Case Study 1

Peter Claussen

10/3/2017

```r
library(gstat)
```

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

```r
library(MASS)
```

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

```r
library(ape)
```

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

```r
library(ncf)
```

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

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

```r
library(ggplot2)
```

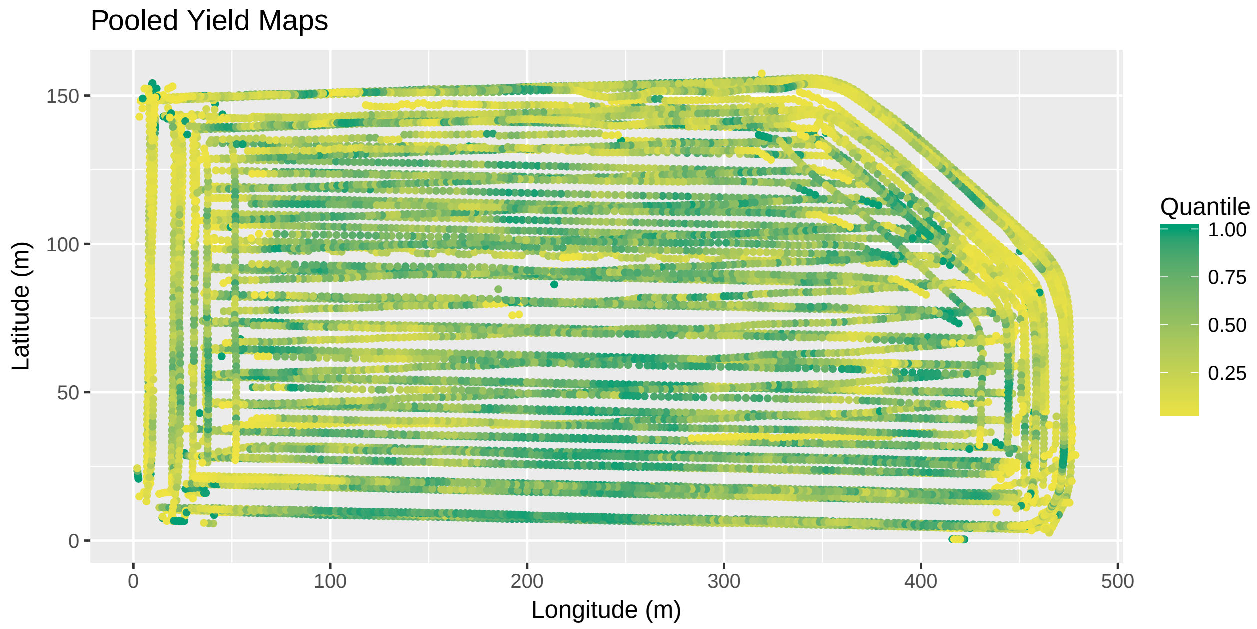
```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

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

```r
load(file="Pooled.Rda")
```

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

```r
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")
```



Grid Cells

Using the same grid cells as before, divide the pooled data into a network. We'll proceed on the assumption that the four years data is uniform enough to be pooled.

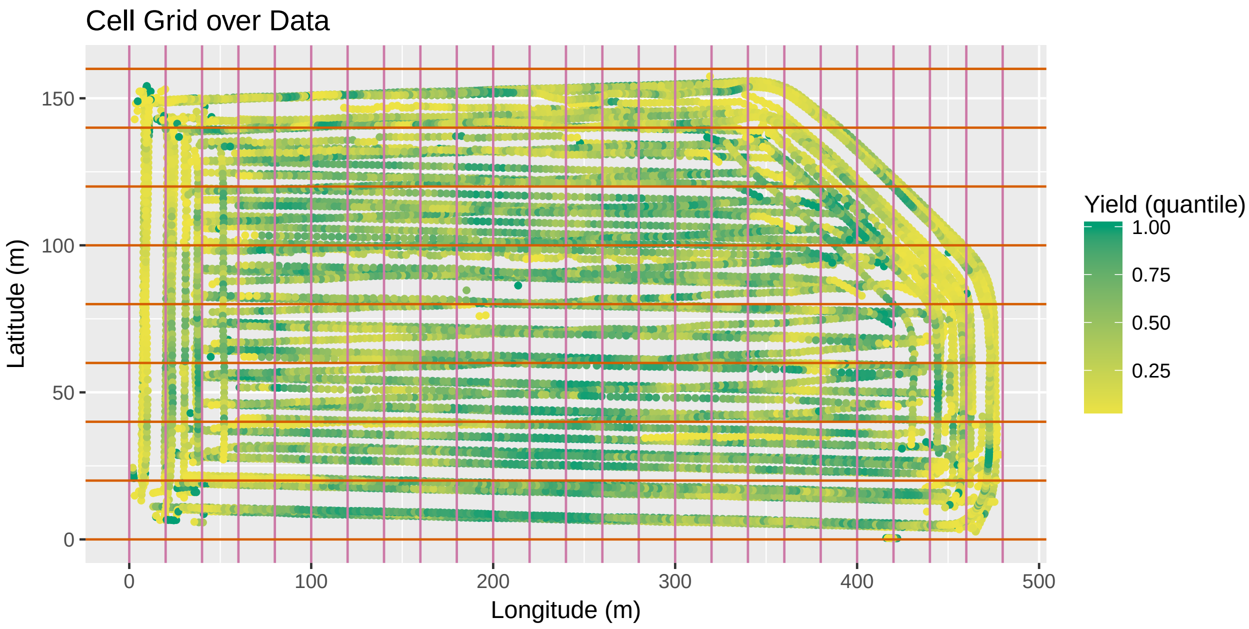
```
Pooled.dat$Row <- ceiling((Pooled.dat$Northing+0.1)/20)
Pooled.dat$Column <- ceiling((Pooled.dat$Easting+0.1)/20)
Pooled.dat$Cell <- as.factor(Pooled.dat$Row):as.factor(Pooled.dat$Column)
```

Cell Borders

```
rowPoints <- c(0,20*(1:ceiling(max(Pooled.dat$Northing+0.1)/20)))

colPoints <- c(0,20*(1:ceiling(max(Pooled.dat$Easting+0.1)/20)))
```

```
ggplot(Pooled.dat, aes(Easting, Northing)) +
  geom_point(aes(colour = Quantile),size=1) +
  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 (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Cell Grid over Data")
```



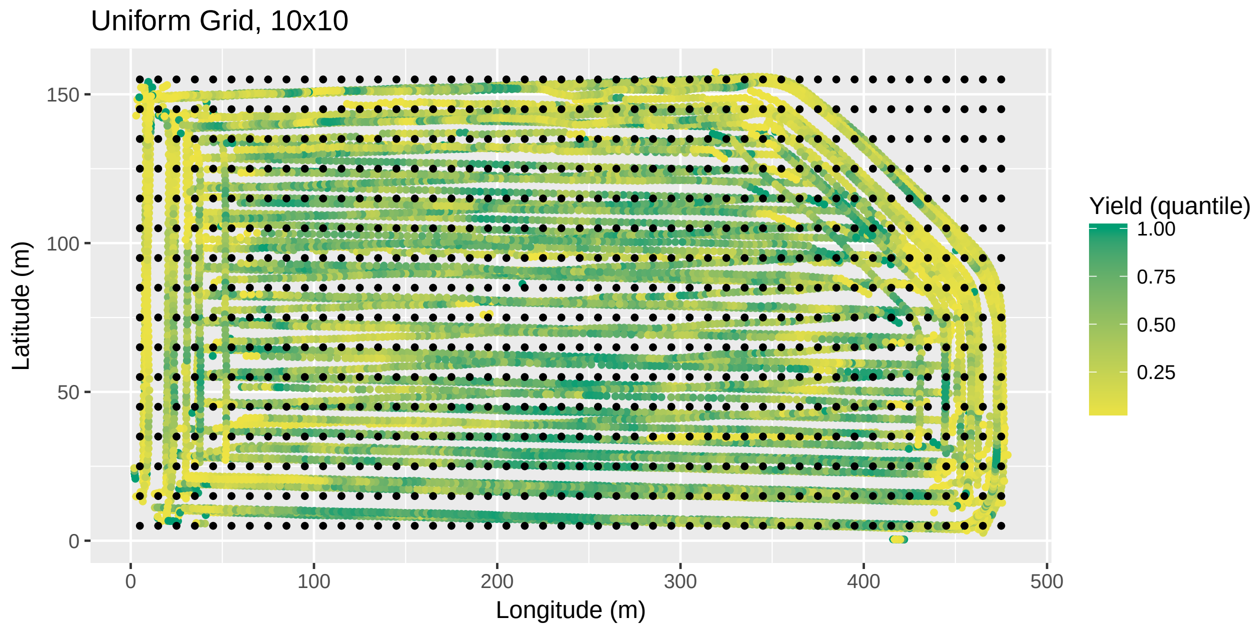
Sample Grid

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 5ms from the origin.

```
columns <- ceiling(max(Pooled.dat$Easting)/10)
rows <- ceiling(max(Pooled.dat$Northing)/10)
Sample10.grd <- expand.grid(Northing=((1:rows)*10)-5,
                            Easting=((1:columns)*10)-5)
Sample10.grd$Row <- ceiling((Sample10.grd$Northing-5+0.1)/20)
Sample10.grd$Column <- ceiling((Sample10.grd$Easting-5+0.1)/20)
Sample10.grd$Cell <- as.factor(Sample10.grd$Row):as.factor(Sample10.grd$Column)
head(Sample10.grd)
```

```
##   Northing Easting Row Column Cell
## 1        5       5   1      1  1:1
## 2       15       5   1      1  1:1
## 3       25       5   2      1  2:1
## 4       35       5   2      1  2:1
## 5       45       5   3      1  3:1
## 6       55       5   3      1  3:1
```

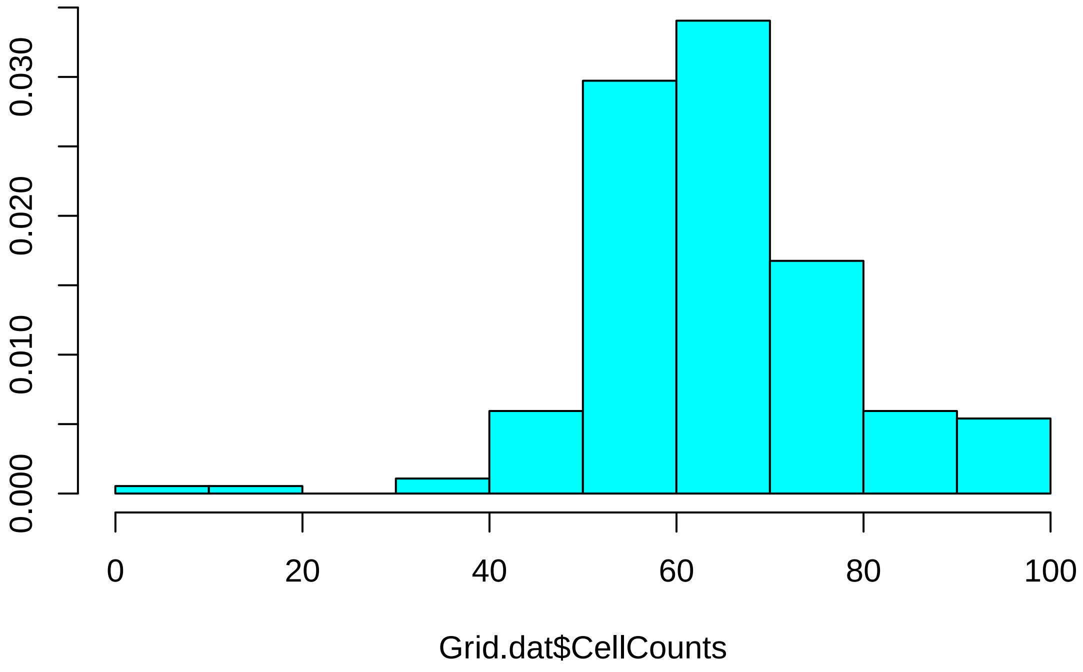
```
ggplot(Pooled.dat, aes(Easting, Northing)) +
  geom_point(aes(colour = Quantile),size=1) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_point(aes(Easting, Northing),data=Sample10.grd, size=1) +
  labs(colour = "Yield (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Uniform Grid, 10x10")
```



```
Grid.dat <- data.frame(
   Cell = levels(Sample10.grd$Cell),
   Easting = as.vector(tapply(Sample10.grd$Easting,list(Sample10.grd$Cell),mean,na.rm=TRUE)),
   Northing = as.vector(tapply(Sample10.grd$Northing,list(Sample10.grd$Cell),mean,na.rm=TRUE))
)

CellMeans = tapply(Pooled.dat$Quantile,list(Pooled.dat$Cell),mean,na.rm=TRUE)
Grid.dat$CellMeans <- as.vector(CellMeans[as.character(Grid.dat$Cell)])
CellCounts = tapply(Pooled.dat$Quantile,list(Pooled.dat$Cell),length)
Grid.dat$CellCounts <- as.vector(CellCounts[as.character(Grid.dat$Cell)])

truehist(Grid.dat$CellCounts)
```

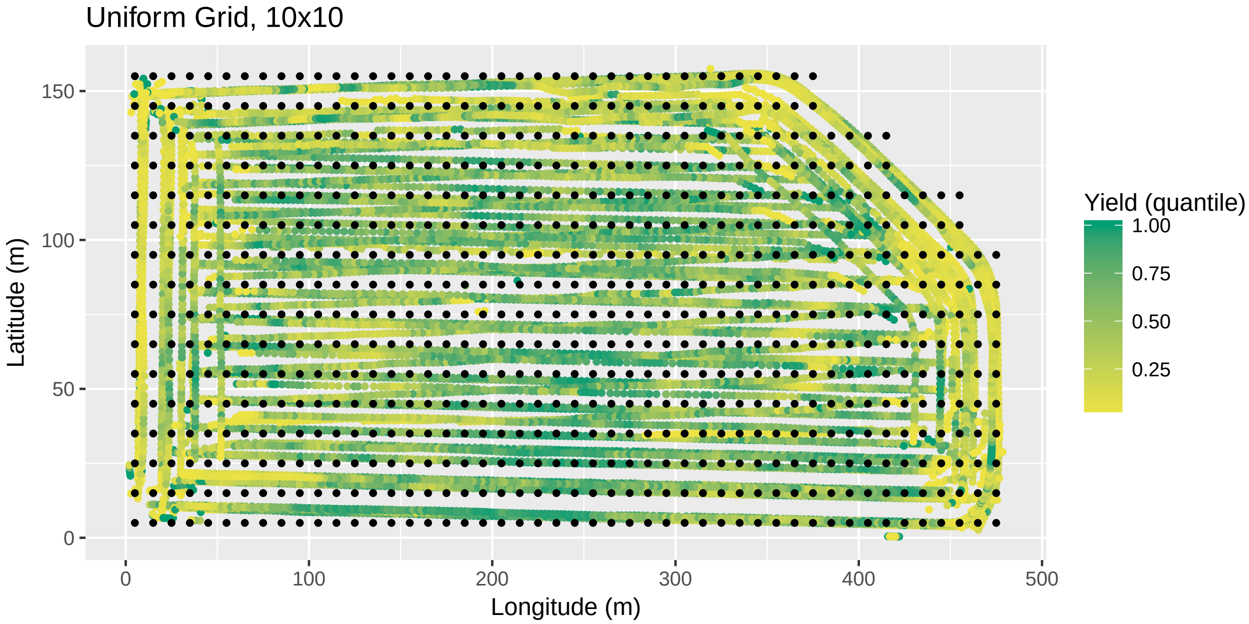


We'll keep any grid cell that has at least 20 observations.

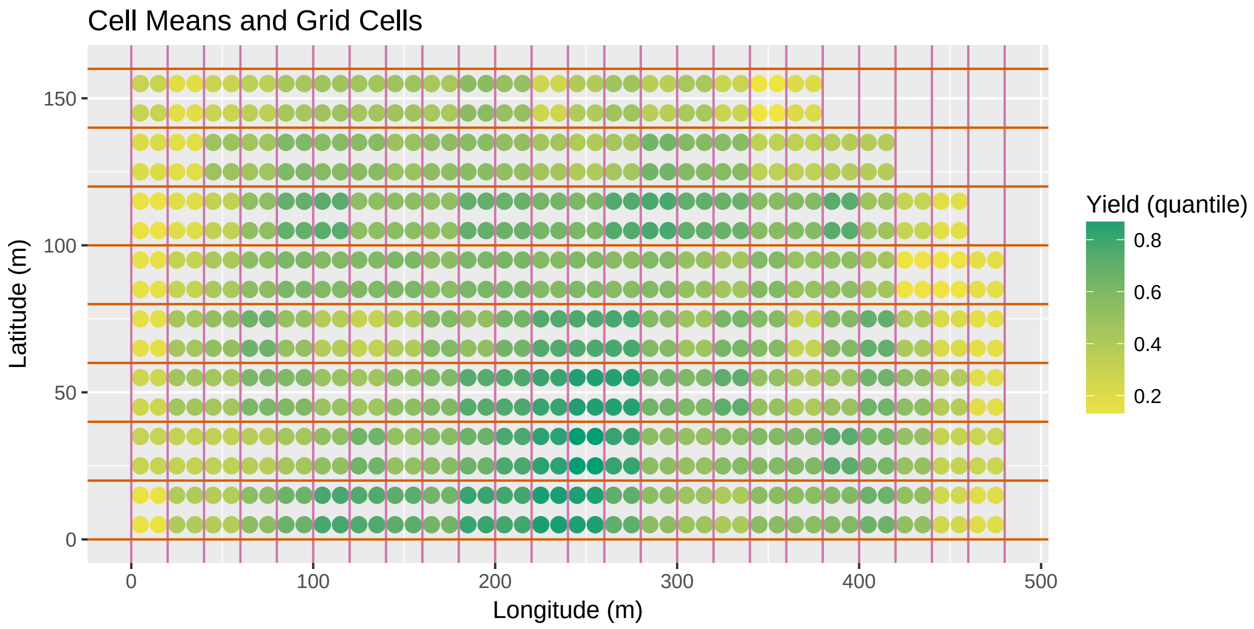
```
Grid.dat <- subset(Grid.dat,Grid.dat$CellCounts>19)
# this is cheap way to remove extra cells
Grid.dat$Cell <- as.factor(as.character(Grid.dat$Cell))
Sample10.grd <- subset(Sample10.grd, Sample10.grd$Cell %in% levels(Grid.dat$Cell))
Sample10.grd$CellMeans <- as.vector(CellMeans[as.character(Sample10.grd$Cell)])
head(Sample10.grd)
```

```
##   Northing Easting Row Column Cell CellMeans
## 1        5       5   1      1  1:1 0.1375757
## 2       15       5   1      1  1:1 0.1375757
## 3       25       5   2      1  2:1 0.2965506
## 4       35       5   2      1  2:1 0.2965506
## 5       45       5   3      1  3:1 0.2657906
## 6       55       5   3      1  3:1 0.2657906
```

```
ggplot(Pooled.dat, aes(Easting, Northing)) +
  geom_point(aes(colour = Quantile),size=1) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_point(aes(Easting, Northing),data=Sample10.grd, size=1) +
  labs(colour = "Yield (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Uniform Grid, 10x10")
```



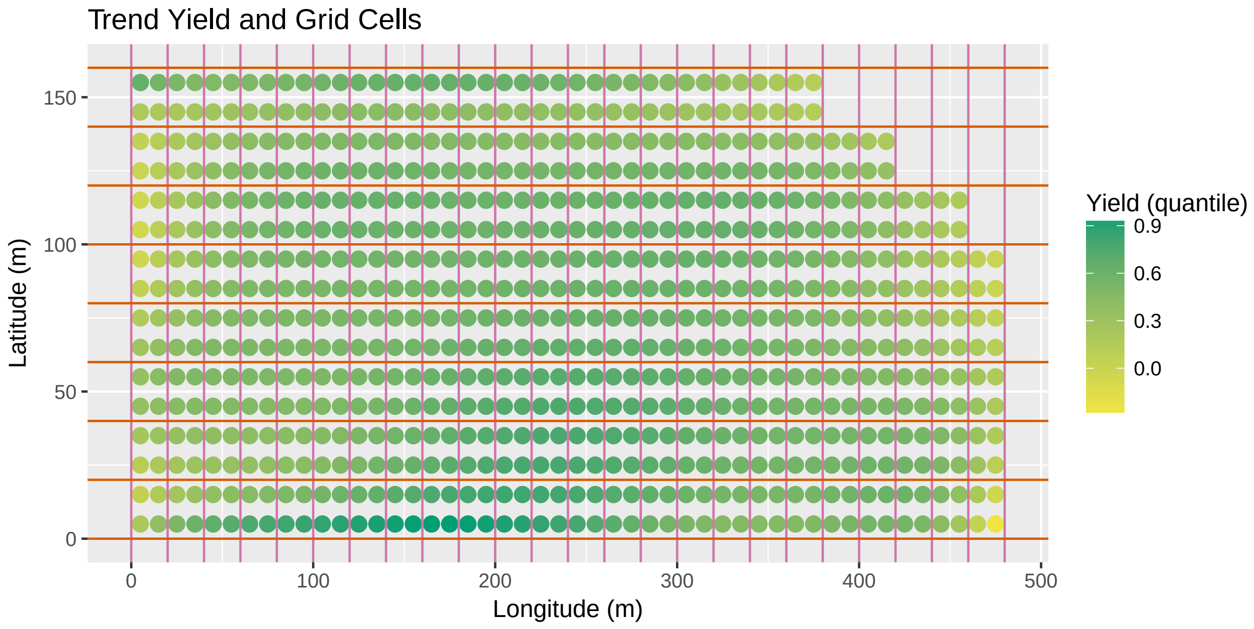
```
ggplot(Sample10.grd, aes(Easting, Northing)) +
  geom_point(aes(colour = CellMeans),size=3) +
  scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Cell Means and Grid C
```



Trend Estimates

```
load(file="Trends.Rda")
Sample10.grd$Quantile.trend <- predict(Quantile7.lm,Sample10.grd)
```

```
ggplot(Sample10.grd, aes(Easting, Northing)) +
  geom_point(aes(colour = Quantile.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 (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Trend Yield and Grid
```



```
TrendMeans = tapply(Sample10.grd$Quantile.trend,list(Sample10.grd$Cell),mean,na.rm=TRUE)
Grid.dat$TrendMeans <- as.vector(TrendMeans[as.character(Grid.dat$Cell)])
```

Kriging

Load our variograms

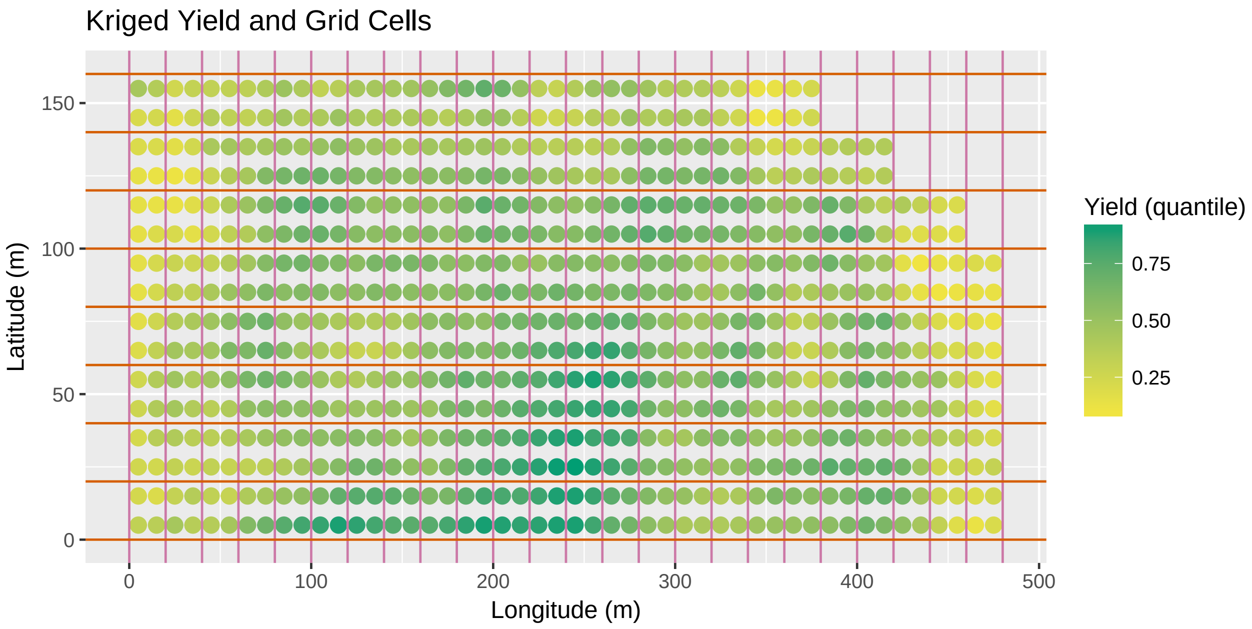
```
load(file="Variograms.Rda")
```

On my machine, kriging this data set takes about 2 hours. We'll save the fitted values

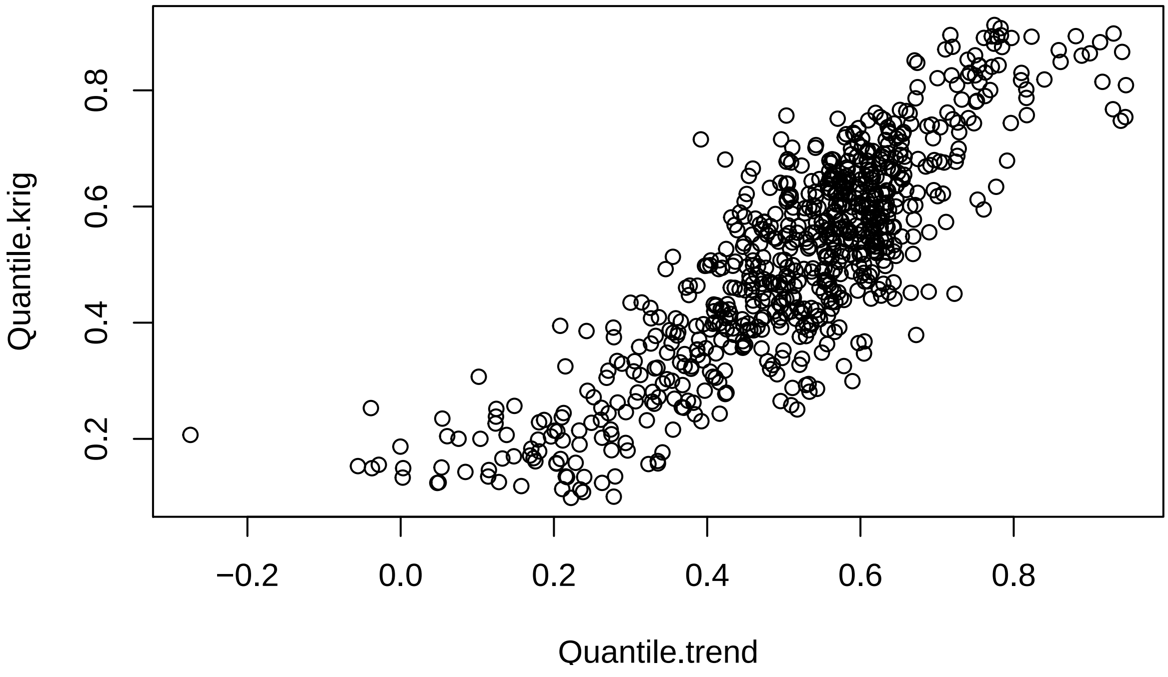
```
if(!file.exists("GridKrig.Rda")) {
  Quantile.krig <- krige(id="Quantile",
                     formula=Quantile~1,
                     data = Pooled.dat,
                     newdata = Sample10.grd,
                     model = Pooled.vgm,
                     maxdist = 100,
  locations=~Northing + Easting)
  save(Quantile.krig,file="GridKrig.Rda")
} else {
  load(file="GridKrig.Rda")
}
Sample10.grd$Quantile.krig <- Quantile.krig$Quantile.pred
KrigMeans = tapply(Sample10.grd$Quantile.krig,list(Sample10.grd$Cell),mean,na.rm=TRUE)
Grid.dat$KrigMeans <- as.vector(KrigMeans[as.character(Grid.dat$Cell)])
```

Plot the kriged estimates

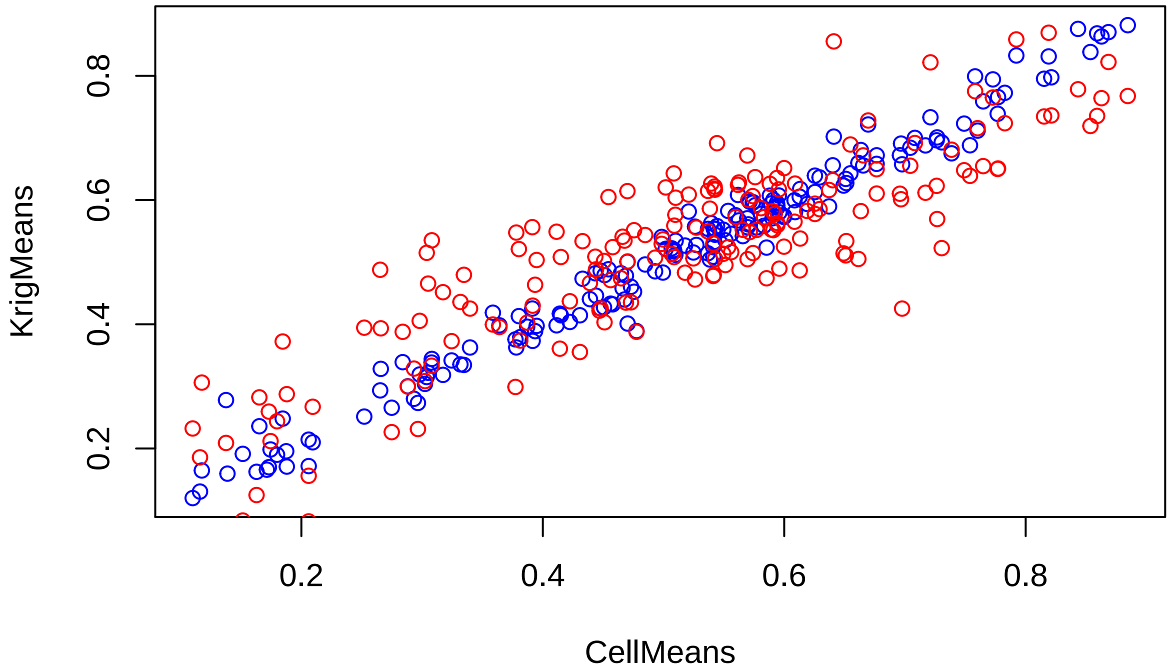
```
ggplot(Sample10.grd, aes(Easting, Northing)) +
geom_point(aes(colour = Quantile.krig),size=3) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
  labs(colour = "Yield (quantile)", x="Longitude (m)", y="Latitude (m)", title = "Kriged Yield and Grid
```



```
plot(Quantile.krig ~ Quantile.trend,data=Sample10.grd)
```



```
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$Easting, Grid.dat$Northing)))
grid.dists <- 1/grid.dists
diag(grid.dists) <- 0

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

```
## $observed
## [1] 0.1676318
##
## $expected
## [1] -0.005494505
##
## $sd
## [1] 0.007176222
##
## $p.value
## [1] 0
```

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

```
## $observed
## [1] 0.1786244
##
## $expected
## [1] -0.005494505
##
```

```
## $sd
## [1] 0.007174873
##
## $p.value
## [1] 0
```

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

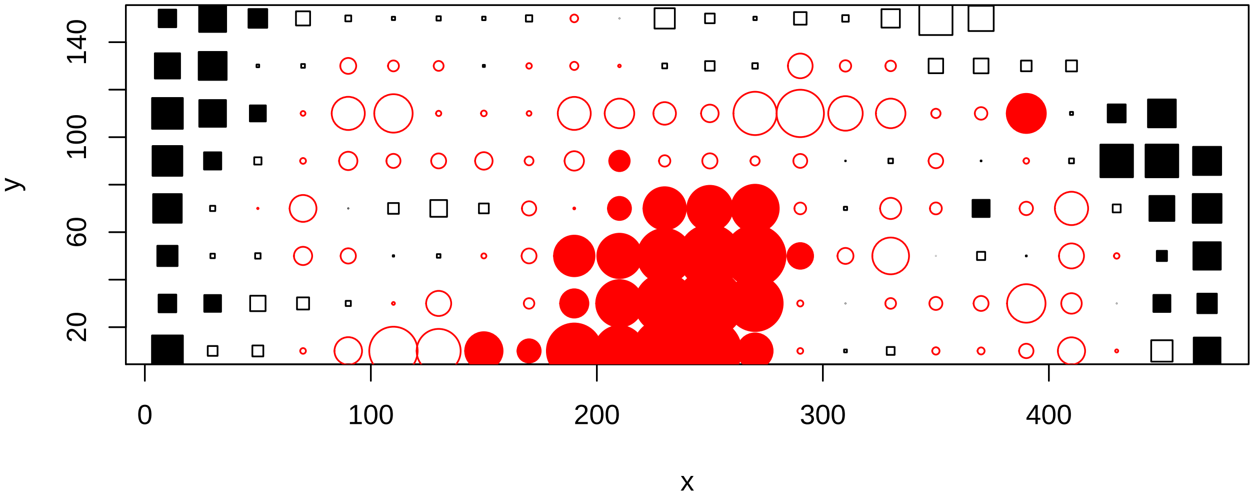
```
## $observed
## [1] 0.2045866
##
## $expected
## [1] -0.005494505
##
## $sd
## [1] 0.007146627
##
## $p.value
## [1] 0
```

Local spatial correlation.

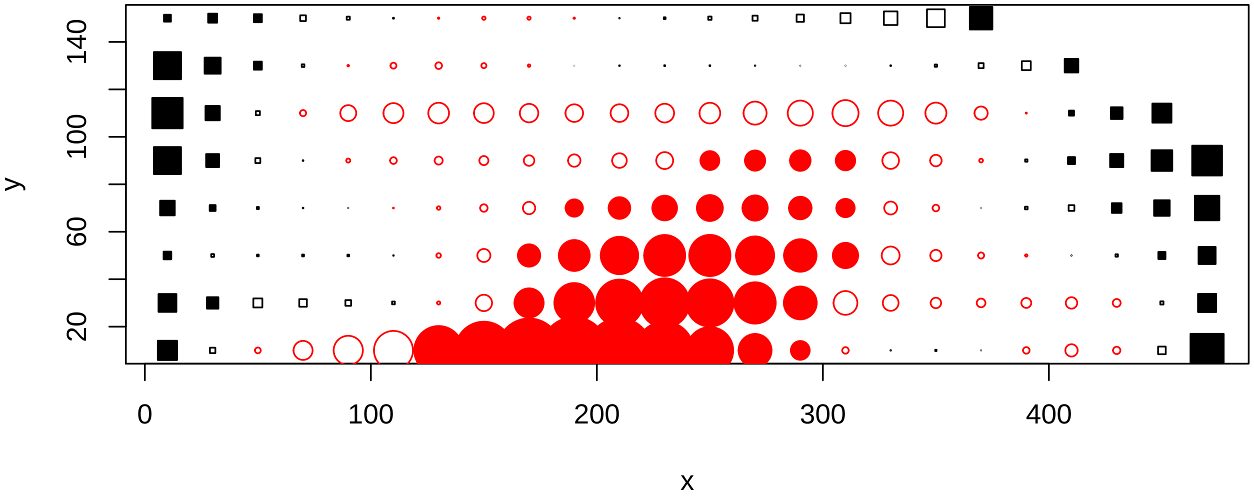
We also compute local spatial measures. We use a neighborhood of 60 - this gives us 3 grid widths.

```
CellMeans.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$CellMeans,
                       neigh=60, resamp=500, quiet=TRUE)
TrendMeans.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$TrendMeans,
                        neigh=60, resamp=500, quiet=TRUE)
KrigMeans.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$KrigMeans,
                       neigh=60, resamp=500, quiet=TRUE)
```

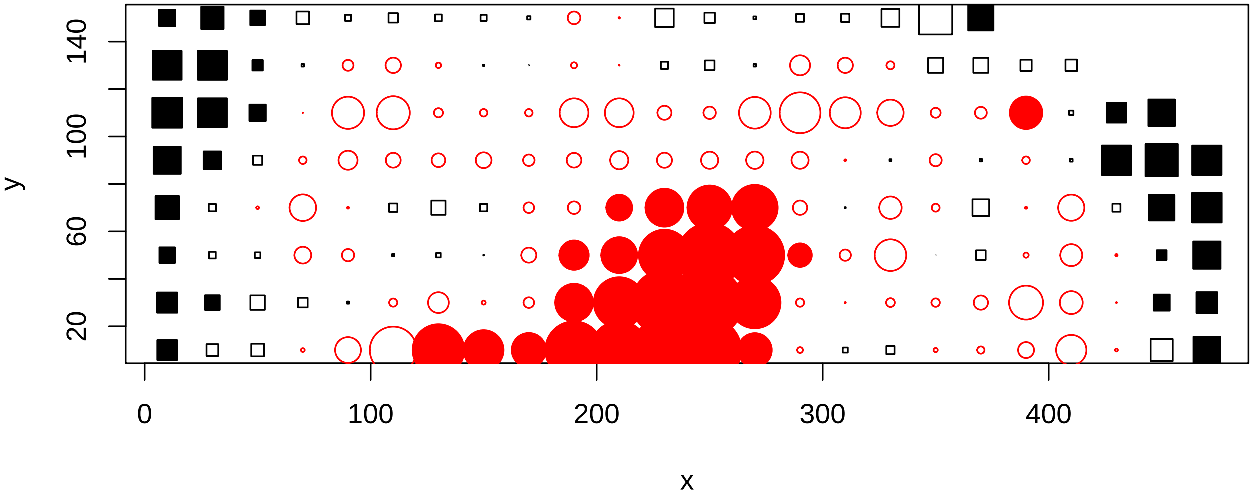
```
plot.lisa(CellMeans.lisa, negh.mean=FALSE)
```



```
plot.lisa(TrendMeans.lisa, negh.mean=FALSE)
```



```
plot.lisa(KrigMeans.lisa, negh.mean=FALSE)
```



Residuals

Global spatial correlation.

```
Pooled.dat$KrigVar <- as.vector(Pooled.dat$Quantile - KrigMeans[as.character(Pooled.dat$Cell)])^2
Pooled.dat$TrendVar <- as.vector(Pooled.dat$Quantile - TrendMeans[as.character(Pooled.dat$Cell)])^2
Pooled.dat$CellVar <- as.vector(Pooled.dat$Quantile - CellMeans[as.character(Pooled.dat$Cell)])^2
```

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

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

```
## $observed
## [1] 0.04502273
```

```
##
## $expected
## [1] -0.005494505
##
## $sd
## [1] 0.007143047
##
## $p.value
## [1] 1.52478e-12
```

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

```
## $observed
## [1] 0.04652926
##
## $expected
## [1] -0.005494505
##
## $sd
## [1] 0.007143273
##
## $p.value
## [1] 3.266276e-13
```

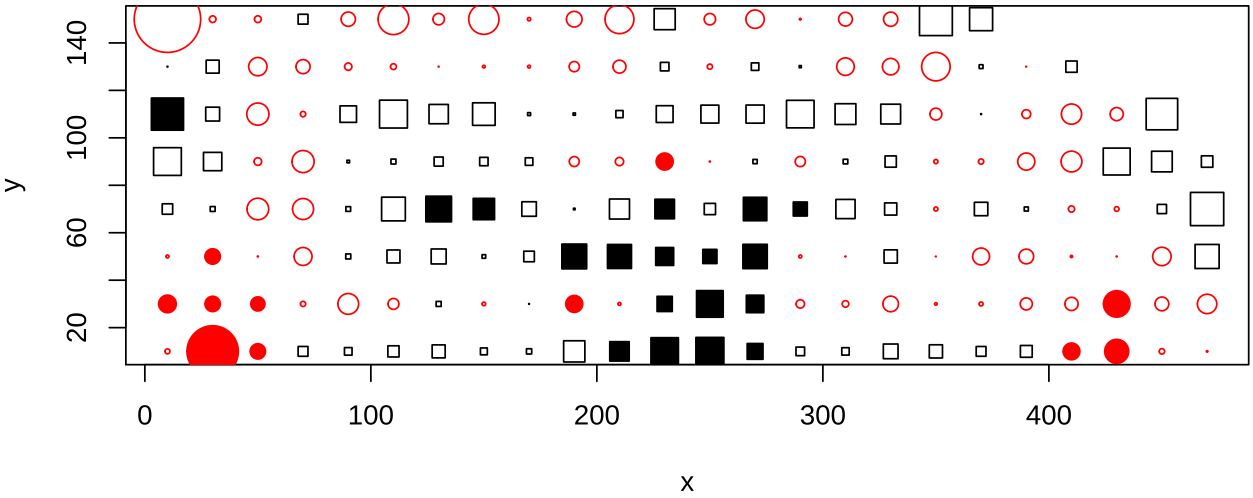
```r
Moran.I(Grid.dat$KrigVar, grid.dists)
```

```
## $observed
## [1] 0.04741974
##
## $expected
## [1] -0.005494505
##
## $sd
## [1] 0.007144507
##
## $p.value
## [1] 1.298961e-13
```

Local spatial correlation.

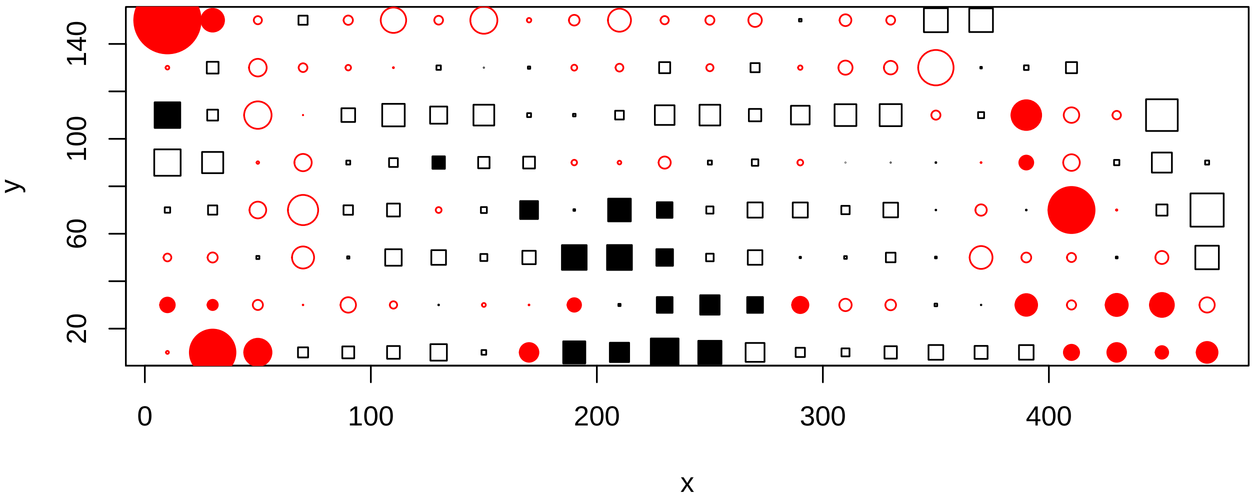
```r
library(ncf)
CellVar.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$CellVar,
                         neigh=60, resamp=500, quiet=TRUE)
```

```r
plot.lisa(CellVar.lisa, negh.mean=FALSE)
```



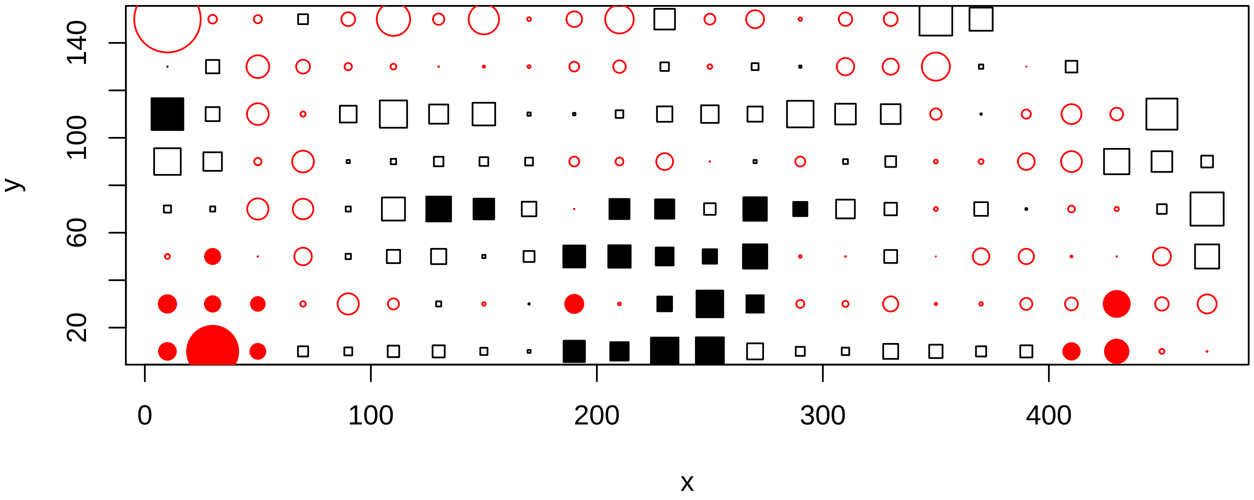
```
TrendVar.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$TrendVar,
                      neigh=60, resamp=500, quiet=TRUE)
```

```
plot.lisa(TrendVar.lisa, negh.mean=FALSE)
```

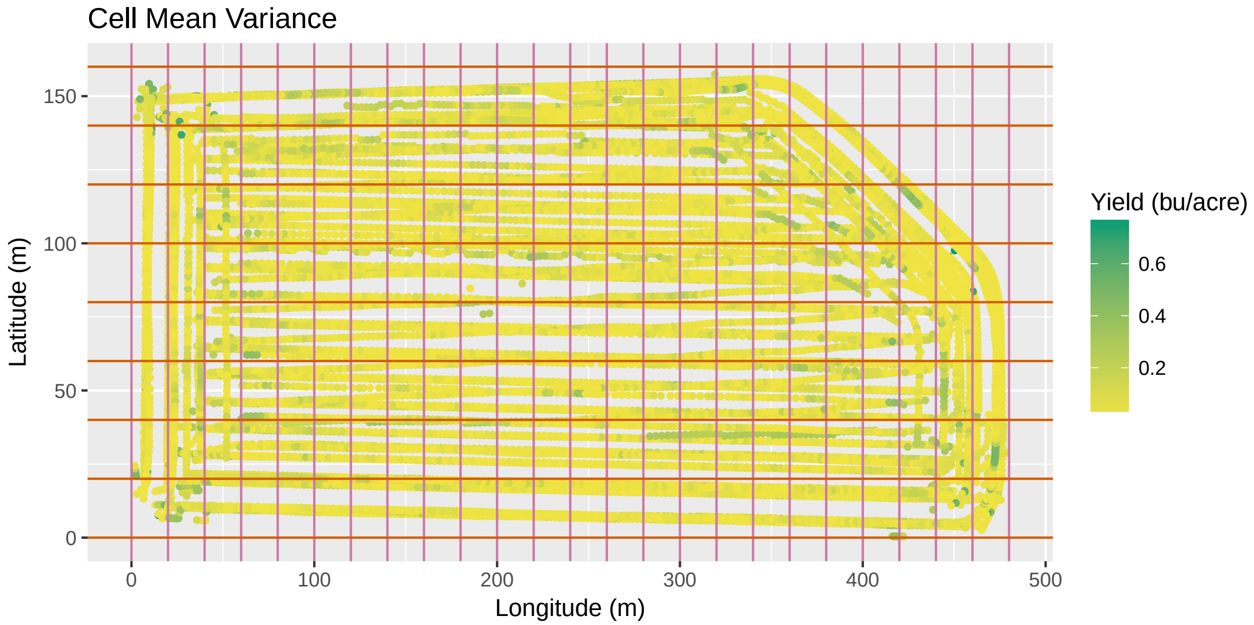


```
KrigVar.lisa <- lisa(Grid.dat$Easting, Grid.dat$Northing, Grid.dat$KrigVar,
                     neigh=60, resamp=500, quiet=TRUE)
```

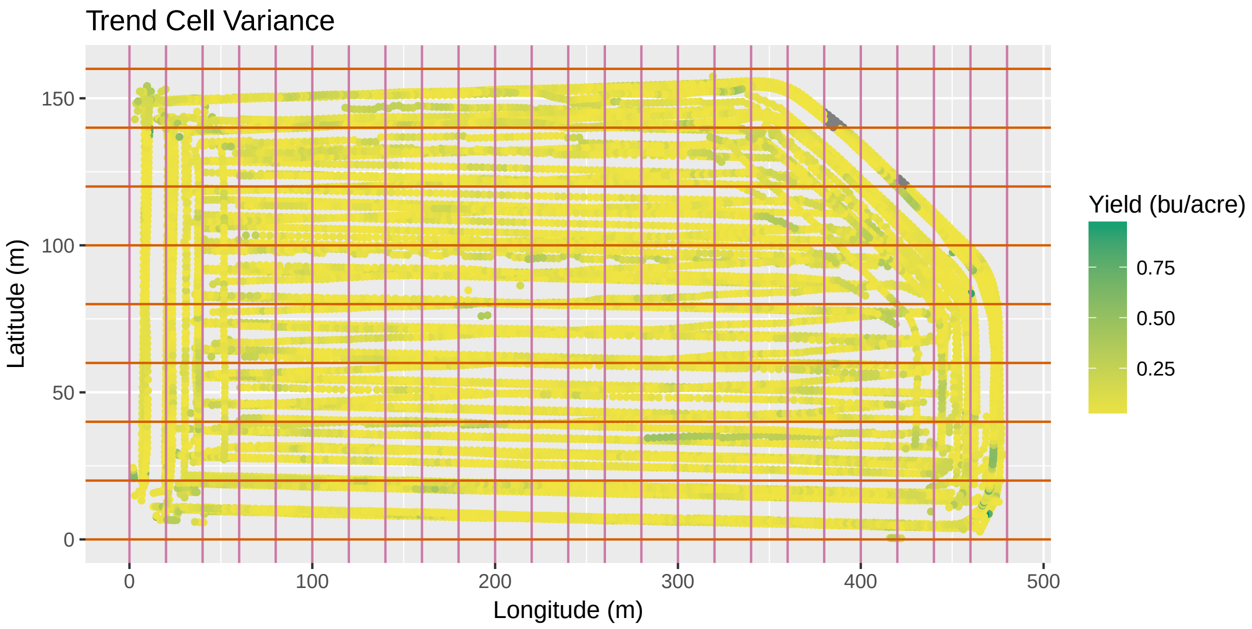
```
plot.lisa(KrigVar.lisa, negh.mean=FALSE)
```



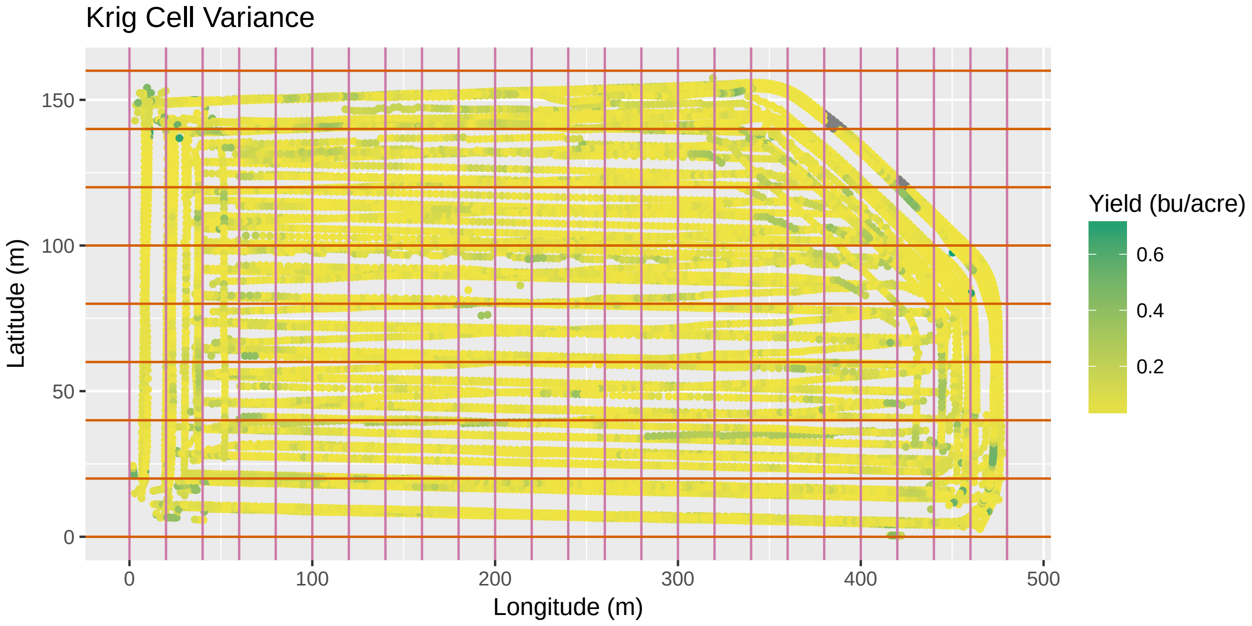
```
ggplot(Pooled.dat, aes(Easting, Northing)) +
geom_point(aes(colour = CellVar),size=1) +
scale_colour_gradient(low=cbPalette[7], high=cbPalette[4]) +
  geom_vline(xintercept = colPoints,color = cbPalette[8]) +
  geom_hline(yintercept = rowPoints,color = cbPalette[6]) +
labs(colour = "Yield (bu/acre)", x="Longitude (m)", y="Latitude (m)", title = "Cell Mean Variance")
```


Cell Mean Variance

```
ggplot(Pooled.dat, aes(Easting, Northing)) +
geom_point(aes(colour = TrendVar),size=1) +
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 Cell Variance")
```


Trend Cell Variance

```
ggplot(Pooled.dat, aes(Easting, Northing)) +
geom_point(aes(colour = KrigVar),size=1) +
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 = "Krig Cell Variance")
```


Krig Cell Variance

```
sample.dists <- as.matrix(dist(cbind(Pooled.dat$Easting, Pooled.dat$Northing)))
sample.dists <- 1/sample.dists
diag(sample.dists) <- 0
if(!file.exists("KrigVarI.Rda")) {
  CellVarI <- Moran.I(Pooled.dat$CellVar, sample.dists,na.rm=TRUE)
  TrendVarI <- Moran.I(Pooled.dat$TrendVar, sample.dists,na.rm=TRUE)
  KrigVarI <- Moran.I(Pooled.dat$KrigVar, sample.dists,na.rm=TRUE)
  save(CellVarI, TrendVarI, KrigVarI, file="KrigVarI.Rda")
} else {
  load(file="KrigVarI.Rda")
}
print(CellVarI)
```

```
## $observed
## [1] 0.03009193
##
## $expected
## [1] -8.516437e-05
##
## $sd
## [1] 0.0003868157
##
## $p.value
## [1] 0
```

```
print(TrendVarI)
```

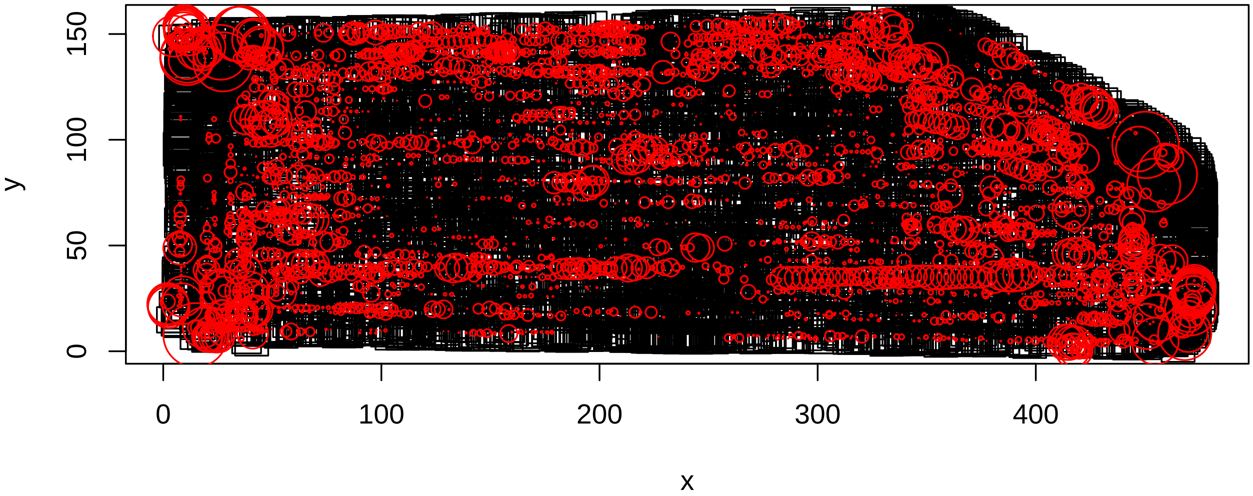
```
## $observed
## [1] 0.02939226
##
## $expected
## [1] -8.516437e-05
##
## $sd
## [1] 0.0003875628
##
## $p.value
## [1] 0
```

```
print(KrigVarI)
```

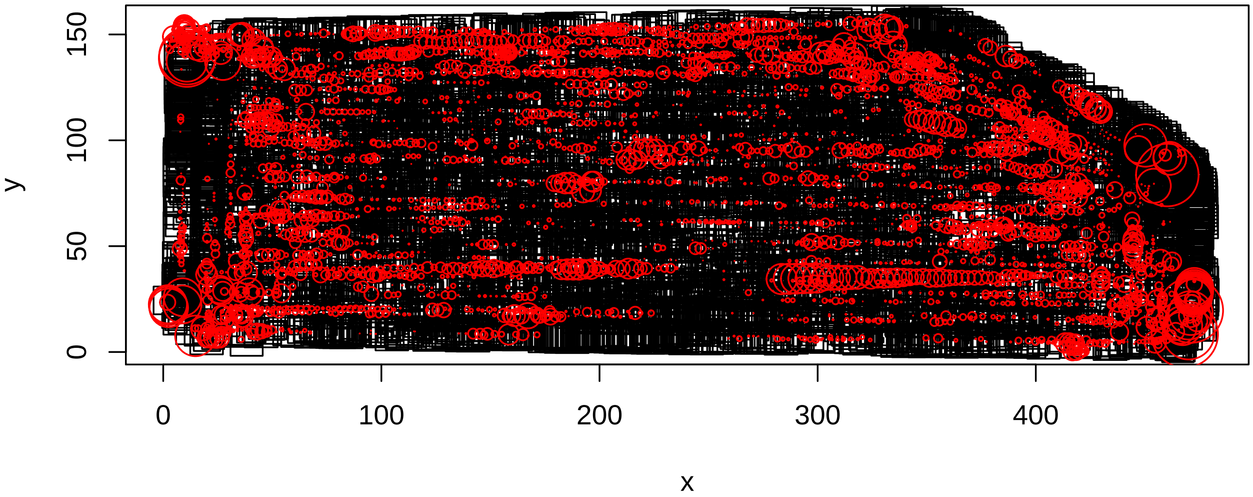
```
## $observed
## [1] 0.02939226
##
## $expected
## [1] -8.516437e-05
##
## $sd
## [1] 0.0003875628
##
## $p.value
## [1] 0
```

```
if(!file.exists("ellVarPooled.Rda")) {
  CellVarPooled.lisa <- lisa(Pooled.dat$Easting, Pooled.dat$Northing, Pooled.dat$CellVar,
                             neigh=50, resamp=10, quiet=TRUE)
  TrendVarPooled.lisa <- lisa(Pooled.dat$Easting, Pooled.dat$Northing, Pooled.dat$TrendVar,
                             neigh=50, resamp=10, quiet=TRUE)
  KrigVarPooled.lisa <- lisa(Pooled.dat$Easting, Pooled.dat$Northing, Pooled.dat$KrigVar,
                             neigh=50, resamp=10, quiet=TRUE)
  save(CellVarPooled.lisa, TrendVarPooled.lisa, KrigVarPooled.lisa, file="KrigVarLISA.Rda")
} else {
  load(file="KrigVarLISA.Rda")
}
```

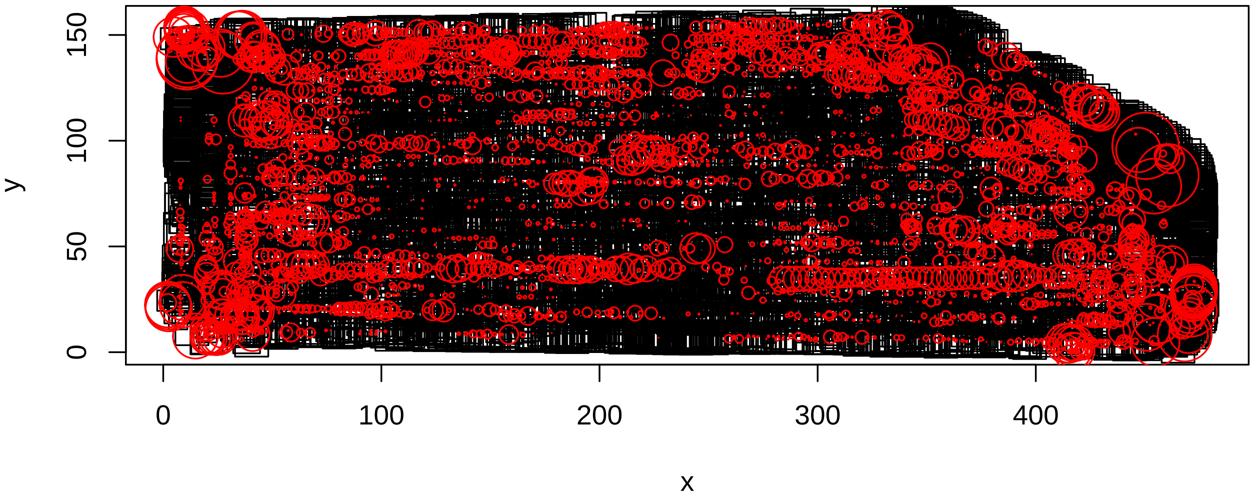
```
plot.lisa(CellVarPooled.lisa, negh.mean=FALSE)
```



```
plot.lisa(TrendVarPooled.lisa, negh.mean=FALSE)
```

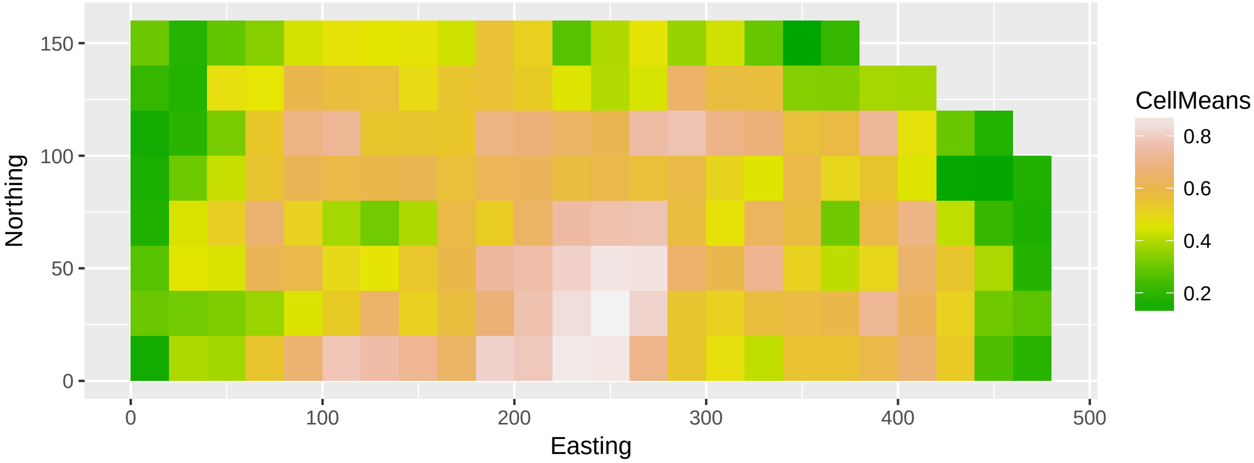


```
plot.lisa(KrigVarPooled.lisa, negh.mean=FALSE)
```

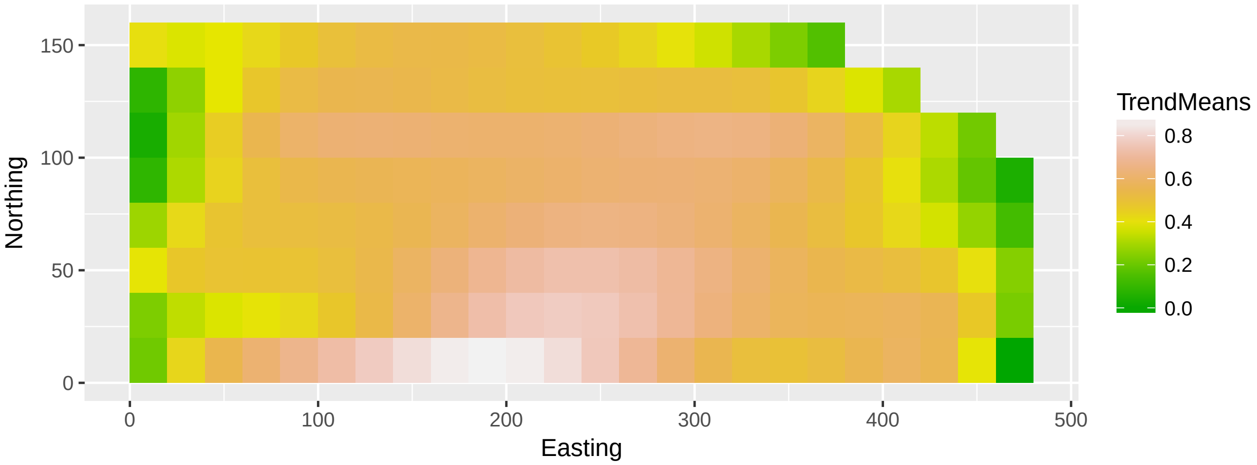


Raster Plots

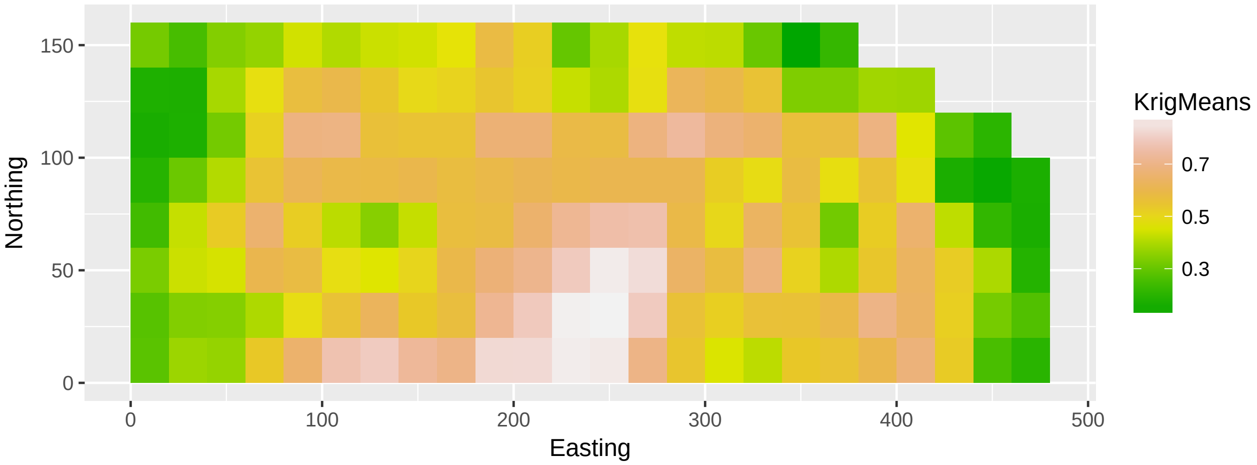
```
ggplot(Grid.dat, aes(Easting, Northing)) +
  geom_raster(aes(fill = CellMeans)) +
  scale_fill_gradientn(colours = terrain.colors(12))
```



```
ggplot(Grid.dat, aes(Easting, Northing)) +
  geom_raster(aes(fill = TrendMeans)) +
  scale_fill_gradientn(colours = terrain.colors(12))
```



```r
ggplot(Grid.dat, aes(Easting, Northing)) +
  geom_raster(aes(fill = KrigMeans)) +
  scale_fill_gradientn(colours = terrain.colors(12))
```


Northing
Easting
KrigMeans