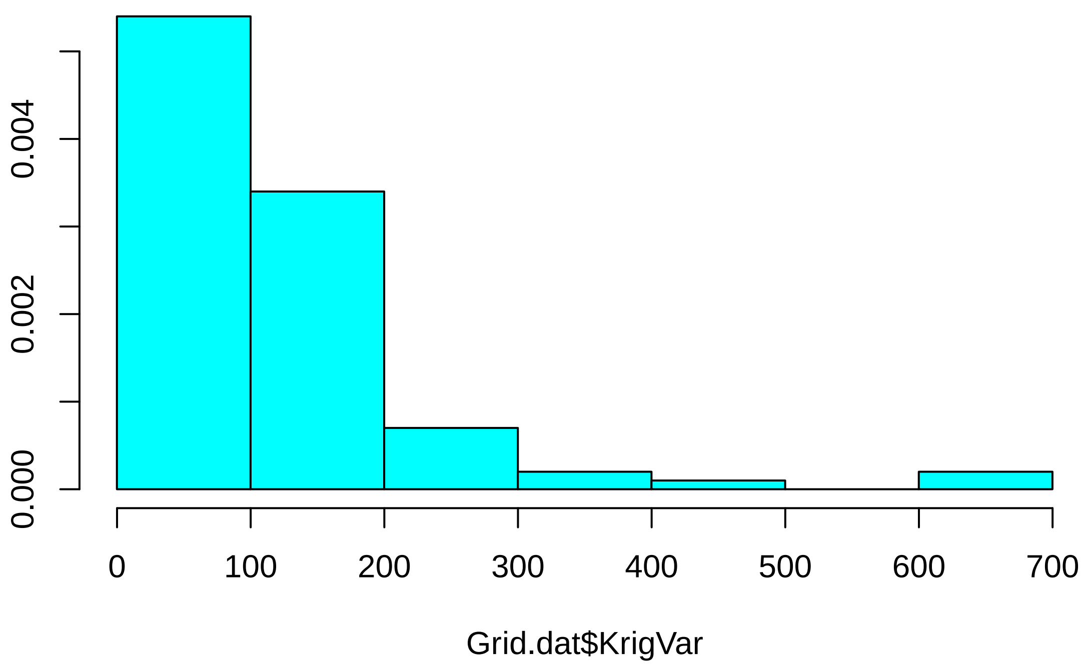
```
truehist(Grid.dat$CellVar)
```



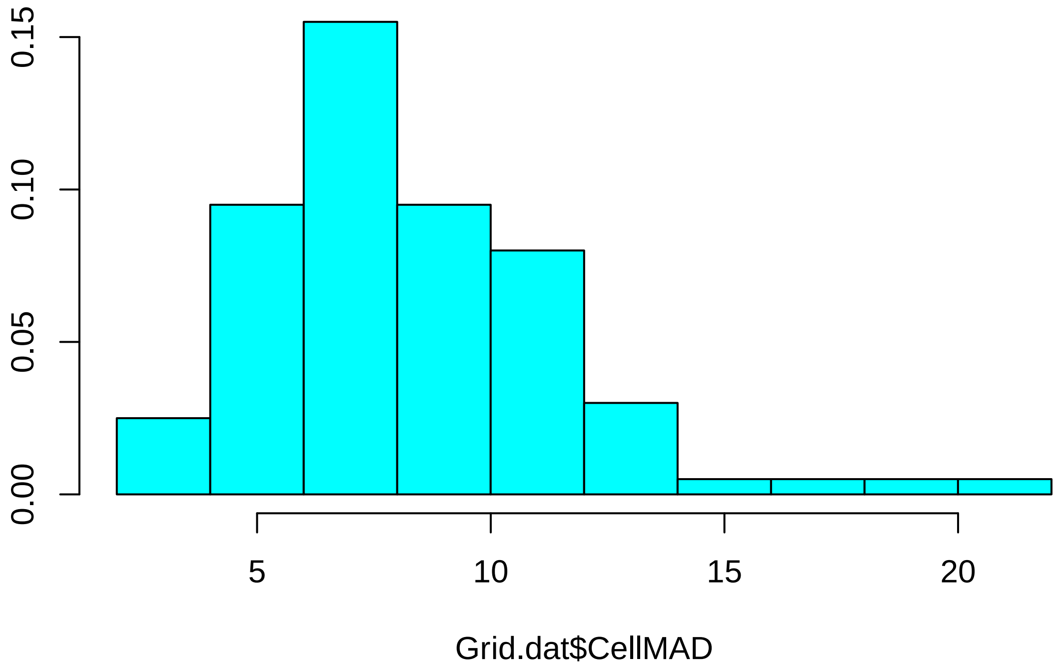
```r
truehist(Grid.dat$TrendVar)
```



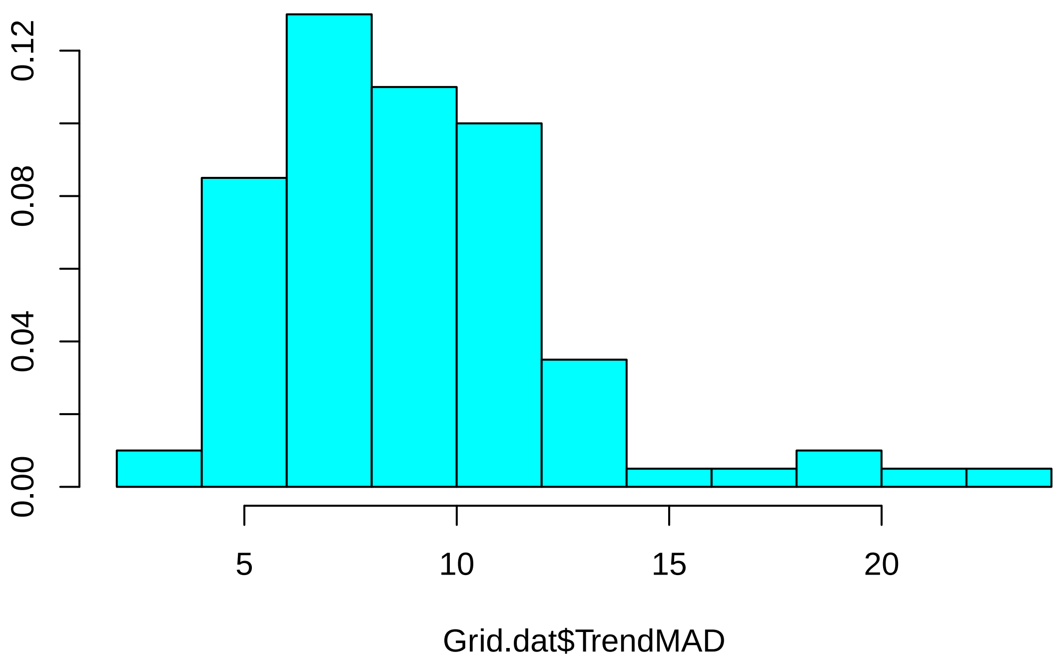

```r
truehist(Grid.dat$KrigVar)
```



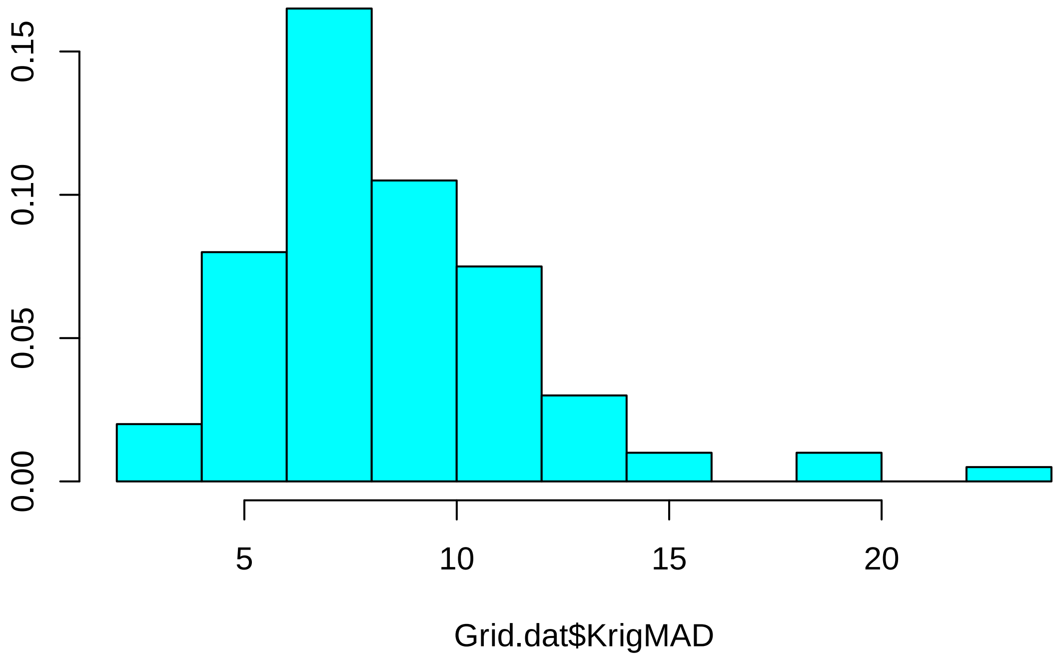

```r
truehist(Grid.dat$CellMAD)
```




```
truehist(Grid.dat$TrendMAD)
```




```
truehist(Grid.dat$KrigMAD)
```




```r
Moran.I(Grid.dat$CellVar, grid.dists)
```

```
## $observed
## [1] 0.01260698
##
## $expected
## [1] -0.01010101
##
## $sd
## [1] 0.009821323
##
## $p.value
## [1] 0.02077155
```

```r
Moran.I(Grid.dat$TrendVar, grid.dists)
```

```
## $observed
## [1] 0.02181996
##
## $expected
## [1] -0.01010101
##
## $sd
## [1] 0.009755206
##
## $p.value
## [1] 0.001067147
```

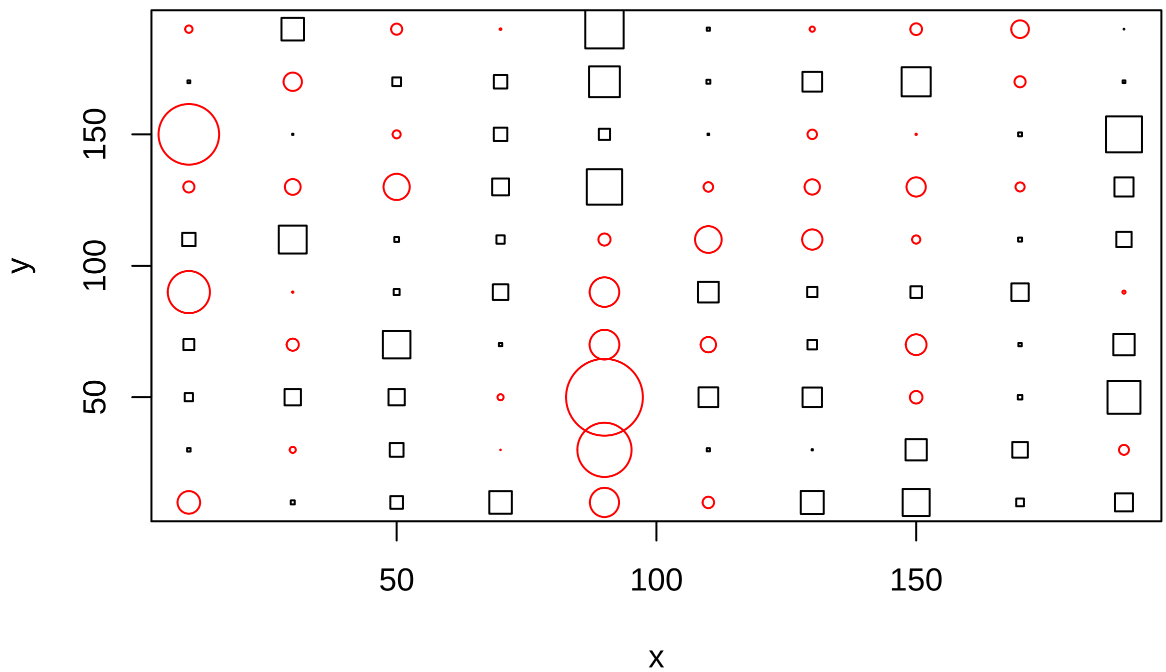
```r
Moran.I(Grid.dat$KrigVar, grid.dists)
```

```
## $observed
## [1] 0.01281254
##
## $expected
## [1] -0.01010101
```

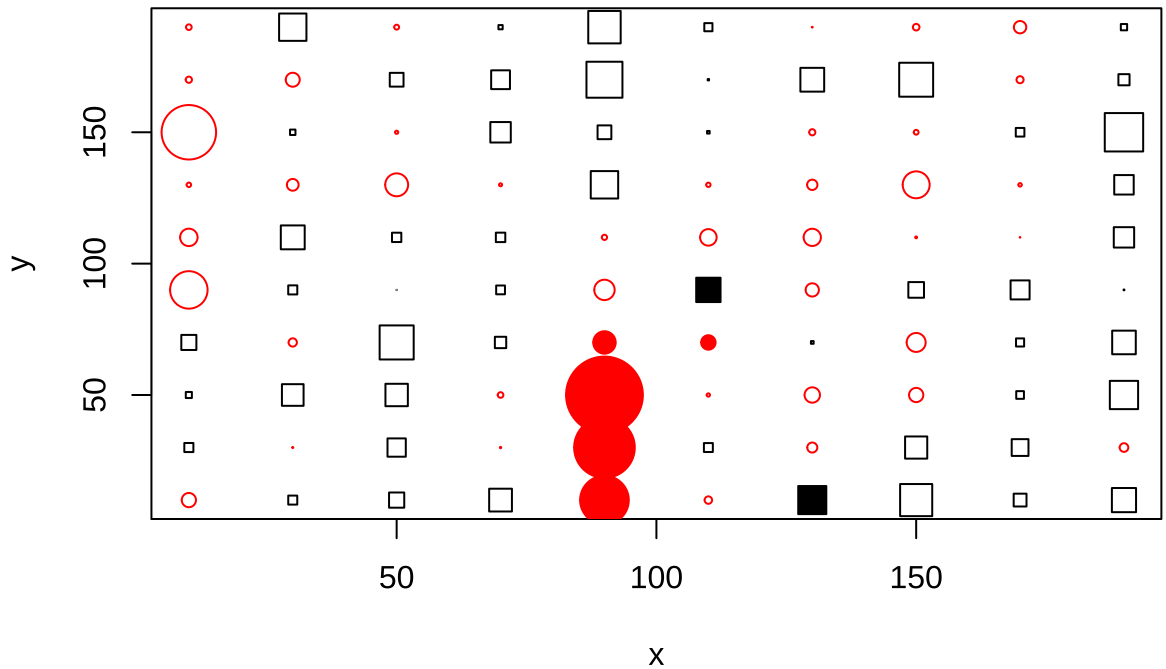
```
##
## $sd
## [1] 0.009821953
##
## $p.value
## [1] 0.01965382
```

### Local spatial correlation.

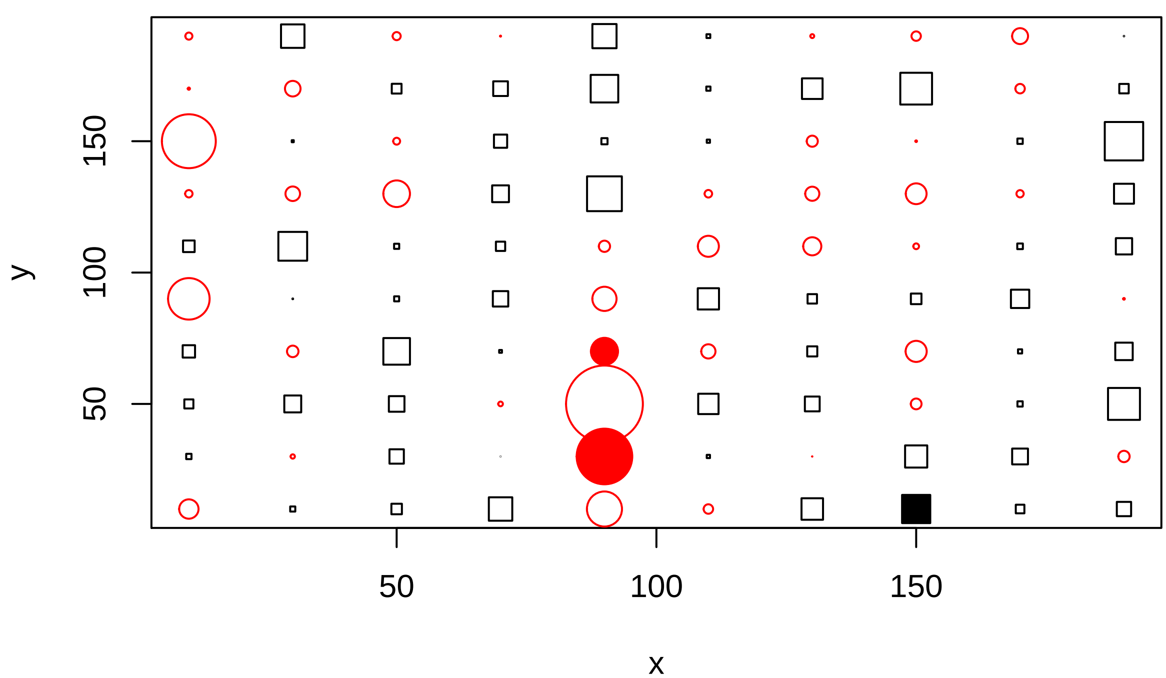
```
CellMAD.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$CellMAD,
                         neigh=60, resamp=500, quiet=TRUE)
plot.lisa(CellMAD.lisa, negh.mean=FALSE)
```



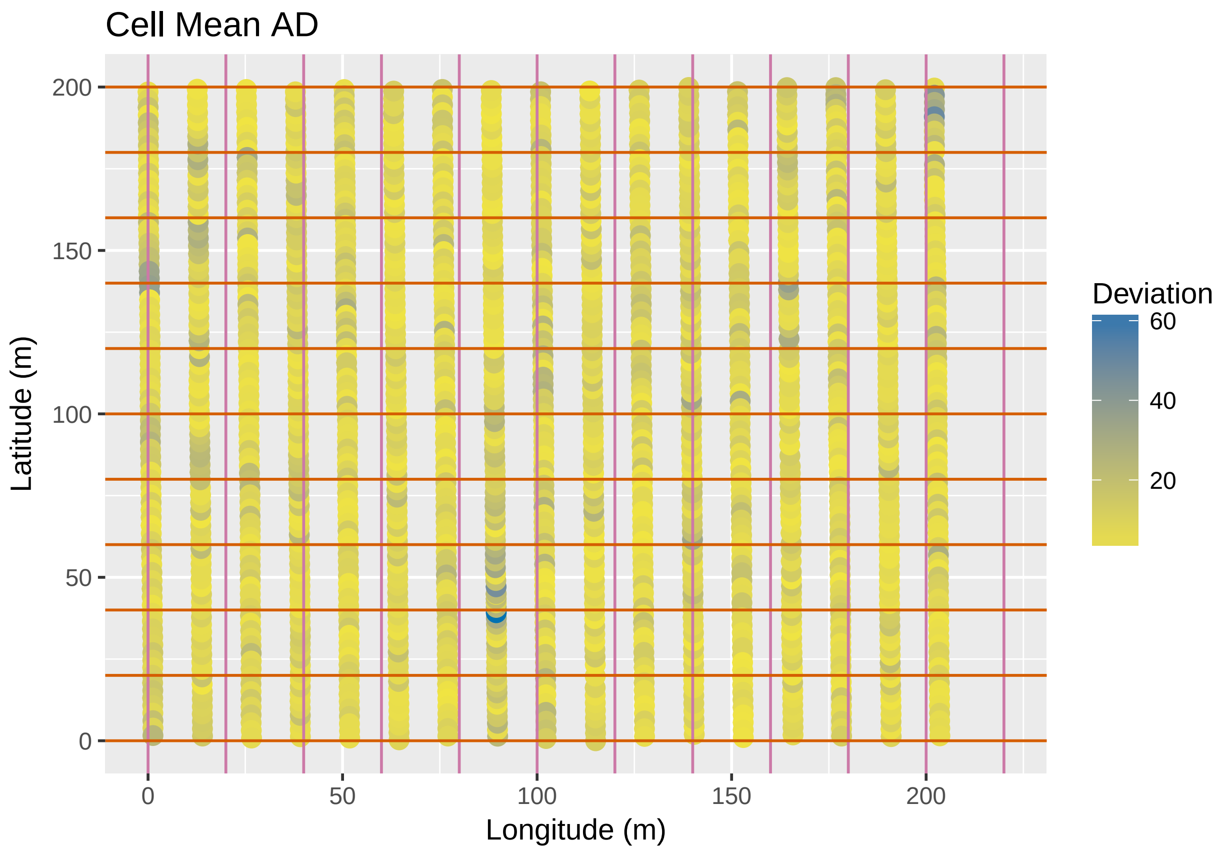

```
TrendMAD.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$TrendMAD,
                         neigh=60, resamp=500, quiet=TRUE)
plot.lisa(TrendMAD.lisa, negh.mean=FALSE)
```



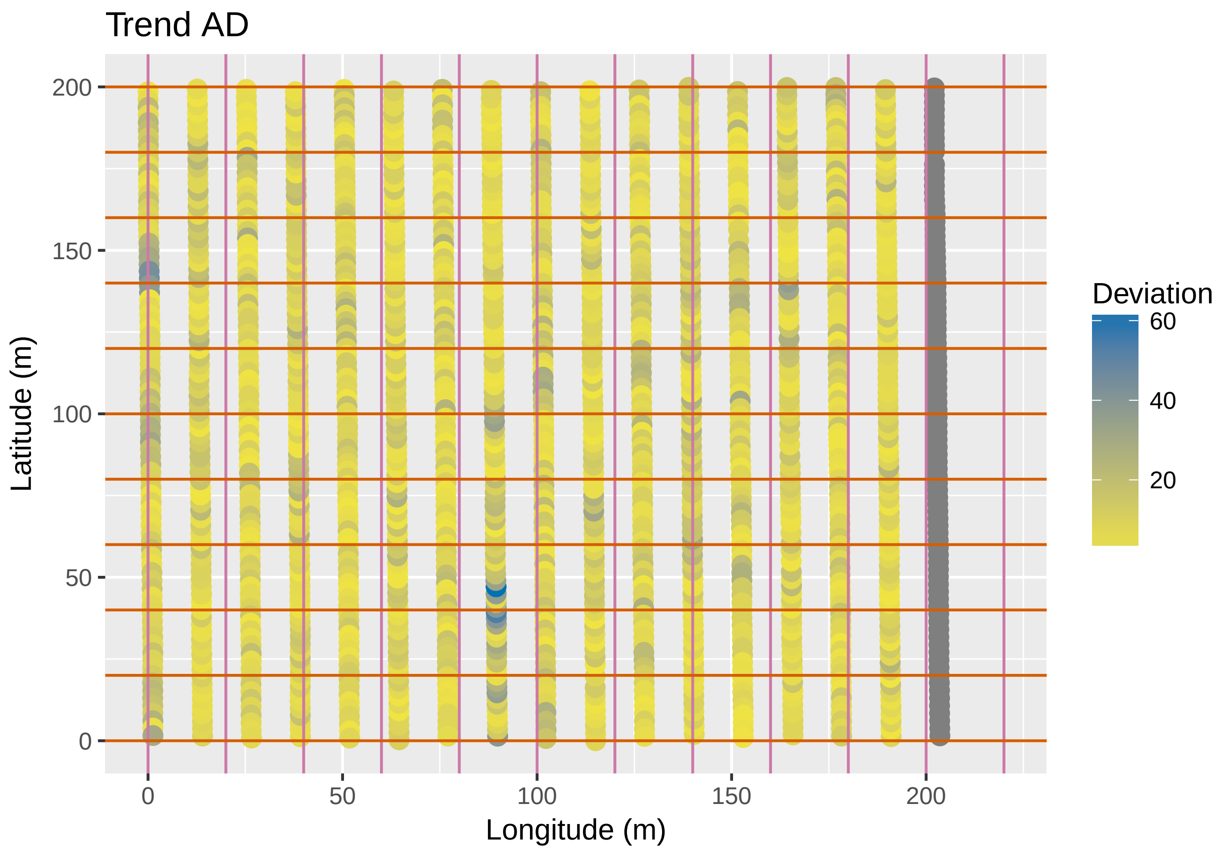

```
KrigMAD.lisa <- lisa(Grid.dat$LonM, Grid.dat$LatM, Grid.dat$KrigMAD,
                     neigh=60, resamp=500, quiet=TRUE)
plot.lisa(KrigMAD.lisa, negh.mean=FALSE)
```



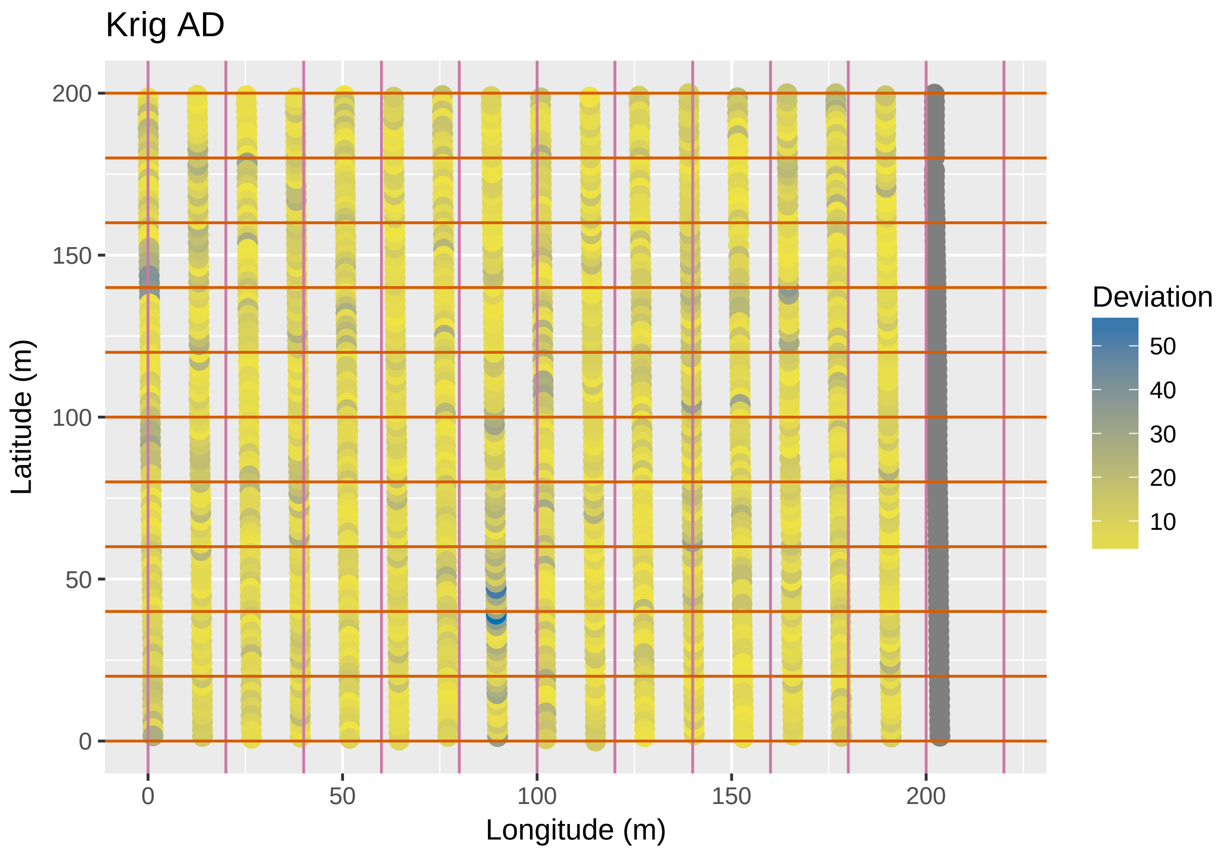

```
ggplot(sample.dat, aes(LonM, LatM)) +
geom_point(aes(colour = CellMAD),size=3) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[5]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
labs(colour = "Deviation", x="Longitude (m)", y="Latitude (m)", title = "Cell Mean AD")
```




```r
ggplot(sample.dat, aes(LonM, LatM)) +
geom_point(aes(colour = TrendMAD),size=3) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[5]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
labs(colour = "Deviation", x="Longitude (m)", y="Latitude (m)", title = "Trend AD")
```




```r
ggplot(sample.dat, aes(LonM, LatM)) +
geom_point(aes(colour = KrigMAD),size=3) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[5]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
labs(colour = "Deviation", x="Longitude (m)", y="Latitude (m)", title = "Krig AD")
```




```r
sample.dists <- as.matrix(dist(cbind(sample.dat$LonM, sample.dat$LatM)))
sample.dists <- 1/sample.dists
diag(sample.dists) <- 0
Moran.I(sample.dat$CellVar, sample.dists,na.rm=TRUE)
```

```
## $observed
## [1] 0.02781502
##
## $expected
## [1] -0.0006684492
##
## $sd
## [1] 0.001411935
##
## $p.value
## [1] 0
```

```r
Moran.I(sample.dat$TrendVar, sample.dists,na.rm=TRUE)
```

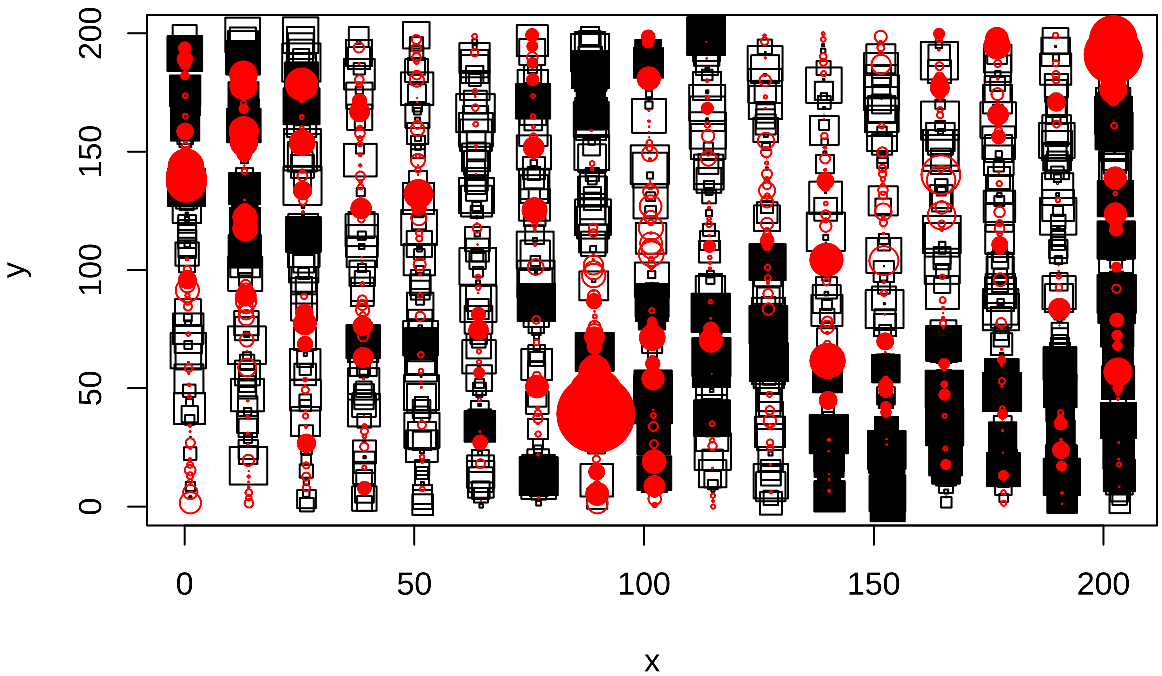
```
## $observed
## [1] 0.04000494
##
## $expected
## [1] -0.0006684492
##
## $sd
## [1] 0.001511464
##
```

```
## $p.value
## [1] 0
```

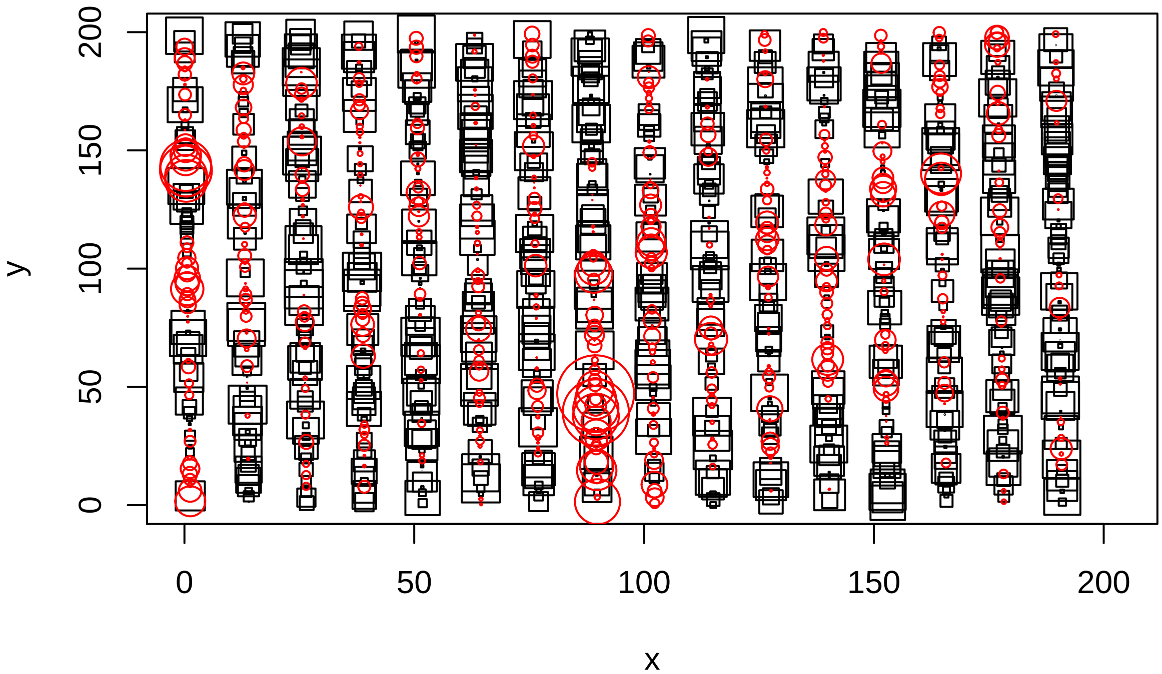
```
Moran.I(sample.dat$KrigVar, sample.dists,na.rm=TRUE)
```

```
## $observed
## [1] 0.03180123
##
## $expected
## [1] -0.0006684492
##
## $sd
## [1] 0.001505139
##
## $p.value
## [1] 0
```

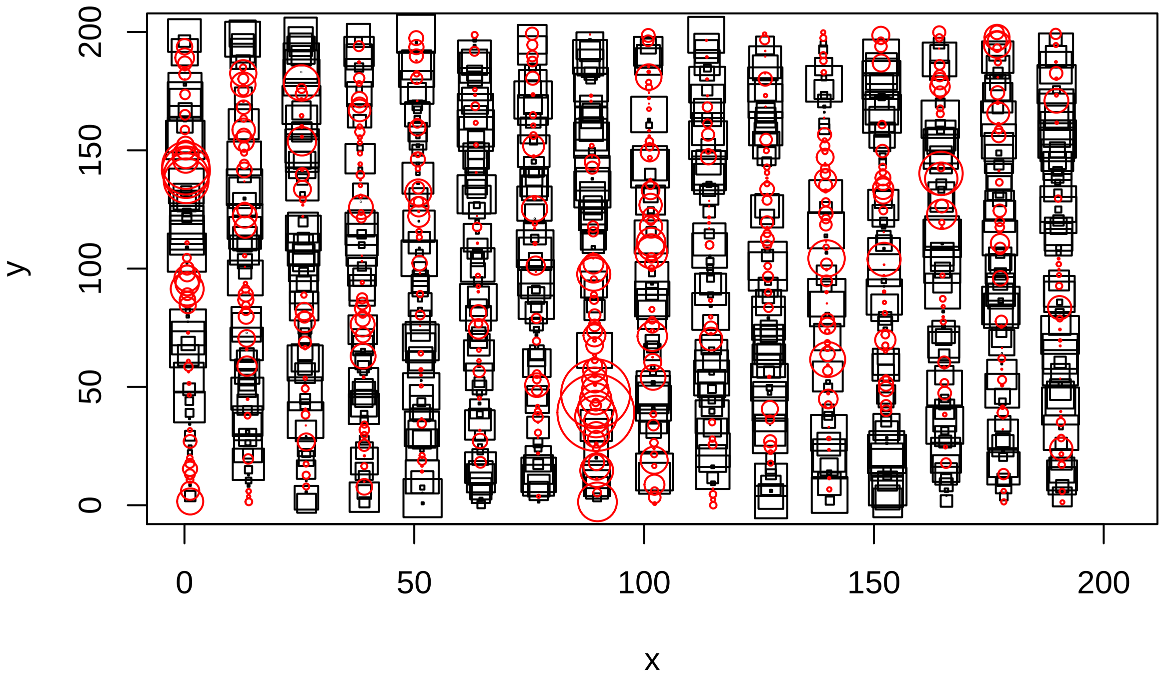
```
CellMAD.lisa <- lisa(sample.dat$LonM, sample.dat$LatM, sample.dat$CellMAD,
                     neigh=50, resamp=500, quiet=TRUE)
plot.lisa(CellMAD.lisa, negh.mean=FALSE)
```




```
TrendMAD.lisa <- lisa(sample.dat$LonM, sample.dat$LatM, sample.dat$TrendMAD,
                      neigh=50, resamp=500, quiet=TRUE)
plot.lisa(TrendMAD.lisa, negh.mean=FALSE)
```




```
KrigMAD.lisa <- lisa(sample.dat$LonM, sample.dat$LatM, sample.dat$KrigMAD,
                     neigh=50, resamp=500, quiet=TRUE)
plot.lisa(KrigMAD.lisa, negh.mean=FALSE)
```

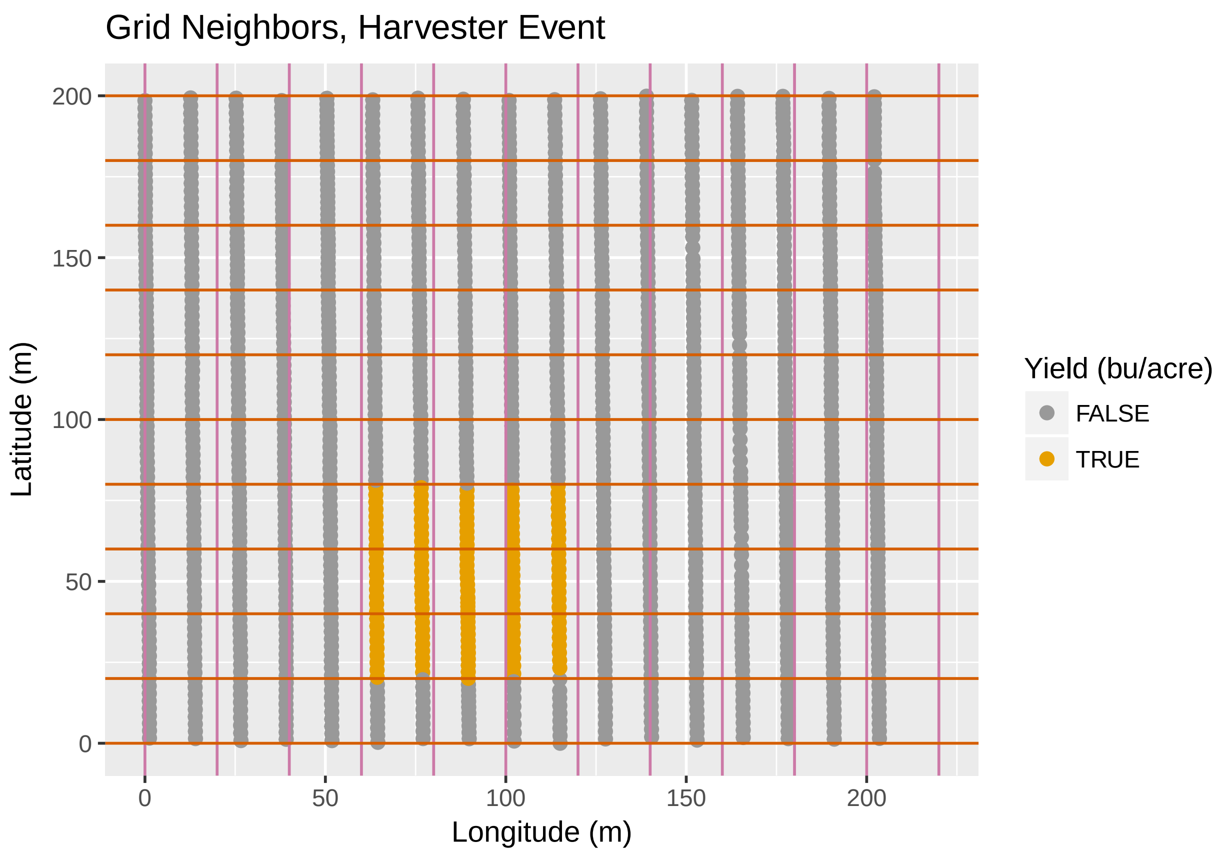



## Details of a Harvester Event

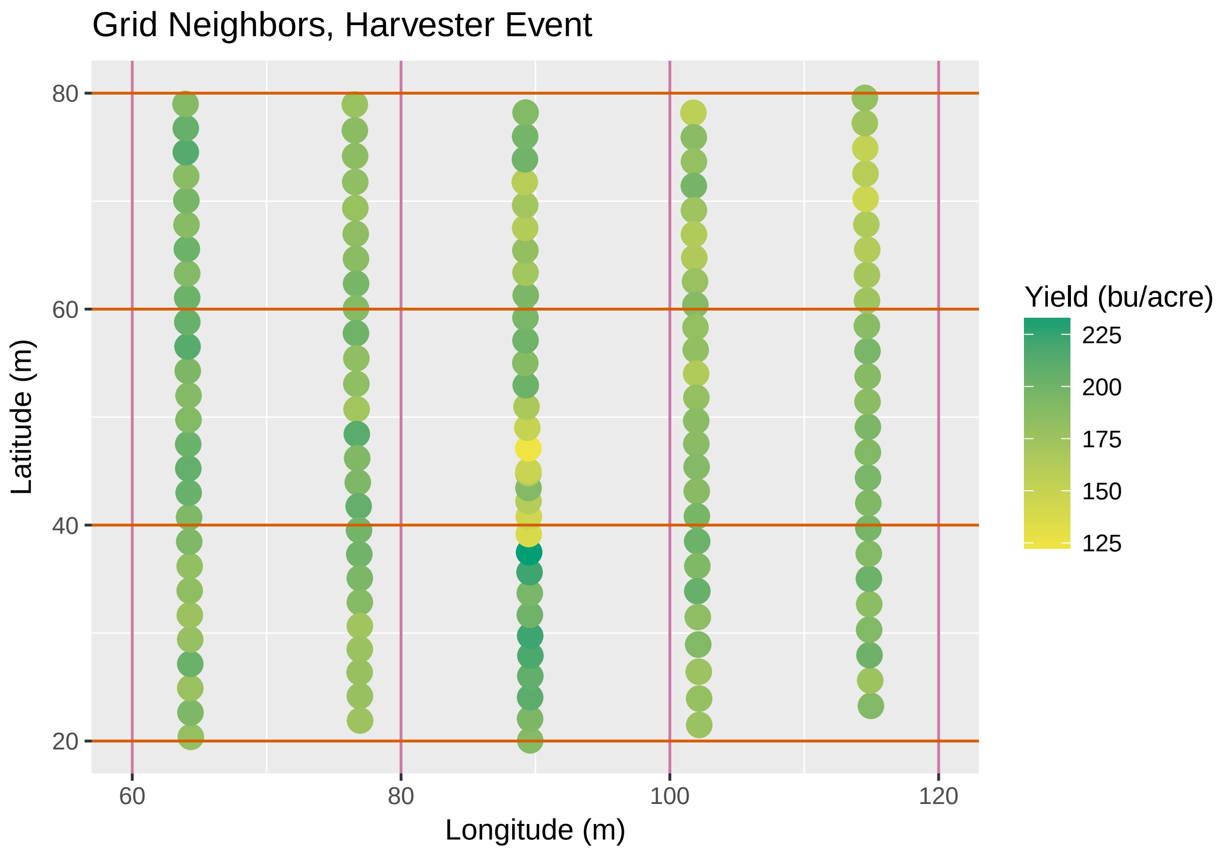
It looks like there was an event that affect harvest that might not be entirely due to spatial variation. We'll take a closer look in this region to consider how different methods of creating cell estimates handle this event.

```r
sample.dat$Event = (sample.dat$row %in% c(2:4) & sample.dat$col %in% c(4:6))
```

```r
ggplot(sample.dat, aes(LonM, LatM)) +
geom_point(aes(colour = Event),size=2) +
scale_colour_manual(values=cbPalette) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Grid Neighbors, Harveste
```

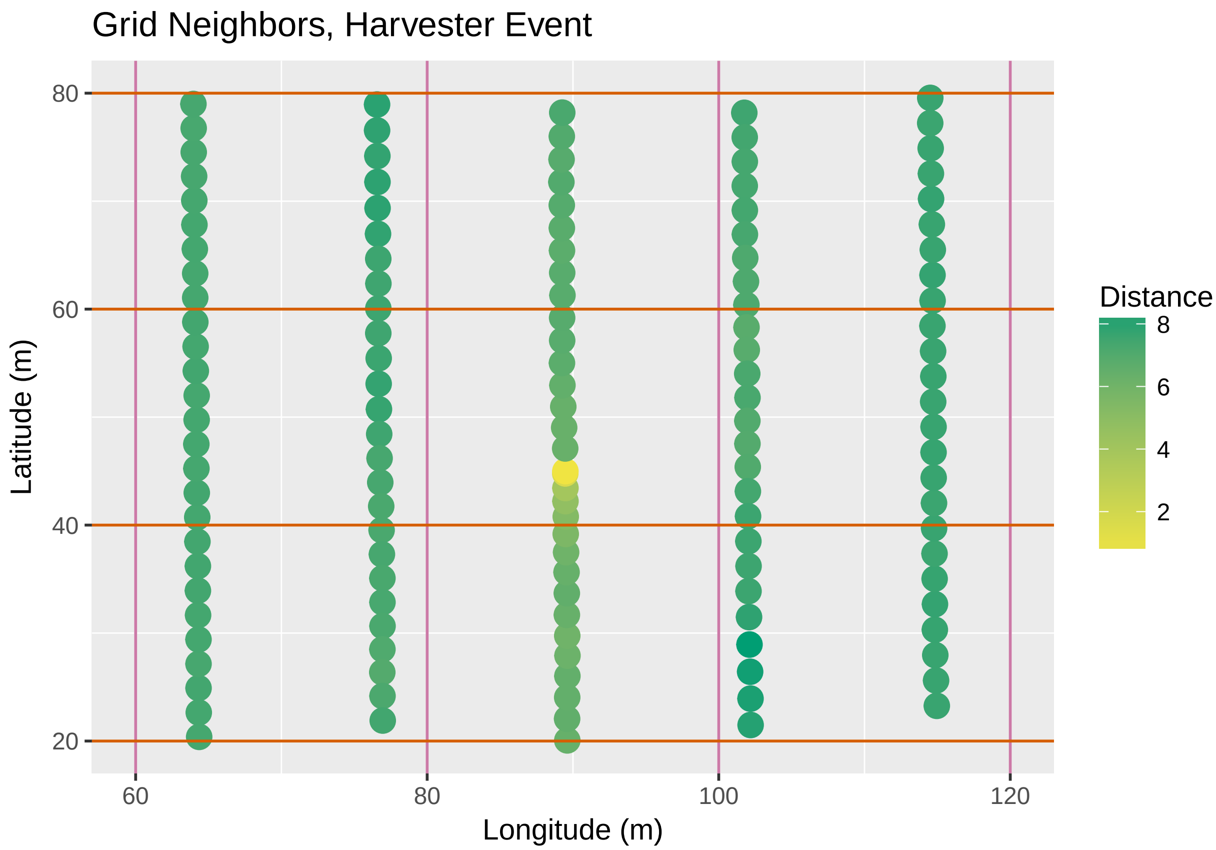



```r
Event.dat <- sample.dat[sample.dat$Event,]
ggplot(Event.dat, aes(LonM, LatM)) +
geom_point(aes(colour = Yield),size=4) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints[4:7],color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints[2:5],color = cbPalette[6]) +
labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Grid Neighbors, Harveste
```

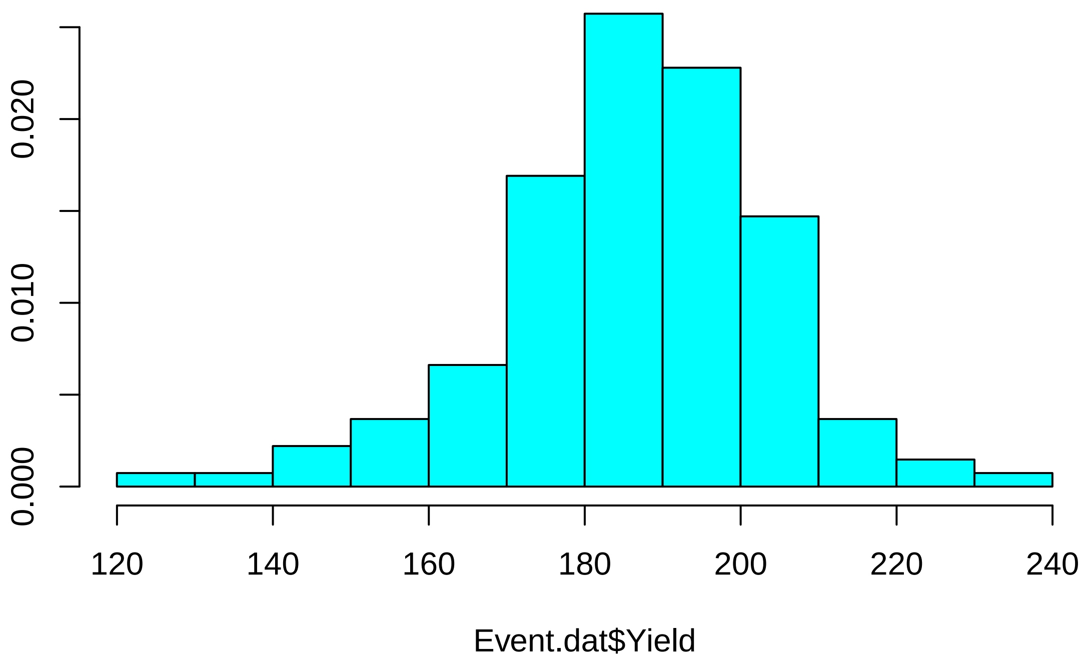



We can examine the distance traveled to help determine what happened during harvest.

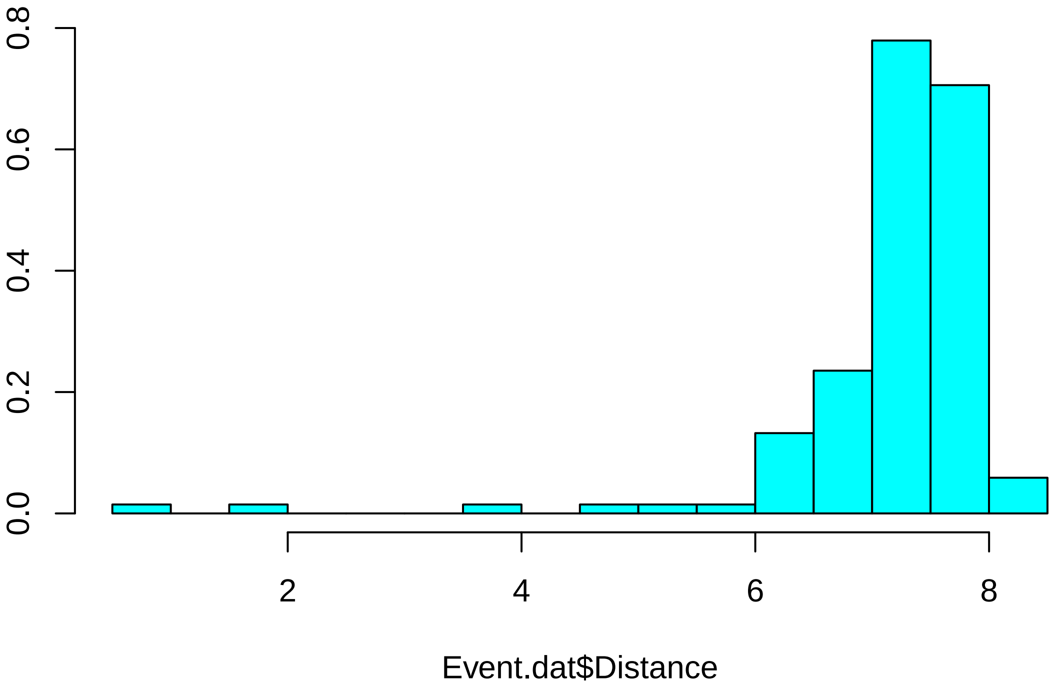
```
ggplot(Event.dat, aes(LonM, LatM)) +
geom_point(aes(colour = Distance),size=4) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints[4:7],color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints[2:5],color = cbPalette[6]) +
labs(colour = "Distance", x="Longitude (m)", y="Latitude (m)", title = "Grid Neighbors, Harvester Event
```




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




```r
truehist(Event.dat$Distance)
```



0.0
0.2
0.4
0.6
0.8
2
4
6
8
Event.dat$Distance